

## PME - PROTÓTIPO DE MÃO ELETROMECAÂNICA CONTROLADA POR VISÃO COMPUTACIONAL

### PME - PROTOTYPE OF ELECTROMECHANICAL HAND CONTROLLED BY COMPUTER VISION

Guilherme Braga do Vale <sup>1</sup>; Vitor Linsbinski de Oliveira <sup>2</sup>

Giovane Maia do Vale <sup>3</sup>; Vlademir de Jesus Silva Oliveira <sup>4</sup>; João Gabriel Ribeiro Damian <sup>5</sup>

<sup>1</sup> UNEMAT- guilherme.vale@unemat.br; <sup>2</sup> UNEMAT- vitor.linsbinski@unemat.br; <sup>3</sup> UNEMAT- vale.giovane@unemat.br; <sup>4</sup> UNEMAT- vlademir.oliveira@unemat.br; <sup>5</sup> UNEMAT- joao.gabriel.ribeiro@unemat.br

#### RESUMO

A necessidade humana de manipular objetos em ambientes desafiadores levou ao desenvolvimento de dispositivos capazes de superar tais desafios como, por exemplo, mãos eletromecânicas. Seguindo esta vertente, criou-se uma réplica anatômica de mão usando impressão 3D com plástico PLA (Ácido Polilático). A essa estrutura, integrou-se, no antebraço, servomotores MG996R, com força de 11,8 kgf. Para realizar flexão e extensão dos dedos, foram usados fios de nylon de alta resistência conectados aos motores, os quais emulam tendões humanos. A mobilidade do polegar foi habilitada utilizando-se um micro servomotor, o qual trabalha em conjunto com o servomotor do antebraço. A rotação da mão ocorre no antebraço, por meio de um servomotor capaz de realizar uma rotação de até 180 graus. O controle dos servomotores é realizado via Arduino e as instruções de movimentação do braço são fornecidas por meio de um algoritmo de 'hand tracking', implementado em Python.

**Palavras-chave** — *Arduino. Hand Tracking. Programação. Python. C++.*

#### ABSTRACT

The human need to manipulate objects in challenging environments has led to the

development of devices capable of overcoming such challenges, such as electromechanical hands. Following this trend, an anatomical replica of a hand was created using 3D printing with PLA (Polylactic Acid) plastic. To this structure, MG996R servomotors, with a force of 11.8 were integrated into the forearm. High-strength nylon wires connected to the motors were used to perform finger flexion and extension, emulating human tendons. Thumb mobility was enabled using a micro servomotor, which works in conjunction with the forearm servomotor. Hand rotation occurs in the forearm through a servomotor capable of a rotation of up to 180 degrees. Control of the servomotors is carried out via Arduino, and arm movement instructions are provided through a 'hand tracking' algorithm implemented in Python.

**Keywords** — *Arduino. Hand Tracking. Programming. Python. C++.*

#### 1. INTRODUÇÃO

Devido aos avanços científicos realizados pela espécie humana, surgiu a necessidade de se efetuar trabalhos de alta precisão em zonas de considerável periculosidade ou de difícil acesso, a exemplo de áreas radioativas, de exploração subaquática, locais de desastres, dentre outras situações. É fundamental destacar que, nesses ambientes, frequentemente, são desenvolvidas



atividades de grande relevância para a sociedade contemporânea, como o manuseio seguro de materiais radioativos, resultantes do descarte de resíduos de usinas nucleares. Com o intuito de viabilizar a execução segura dessas tarefas, foi desenvolvido um protótipo de braço eletromecânico capaz de reproduzir movimentos semelhantes aos da anatomia humana. Tal dispositivo possibilitaria o manejo, com naturalidade, de instrumentos e objetos em qualquer caso em que um operador humano não consiga atuar com segurança. Assim, o objetivo principal da pesquisa foi a compreensão funcional e individual de cada dedo do PME para realização do controle de flexão. Buscou-se então tornar possível a realização de gestos e movimentos comuns e inatos do ser humano.

Foi usado como base para o desenvolvimento do sistema computacional de hand tracking, destinado ao controle do protótipo, o artigo de Raheja et al. (2010), intitulado "Real-Time Robotic Hand Control Using Hand Gestures". Por fim, com o objetivo de se averiguar o funcionamento adequado do PME, foi realizada uma série de ações comuns, similares as que são efetuadas cotidianamente por seres humanos, como segurar copos ou gesticular.

## 2. MATERIAIS E MÉTODOS

O objetivo da pesquisa foi desenvolver um protótipo de mão eletromecânica, conectada a um antebraço, que realizasse uma movimentação análoga àquela empreendida por um ser humano. Assim, na primeira fase do desenvolvimento do protótipo, foi necessário confeccionar a estrutura da mão e do antebraço, as quais foram impressas em uma impressora 3D, utilizando plástico PLA. Para a contração dos dedos foram utilizados servomotores MG996R, com 11,8 *kgf*, os quais foram alocados no interior do antebraço, fixos em um suporte interno e conectados às linhas de nylon de 12 *kgf*. Estas linhas passam por dentro da estrutura da palma da mão do protótipo e por cada dedo em particular. Dessa forma, é possível realizar a contração dos dedos individualmente. Porém, a distensão, isto é, o “relaxamento” dos dedos, se dá pelo uso de fibras elásticas, as quais são tencionadas

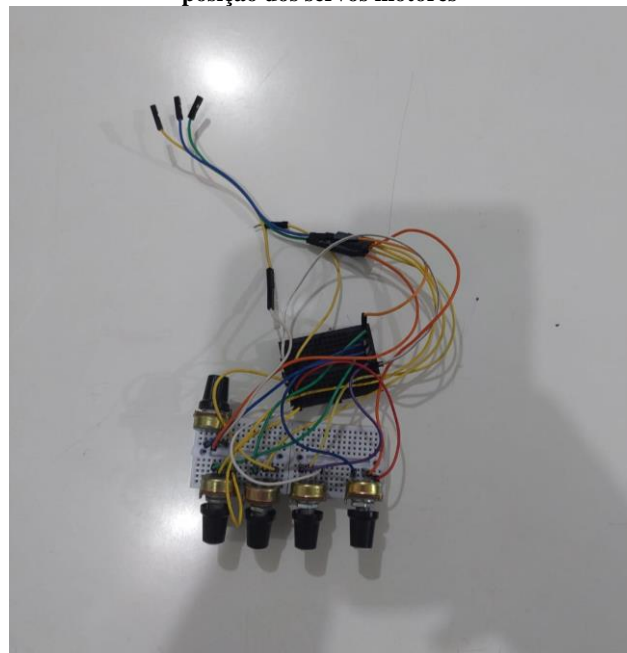
durante o fechamento (ou contração) dos dedos. Logo, cabe às fibras elásticas fazerem com que cada dedo retorne a sua posição inicial, forçando-os à abertura dos dedos, quando os motores deixam de tracionar.

A cada motor, associado a um dos dedos do protótipo, foi acoplado um potenciômetro, destinado ao seu acionamento e respectivo movimento de dedo.

Um importante procedimento a ressaltar, foi a sincronização do ângulo inicial de cada motor com relação a seu respectivo potenciômetro, antes mesmo que estes fossem fixados no antebraço do protótipo. Para tal sincronização, foi empregado um código em Arduino que, utilizou a biblioteca “Servo.h” e a função *map*. Assim, foi possível equiparar a posição inicial de cada servomotor com relação ao valor “0” (zero), advindo de seu respectivo potenciômetro.

Os valores provenientes dos potenciômetros podem variar de “0” a “1023”. Nestes termos, a posição inicial de cada motor correspondente ao valor 0 (potenciômetro) e sua rotação está atrelada à variação de seu potenciômetro associado. Na imagem da Figura 1 tem-se o conjunto de potenciômetros citados acima.

Figura 1 – Potenciômetros utilizados para o ajuste da posição dos servos motores



Fonte: Própria (2023).

Além da parte física do projeto, foi necessária a criação de dois códigos em duas linguagens distintas para o controle da mão:

- 1) **Python:** esta linguagem foi usada na programação do *hand tracking*, que é a rotina responsável pela aquisição da imagem da mão do usuário, capturada via *webcam*, e interpretação de sua posição. Após a interpretação da posição da mão do usuário, a rotina envia comandos ao Arduino que, por sua vez, aciona os respectivos servomotores, gerando o movimento desejado.
- 2) **C++:** o código implementado em C++, embarcado no Arduino, é responsável por receber os dados booleanos (1 ou 0) oriundos do código em Python (*hand tracking*) para o controle dos servomotores conectados a cada dedo, de forma individual, e para o acionamento do servomotor que efetua a rotação do antebraço.

Para o desenvolvimento do algoritmo de visão computacional em Python foram utilizadas as bibliotecas OpenCV, CVZone e pySerial. Estas bibliotecas, que utilizam métodos previstos em Gonzalez e Woods (2010), foram empregadas da seguinte forma:

- **OpenCV:** em primeira instância é necessário que seja iniciada a captação de imagens através da *webcam* presente no computador. No entanto, para que o processo em questão funcione, é preciso definir o tempo de captura de *frames*. Assim, é possível que o vídeo capturado seja lido *frame a frame* e utilizados apenas alguns destes *frames*, segundo uma taxa de coleta pré-definida. A rotina destinada à esta amostragem encontra-se na OpenCV. Os *frames* coletados são então subsequentemente analisados.
- **CVZone:** esta biblioteca é utilizada na interpretação dos dados (imagens), via *hand tracking*. Ou seja, os *frames* capturados via OpenCV são analisados pelo algoritmo de *hand tracking*.e, a partir da análise dos *frames*, é construída uma lista de valores, relativos ao estado de cada dedo. A lista é então enviada ao Arduino. A cada imagem

analisada, uma nova lista de valores é gerada e enviada ao Arduino que, por sua vez, atualizará a posição dos dedos do protótipo. Para tal, foram utilizados os parâmetros presentes na própria biblioteca, como o ilustrado na Figura 2.

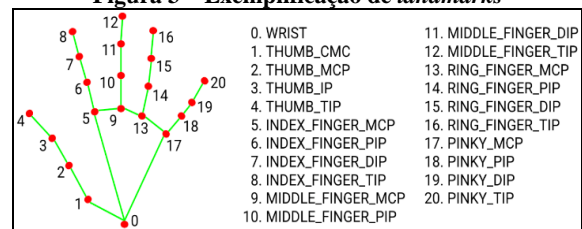
Figura 2 – Parâmetros inerentes à biblioteca CVzone para *hand tracking*.

```
from cvzone.HandTrackingModule import HandDetector
```

Fonte: Própria (2023).

Durante o procedimento de detecção, são marcados 21 pontos na imagem da mão a fim de se estipular a diferença, a distância e o estado de cada dedo em um espaço tridimensional, pois, como trata-se de uma imagem, não há, para a máquina, a percepção de profundidade. Tais pontos são chamados comumente de *landmarks* e são usados para identificar a segmentação das falanges dos dedos como apresentado abaixo (Figura 3).

Figura 3 – Exemplificação de *landmarks*



Fonte: Mediapipe (2023).

Assim, a partir das coordenadas demarcadas na imagem são traçadas retas entre os pontos, sendo que, a cada par de pontos há uma reta distinta que os conecta. Dessa forma, o reconhecimento da flexão ou distensão do dedo se dá a partir da análise do ângulo entre tais retas. Quando a reta referente a um dedo específico e as retas que delimitam a palma da mão formam um ângulo menor ou igual a 90° o programa compreende que o dedo está flexionado e há o envio, para o Arduino, do valor “0”, relativo ao respectivo dedo.

Os valores, relativos à cada dedo, são integrados à lista, citada anteriormente. A lista é atualizada através de lógica análoga à função “.append()”, da linguagem Python. O conjunto de valores iniciais “1, 1, 1, 1, 1”, relativos a cada um dos cinco dedos, indica que a mão está aberta, não havendo dedos

flexionados. A lista é então atualizada de acordo com a taxa de amostragem definida por meio da própria interface Python. Portanto, sempre haverá um único valor presente para cada dedo, o qual corresponderá ao estado mais recente da mão do operador. Este método, previne que haja um valor anteriormente salvo que possa ser lido de forma errônea pelo microcontrolador.

Apesar das imagens serem captadas pela *webcam* em tempo real, há um *delay* estipulado em 30 milissegundos, através da função “`cv2.waitKey(30)`”, a fim de que o programa aguarde o tempo determinado antes de passar para o próximo *frame*. Este processo viabiliza o cancelamento do programa (interrupção da execução) ao pressionar determinada tecla e ainda mantém a fluidez do vídeo. Tal taxa de exibição pode ser regulada a critério do usuário. No entanto, a fim de se realizar testes com maior precisão, foi mantido um valor intermediário de *delay*.

- **pySerial:** a fim de realizar a comunicação entre o computador e a placa de controle dos motores (Arduino), foram utilizadas as funções oriundas da biblioteca PySerial para envio de dados binários (0 ou 1), presentes na lista em formato “*int*”. Ou seja, para a realização do processo, o Arduino deve estar plugado ao computador via porta USB, permitindo a comunicação serial entre a porta COM, do computador, e ele. A Figura 4 ilustra o código relativo à esta conexão.

Figura 4 – Especificação da Porta Serial

```
arduino = serial.Serial('COM4', 9600)
```

Fonte: Própria (2023).

Além disso, a fim de assegurar o funcionamento adequado do envio dos dados presentes na lista, foi criada uma função destinada a verificar tal conectividade. A Figura 5 ilustra o referido código.

Toda a programação do sistema de visão computacional foi desenvolvida no *software* de programação Pycharm. O programa desenvolvido no Pycharm permitiu com que os algoritmos de *hand tracking* e o sistema de

controle de motores atuassem de modo conjugado. Para o primeiro, têm-se como entrada (*input*) a posição da mão do usuário e de seus dedos, capturada via *webcam*. Após o processamento de dados, gera-se como saída uma lista de valores booleanos (0 ou 1). Cada valor é inerente a um dedo e indica o seu estado: distendido ou flexionado. Ou seja, para cada dedo tem-se valores “1” ou “0”, que indicam aos servomotores, respectivamente, se o dedo estará estendido (1) ou flexionado (0). O valor booleano inerente à posição relativa da mão, correlacionada à rotação do antebraço, também está presente na lista.

Figura 5 – Função de verificação de conectividade

```
while True:
    try:
        print('Arduino conectado')
        break
    except:
        print("Falha ao conectar...")
        break
```

Fonte: Própria (2023).

O diagrama, presente na Figura 6, especifica, sequencialmente, os procedimentos executados pelos sistemas envolvidos, desde a captura dos dados, até a realização dos movimentos do PME.

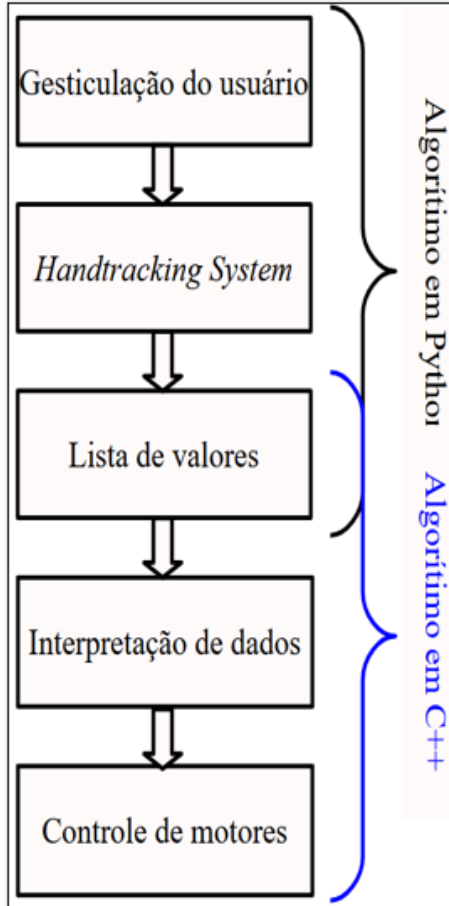
O sistema que efetua a comunicação entre o algoritmo de visão computacional, responsável pela identificação dos dedos de uma mão, e o microcontrolador, foi crucial no desenvolvimento do projeto, pois ele tem a função crucial de facilitar a interação entre os *hardwares*. Cabe ressaltar que, o código em Python foi elaborado para se utilizar da comunicação serial tal e tal processo envolve a transmissão de informações *bit a bit*, por meio de uma porta digital ou barramento, no caso, a porta USB do computador. A função do programa é atuar como um emissor de sinais booleanos.

No contexto dos sistemas embarcados, especificamente aquele inserido na placa de controle Arduino, especifica-se que o código foi desenvolvido em linguagem C++, adaptada para a plataforma Arduino IDE. Esse algoritmo lida com as informações provenientes da leitura das



articulações dos dedos, atuando como um receptor de sinais. Além disso, ele define as saídas para cada motor utilizado no protótipo da mão e os aciona de acordo com uma lógica preestabelecida.

Figura 6 – Diagrama sequencial de funcionamento do PME



Fonte: Própria (2023).

No código para Arduino, foram empregados métodos de processamento de dados simples. Inicialmente, foi verificada a presença de comunicação serial na porta selecionada. Como as informações chegam no formato de “strings”, ou seja, cadeias de caracteres, não é possível executar operações booleanas ou inteiras diretamente. Para contornar esse obstáculo, foi necessário dividir a “string” em “substrings” separadas por vírgulas, colchetes e espaços. Desse modo a recepção dos dados se dá, por exemplo, no seguinte formato: “[0,1,0,1,1]”. Essa tarefa foi realizada por meio do uso de métodos nativos da linguagem C++, como a função “strtok”, como o mostrado na Figura 7, abaixo.

Figura 7 – Separação da string por vírgulas

```
char *value = strtok(&input[0], ", [ ] ");
```

Fonte: o autor (2023).

Para facilitar a manipulação de cada dedo, uma variável do tipo lista, denominada “fingersArray”, foi instanciada com tamanho fixo de cinco para armazenar valores inteiros (0 ou 1). Um looping foi então executado na “string” recebida até o último caractere, convertendo cada 0 e 1 do formato de “string” para o formato de int (número inteiro) com o método “atoi” e inserindo-os na lista. O resultado final seria, por exemplo, [0,1,0,1,1]. Na Figura 8 tem-se o código usado para a referida conversão.

Figura 8 – Loop de Conversão de string para números inteiros

```
while (value != NULL && i < 5) {
  fingersArray[i] = atoi(value);
  value = strtok(NULL, ", [ ] ");
  i++;
}
```

Fonte: Própria (2023).

Por fim, se fez necessária a criação de uma função intitulada de “rotateFingers”, com retorno inteiro, que recebe como parâmetro um número inteiro (0 ou 1). Ela converte 0 em 180° e 1 em 0°, os quais representam respectivamente os graus que o servomotor deve girar em relação ao seu eixo. Cada motor, então, chama a função “write” com o valor correspondente da lista de valores inteiros, enviando à porta analógica do Arduino as informações necessárias para execução do movimento de rotação dos servomotores, e conseqüentemente, a contração, conforme o preconizado por Raheja et al. (2010). O referido código pode ser visto na Figura 9, abaixo.

Figura 9 – Funções referentes ao controle de rotação dos motores

```
servoPolegar.write(rotateFinger(!fingersArray[0]));
servoPolegar2.write(rotateFinger(!fingersArray[0]));
servoIndicador.write(rotateFinger(!fingersArray[1]));
servoMedio.write(rotateFinger(fingersArray[2]));
servoAnelar.write(rotateFinger(fingersArray[3]));
```

Fonte: Própria (2023).

Em suma, a partir da webcam de um laptop, a gesticulação da mão de um usuário é capturada. Por sua vez, as imagens coletadas são processadas pelo algoritmo de hand

*tracking*, executado no *laptop*, o qual gera a lista de valores booleanos necessários ao Arduino. Este último, via código C++, recebe e interpreta os dados recebidos e aciona coerentemente os motores do PME, replicando os movimentos executados, *a priori*, pelo usuário. Do exposto, verifica-se ser evidente a dependência que o segundo sistema (Algoritmo em C++) possui com relação à saída do primeiro (Algoritmo em Python). A interação entre os algoritmos constitui um sistema instantâneo (Lathi, 2006), tendo em vista que o acionamento ou não dos servomotores independe de um armazenamento prévio de informações para assumir seu próximo estado. O sistema é também considerado causal, conforme o preconizado por Nise (2012).

Para a alimentação do *hardware* foi utilizado um filtro de linha com seis entradas, nas quais foram acopladas três fontes de alimentação de câmeras de segurança, cada uma com tensão de **6V** e corrente de **2A** (valores nominais). Foram também utilizadas outras duas fontes de alimentação com tensão de **5V** e corrente de **2A** (valores nominais). As fontes destinam-se à alimentação dos motores e necessitam de uma tensão de entrada de **110V**. Às saídas das fontes foram acoplados bornes. Os bornes atuam como facilitadores de conexão para os polos positivos e negativos.

Por fim, para que o protótipo funcionasse a contento, foi necessário ligar os fios terra da saída das fontes de energia ao ground (GND) do Arduino, a fim de poder se controlar cada motor, sem que houvesse o risco de curto-circuito. Quanto ao polo positivo, estes foram ligados diretamente aos motores do protótipo.

Cabe ressaltar que, nessa pesquisa se construiu um Protótipo de Mão Eletromecânica, e que, justamente por se tratar de um protótipo, o material, do qual sua estrutura é composta, não possui propriedades físicas que o capacite para a realização de trabalhos em condições críticas de estresse ou de altas temperaturas. Especifica-se que, o filamento plástico PLA tem como temperatura de fusão  $57^{\circ}$  Celsius, o que inviabiliza ações simples e cotidianas como, por exemplo, utilizar o PME para segurar um copo com café quente, visto que sua temperatura média varia entre  $82$  a  $85^{\circ}$  Celsius.

A fim de sanar esta limitação, a região referente à palma da mão e ao interior dos dedos pode ser revestida com borracha. Dessa forma, o revestimento poderia inibir a troca de calor entre o material da mão e o da substância aquecida. Colateralmente, o uso do revestimento poderia aumentar, tanto a resistência mecânica da mão, quanto a aderência, ao se segurar objetos mais lisos.

### 3. RESULTADOS E DISCUSSÃO

Com o propósito de efetuar uma avaliação meticulosa da eficácia e funcionalidade do protótipo, foram conduzidos testes para se aferir o correto desempenho da mão eletromecânica. Essa abordagem possibilitou a identificação de inconsistências e a implementação de ajustes necessários. Os testes em questão foram abrangentes e destinaram-se a averiguar a capacidade da mão em replicar os gestos realizados pelo usuário.

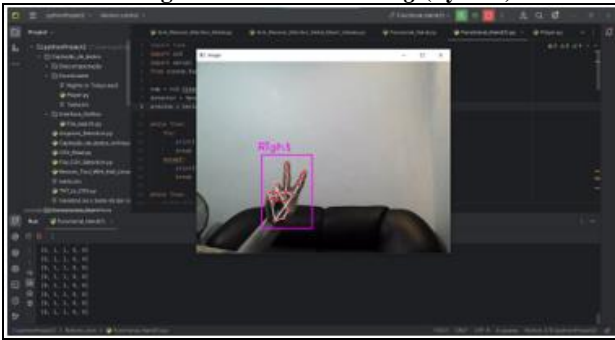
Dentre as diversas gestualidades examinadas, um dos exemplos digno de nota foi a reprodução do sinal de "paz e amor", que consiste em elevar os dedos indicador e médio enquanto os demais permanecem abaixados. Nesse caso, obteve-se a lista resultante: [0, 1, 1, 0, 0], em que "0" indica um dedo flexionado e "1" representa um dedo estendido.

Cabe especificar que, a partir dos dados coletados pelo algoritmo de visão computacional, foi gerada uma lista de valores booleanos, os quais foram atualizados, aproximadamente, a cada 0,043 segundos. Assim, existe um atraso (*delay*) insignificante entre a captura da imagem da mão do usuário e a geração de um novo conjunto de valores para representar essa situação específica.

Esses dados foram transmitidos de forma contínua para o Arduino por meio de comunicação serial. O Arduino, por sua vez, executou com precisão o acionamento e controle dos servomotores, viabilizando assim a reprodução análoga ao gesto realizado pelo usuário. A Figura 10 mostra a interface do programa de *hand tracking*, quando da captura e interpretação da imagem da mão do usuário. Por sua vez, a imagem presente na Figura 11

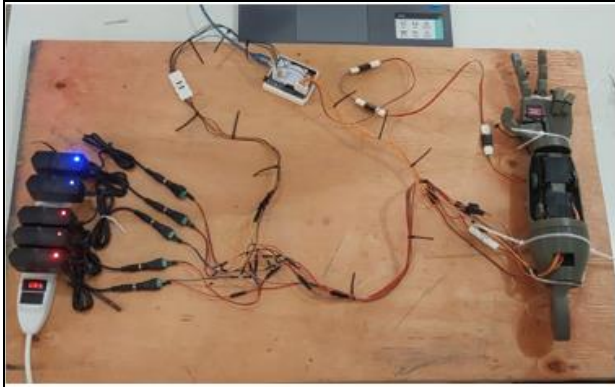
apresenta o PME replicando o referido movimento.

**Figura 10. Reconhecimento das posições dos dedos através do algoritmo de *hand tracking* (Python)**



Fonte: Própria (2023).

**Figura 11. PME reproduzindo a atitude da mão do usuário**



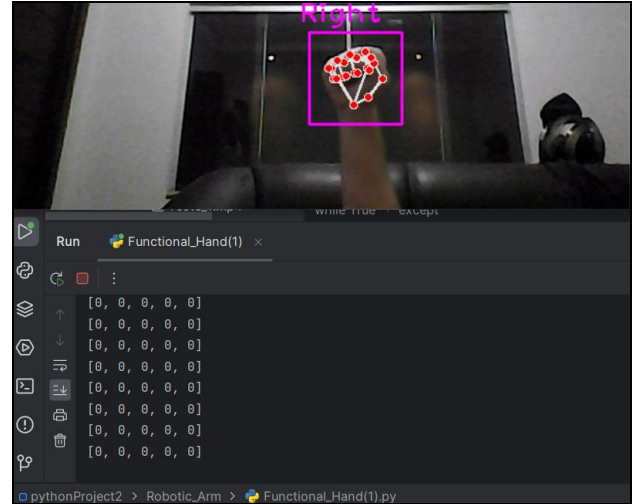
Fonte: Própria (2023).

A representação gráfica da interface do software de *hand tracking* é evidenciada na Figura 12. Na porção superior desta representação, é possível visualizar a mão do usuário, acompanhada de sua correspondente interpretação gráfica. Esta interpretação é meticulosamente destacada através de uma combinação de cores vermelhas e brancas, cuidadosamente sobrepostas à imagem da própria mão. Por outro lado, a seção inferior da figura apresenta uma sequência detalhada de listas sucessivas, elaboradas com o intuito específico de direcionar e orientar os movimentos executados pela mão eletromecânica (PME).

Com o propósito de realizar uma análise minuciosa da eficácia e funcionalidade do protótipo, foram conduzidos testes adicionais para avaliar sua capacidade de segurar objetos em suas posições corretas de uso. Um desses testes consistiu em segurar um rolo grande de fita adesiva sem utilizar de qualquer apoio ou suporte, visando determinar a estabilidade e

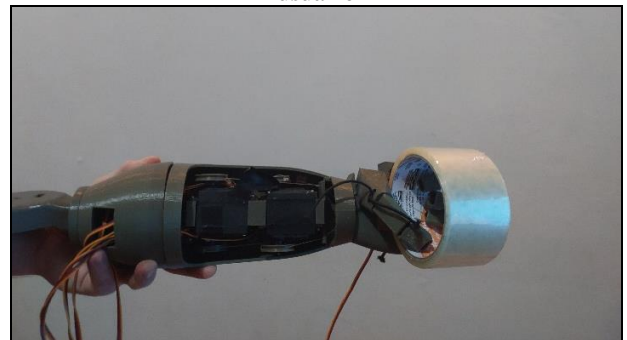
firmeza proporcionadas pelo dispositivo. A Figura 13 ilustra o desfecho desse teste, fornecendo um exemplo prático da operação do protótipo sobre o desempenho do protótipo nesse contexto específico.

**Figura 12. Gesticulação da mão do usuário e dados enviados ao Arduino**



Fonte: Própria (2023).

**Figura 13. PME segurando um rolo grande de fita adesiva a partir das ações da mão do usuário**

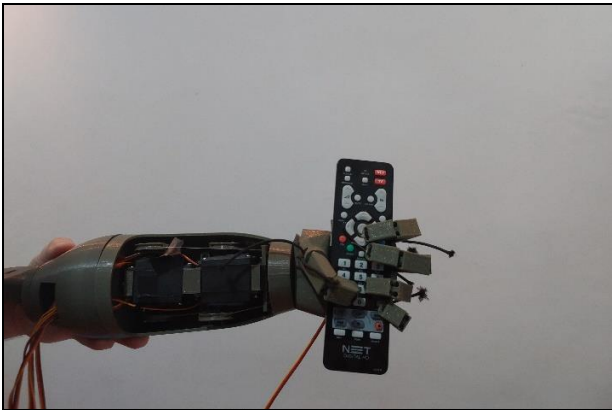


Fonte: Própria (2023).

Outro teste relevante consistiu na avaliação do PME ao segurar um controle remoto convencional de televisão, como indicado na Figura 14. Este teste foi conduzido com o objetivo de verificar a capacidade do dispositivo em manipular objetos presentes no cotidiano, indiferentemente de suas diferentes formas e tamanhos, proporcionando assim uma visão abrangente de sua adaptabilidade e versatilidade. A análise dos resultados deste teste complementa a avaliação geral da funcionalidade e usabilidade do protótipo em contextos habituais.



**Figura 14. PME segurando copo a partir das ações da mão do usuário**



Fonte: Própria (2023).

Após a realização dos testes planejados e executados meticulosamente, tornou-se evidente que o protótipo demonstra uma combinação equilibrada de força e precisão, o que o capacita a manter uma firmeza adequada ao segurar objetos de uso cotidiano. Esta constatação ressalta que o dispositivo de mão eletromecânica (PME) alcançou resultados que estão em consonância com as expectativas estabelecidas e os objetivos delineados desde o início do desenvolvimento do projeto.

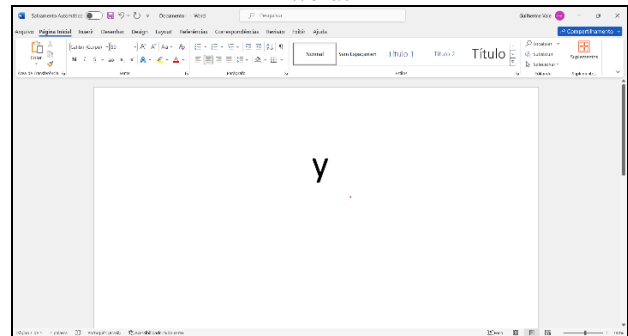
Para encerrar de maneira abrangente e conclusiva, procedeu-se com a realização de um último teste, visando explorar e avaliar a versatilidade do protótipo em questão. Neste contexto, o dispositivo foi empregado com o propósito específico de pressionar uma tecla localizada no teclado de um notebook, o qual encontrava-se exibindo o software de leitura e edição de texto *Word* da *Microsoft*, que se situava com um documento em branco aberto e disponível para alteração. Além disso, as configurações da fonte foram ajustadas para um tamanho notável, especificamente de 50 pontos.

A escolha dessa configuração específica foi deliberada, pois garantia uma visualização clara e imediata das ações realizadas pelo protótipo. Cada pressionamento de tecla resultava em uma exibição evidente na tela, possibilitando uma observação detalhada do funcionamento do dispositivo em tempo real.

Com o intuito de avaliar a precisão e a confiabilidade do protótipo, foi decidido, de forma aleatória, pressionar a tecla "Y" do teclado. Essa escolha arbitrária permitiu uma análise direcionada da capacidade do protótipo

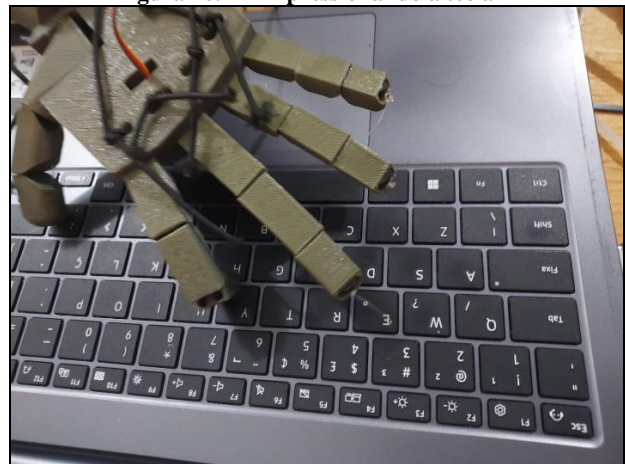
em executar uma tarefa específica com precisão. O objetivo era verificar se o dispositivo seria capaz de acionar uma única vez uma tecla designada com exatidão e consistência. Nas Figuras 15 e 16, é possível observar a letra "Y" digitada no programa e a tecla pressionada pelo PME.

**Figura 15. Resultado do pressionamento da tecla "Y" no Word.**



Fonte: Própria (2023).

**Figura 16. PME pressionando a tecla Y**



Fonte: Própria (2023).

A Figura 16 mostra o exato momento em que o dedo do protótipo pressiona a tecla em questão, para que a captura desse momento fosse possível, foi necessário fotografar ação da mão múltiplas vezes, pois, para esse teste, era desejado que a letra "Y" digitada uma única vez no software de edição de texto.

Através deste processo meticuloso, foi possível examinar minuciosamente a precisão do protótipo ao realizar uma operação tão fundamental quanto o pressionamento de uma tecla no teclado. Os resultados obtidos nesse teste final forneceram *insights* valiosos sobre o desempenho do dispositivo em situações práticas e cotidianas, contribuindo assim para



uma avaliação abrangente de sua funcionalidade e utilidade em ambientes de uso real.

Uma das limitações presentes no estado atual do protótipo está relacionada à sua abordagem de controle binário. Atualmente, a avaliação da posição dos dedos do usuário não inclui a leitura analógica da angulação dos dedos em relação à palma da mão. Como resultado, o Protótipo de Mão Eletromecânica funciona com apenas dois estados para cada dedo: estendido ou flexionado, sem considerar possíveis ângulos intermediários.

Essa limitação na capacidade de capturar nuances na posição dos dedos pode impactar a precisão e a naturalidade dos movimentos executados pelo protótipo. A ausência de uma leitura mais detalhada dos ângulos dos dedos pode limitar a gama de gestos que o dispositivo é capaz de reproduzir de forma fiel.

Para superar essa limitação e aprimorar a funcionalidade do protótipo, seria necessário implementar um sistema que permita a leitura e interpretação mais precisa dos ângulos de flexão dos dedos em relação à palma da mão. Isso proporcionaria uma maior variedade de gestos e movimentos possíveis, tornando a interação com o dispositivo mais intuitiva e próxima à experiência natural de movimentar a mão humana.

Para tal, considerou-se um próximo passo que envolveria o desenvolvimento de uma luva especial equipada com sensores flexíveis analógicos incorporados à sua palma. Esses sensores seriam conectados a uma placa de controle Arduino com comunicação sem fio (*wi-fi* ou *bluetooth*) utilizando de módulos compatíveis, ou, alternativamente, utilizar outra placa de controle com comunicação *wireless* nativa, a exemplo do Nodemcu (ESP).

A placa em questão seria responsável por transmitir os dados captados pelos sensores à placa controladora do protótipo. Ao receber esses dados em formato analógico, a placa controladora seria capaz de interpretá-los e executar as correspondentes gesticulações de forma precisa e eficiente. Esse avanço proposto ampliaria ainda mais a funcionalidade e a versatilidade do protótipo, possibilitando interações mais intuitivas e naturais com o ambiente ao redor. Desse modo, o sistema

dispensaria a necessidade de um computador com câmera acoplada para sua operação. Essa solução permitiria ao protótipo ganhar mobilidade, possibilitando seu uso em diferentes contextos e ambientes, sem depender de uma configuração fixa.

Além disso, um dos possíveis aprimoramentos para o controle do protótipo é a implantação de sensores de pressão nos dedos e na palma da mão. Esta inovação permitiria otimizar o controle de tração de cada motor através do controle de corrente disponibilizada, proporcionando ao usuário a capacidade de perceber a pressão que a mão exerce sobre um objeto segurado. Por exemplo, ao segurar um copo de vidro, é essencial ajustar precisamente a pressão aplicada para evitar possíveis quebras, trincas ou até mesmo escorregões acidentais. Com os sensores de pressão integrados, o protótipo poderia detectar a quantidade exata de força exercida pelos dedos e pela palma da mão, proporcionando ao usuário um feedback tátil ou visual em tempo real sobre a intensidade da pressão aplicada.

Essa funcionalidade não apenas aumentaria a segurança e a precisão ao manipular objetos delicados, mas também abriria possibilidades para uma interação mais sofisticada e adaptativa com o ambiente. Ademais, a capacidade de ajustar a pressão de forma dinâmica e sensível proporcionaria uma experiência de uso mais próxima da habilidade natural da mão humana em adaptar sua força de acordo com a situação.

#### 4. CONCLUSÃO

O objetivo geral da pesquisa no desenvolvimento do protótipo consistiu no projeto e construção de uma mão eletromecânica controlada via visão computacional. Após a realização dos testes, constatou-se que o comportamento do protótipo atendeu satisfatoriamente às expectativas estabelecidas.

No que diz respeito aos motores empregados, estes demonstraram ser adequados e eficientes com relação à sua potência e precisão. Outrora, a estrutura física do protótipo, construída com PLA, exibiu uma resistência adequada para suportar as demandas mecânicas as quais foi submetida.

Adicionalmente, tanto o controle por gestos, quanto a comunicação serial, revelaram um alto grau de refinamento, possibilitando a replicação de parte dos movimentos executados por uma mão humana.

Considerando os diversos aspectos explorados e as possibilidades de aprimoramento detalhadas ao longo deste processo de análise e expansão dos textos, é possível concluir que o desenvolvimento e a evolução do protótipo de mão eletromecânica representam um campo fértil para inovação e progresso tecnológico.

Através da revisão detalhada das limitações atuais do protótipo, como a abordagem de controle binário e a falta de sensores de pressão, foi possível identificar áreas-chave para melhorias significativas. A implementação de uma abordagem de controle mais refinada, que inclua a leitura dos ângulos dos dedos e a integração de sensores de pressão, tem o potencial não apenas de aumentar a precisão e a versatilidade do protótipo, mas também de proporcionar uma experiência de usuário mais natural e intuitiva.

Além disso, a introdução de tecnologias como sensores de pressão não apenas aprimoraria a funcionalidade do protótipo em termos de controle de força e precisão, mas também abriria novas possibilidades para interações mais complexas e adaptativas com o ambiente. Isso inclui a capacidade de ajustar dinamicamente a pressão exercida durante a manipulação de objetos delicados, aumentando a segurança e reduzindo o risco de danos.

Outro ponto importante a ser considerado é o potencial impacto social e econômico dessas melhorias. Um protótipo de mão eletromecânica mais avançado e sofisticado poderia beneficiar uma ampla gama de usuários, incluindo pessoas com deficiências físicas, trabalhadores em ambientes industriais e até mesmo profissionais da área da saúde. Ao aumentar a eficiência e a segurança na manipulação de objetos, essas tecnologias têm o potencial de melhorar significativamente a qualidade de vida e a produtividade de diversos grupos.

Em suma, a pesquisa realizada foi exitosa, pois, além dos ganhos teóricos e empíricos, provenientes de seu desenvolvimento, o

protótipo, em sua totalidade, obteve desempenho superior ao esperado nos testes realizados.

## REFERÊNCIAS

- Raheja, J. L.; Shyam, R.; Kumar, U.; Prasad, P. B. (2010). Real-Time Robotic Hand Control Using Hand Gestures. *International Conference on Machine Learning and Computing*, Bangalore, India, pp. 12-16, doi: 10.1109/ICMLC.2010.12.
- Gonzalez, R. C.; Woods R. E. (2010). *Processamento Digital de Imagens*. São Paulo: Editora Edgard Blücher Ltda. SP.
- MediaPipe - Google Research Team. (2023). Disponível em: <https://google.github.io/mediapipe/solutions/hands.html>.
- Nise, N. S. (2012). *Engenharia de Sistemas de Controle*, 6ª Edição, LTC.
- Lathi, B. P. (2006). *Sinais e Sistemas Lineares*. 2ª Edição, Porto Alegre: Editora Bookman.