

DESENVOLVIMENTO DE UMA CÂMERA AUTÔNOMA DE RASTREAMENTO DE OBJETOS

**Ana Beatriz Silva Carvalho¹; Davi Pereira da Silva²; Pedro Renato Araújo Santos³
;Anderson Pablo Freitas Evangelista⁴**

RESUMO

Nos últimos anos, temos presenciado avanços significativos nos campos da visão computacional e da robótica, resultando no desenvolvimento de sistemas inteligentes e autônomos. Portanto, este projeto de pesquisa propõe o desenvolvimento de uma câmera autônoma com a capacidade de rastrear e seguir objetos em movimento. Utilizando técnicas de visão computacional e controle robótico, o objetivo é explorar e integrar algoritmos de detecção, rastreamento e controle para criar um sistema altamente eficiente e preciso, capaz de acompanhar objetos em tempo real.

Palavras-chave: Dispositivo Autônomo; Reconhecimento de Padrões; Visão computacional.

Órgão financiador: IFMA.

1 Discente do curso de Automação Industrial do IFMA – Campus Açailândia; E-mail: carvalho.ana@acad.ifma.edu.br

2 Discente do curso de Automação Industrial do IFMA – Campus Açailândia; E-mail: davi.p@acad.ifma.edu.br

3 Discente do curso de Automação Industrial do IFMA – Campus Açailândia; E-mail: renatoa@acad.ifma.edu.br

4 Docente do curso de Automação Industrial IFMA – Campus Açailândia; E-mail: anderson.evangelista@ifma.edu.br

1 INTRODUÇÃO

A robótica e a visão computacional têm-se desenvolvido rapidamente, permitindo a criação de sistemas autônomos para diversas áreas, como segurança, logística e indústria. Esses sistemas combinam análise visual com controle robótico, resultando em soluções mais eficientes e precisas. A integração de inteligência computacional e robótica tem permitido o avanço de tecnologias que realizam tarefas de forma autônoma e eficaz. Nos últimos anos, temos presenciado avanços significativos nos campos da inteligência computacional e da robótica, resultando no desenvolvimento de sistemas inteligentes e autônomos (Lobo, 2018; Teixeira, 2016; Gudwin, 2005). De acordo com Gudwin (2005), embora essas áreas sejam distintas e tenham métodos e objetivos próprios, os seres humanos têm sido criativos ao integrar abordagens desses dois domínios.

A visão computacional é uma área central nesse desenvolvimento, possibilitando que máquinas interpretem imagens e vídeos para entender o ambiente ao seu redor. Por meio de algoritmos de detecção e rastreamento, dispositivos robóticos conseguem identificar objetos em movimento e tomar decisões com base nessas informações. Dentro do amplo espectro da Inteligência Artificial (IA), existe a visão computacional, uma subárea crucial que capacita sistemas computacionais a interpretar e compreender informações visuais. Por meio do uso de técnicas e algoritmos especializados, a visão computacional permite que os computadores analisem imagens e vídeos para extrair informações úteis, tais como reconhecimento de objetos, detecção de padrões, estimativa de profundidade e interpretação de movimentos (Milano; Honorato, 2010; Neves; Neto; Gonzaga, 2012). Assim, podemos aplicar metodologias da visão computacional para dotar dispositivos robóticos da capacidade de compreender e reagir a imagens, tornando-os autônomos em contextos específicos (Amaral; Gasparotto, 2021).

Este projeto tem como objetivo desenvolver uma câmera autônoma capaz de rastrear e seguir objetos em movimento, utilizando técnicas de visão computacional e controle robótico. O sistema proposto visa aplicar esses algoritmos em tempo real para criar soluções práticas em áreas como segurança, monitoramento e automação, demonstrando o potencial dos sistemas autônomos para melhorar a eficiência em diferentes contextos. Pretende-se explorar e integrar algoritmos de detecção, rastreamento e controle para criar um sistema eficiente e preciso capaz de acompanhar objetos em tempo real.

2 METODOLOGIA

O desenvolvimento da câmera autônoma seguiu etapas desde a definição dos requisitos até os testes finais.

- **Definição de Requisitos:** Primeiro, foram estabelecidos os objetivos do projeto e os requisitos do sistema. Foi definido quais objetos seriam rastreados, a precisão necessária e as condições em que a câmera iria operar. Também foram escolhidos os algoritmos de detecção e rastreamento mais adequados para funcionar em tempo real.
- **Seleção de Componentes:** Depois, foram escolhidos os componentes eletrônicos, como câmera, sensores e processador, considerando fatores como resolução, velocidade e consumo de energia. Paralelamente, foi planejada a estrutura mecânica e eletrônica para que a câmera pudesse se mover e rastrear objetos corretamente.
- **Integração de Hardware e Software:** Em seguida, iniciou-se a montagem do sistema. A câmera e os sensores foram conectados e alimentados. No software, foram implementados os algoritmos de visão computacional usando Python, OpenCV e Arduino para detectar e seguir os objetos automaticamente.
- **Testes e Ajustes:** Por fim, foram realizados testes para verificar se tudo funcionava corretamente. Primeiro, cada componente foi testado individualmente, depois o sistema como um todo. Ajustes foram feitos quando necessário. O sistema foi testado em diferentes situações, e os resultados foram registrados para futuras melhorias.

3 RESULTADOS E DISCUSSÃO

Com a conclusão do processo, o projeto resultou no desenvolvimento de uma câmera autônoma de rastreamento de objetos.

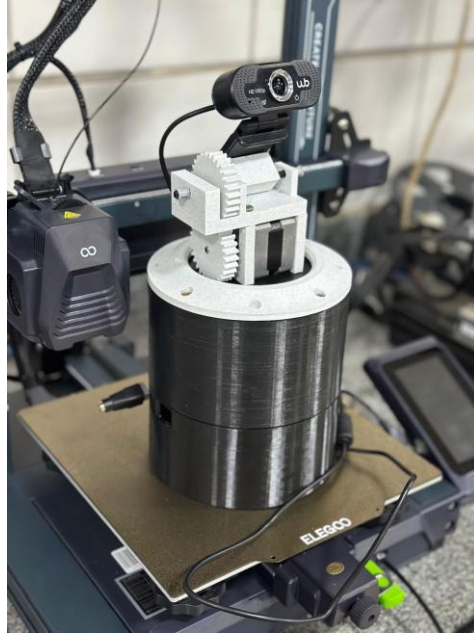
Estrutura Mecânica e Eletrônica

O projeto foi concluído com a integração de todos os aspectos mecânicos, eletrônicos e de programação, resultando em um sistema funcional. Os motores de passo NEMA 17 foram incorporados à estrutura e controlados por meio de dois drivers A4988, cada um montado em suas respectivas placas de expansão. Esse conjunto atendeu às exigências de torque e precisão, além de facilitar as conexões entre os componentes.

Do ponto de vista físico, o mecanismo utilizado para movimentar os eixos manteve-se inalterado, mas houve melhorias na organização do sistema eletrônico. Foi adicionado um segundo compartimento, localizado abaixo do motor do eixo X, com a função de abrigar a maior parte dos componentes eletrônicos sem interferir no funcionamento mecânico. Nesse compartimento foram instalados o Arduino Uno e os dois drivers A4988, juntamente com a fiação necessária. Além disso, foram disponibilizados conectores externos para a fonte de 12V, para o cabo USB do Arduino e para o cabo USB da câmera.

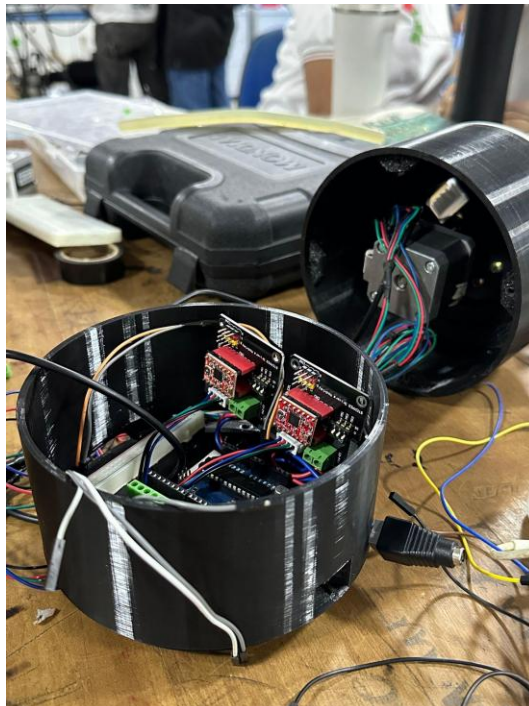
A Figura 1 mostra a aparência final do protótipo posicionado sob uma impressora 3D, destacando sua portabilidade. Já a Figura 2 apresenta a vista lateral do robô com os compartimentos eletrônicos expostos, evidenciando a organização interna. A Figura 3 ilustra a disposição dos componentes no segundo compartimento, onde foi concentrada a parte lógica do sistema. Por fim, a Figura 4 mostra a parte inferior do primeiro compartimento, evidenciando a instalação do motor do eixo X. Essas mudanças estruturais resultaram em um sistema mais organizado e seguro, facilitando ajustes e manutenções. A centralização da parte eletrônica em um único compartimento reduziu riscos de mau contato e falhas, além de simplificar o transporte do protótipo.

Figura 1 - Aparência final do robô sob uma impressora 3D



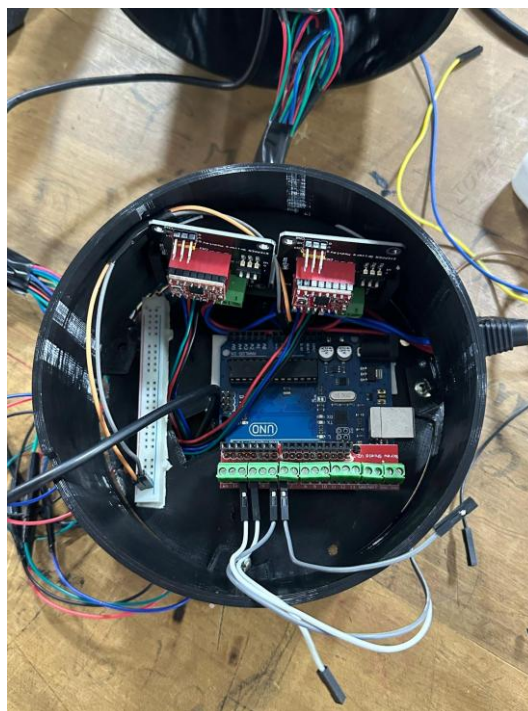
Fonte: Autores, 2025.

Figura 2 - Vista do robô com os compartimentos expostos



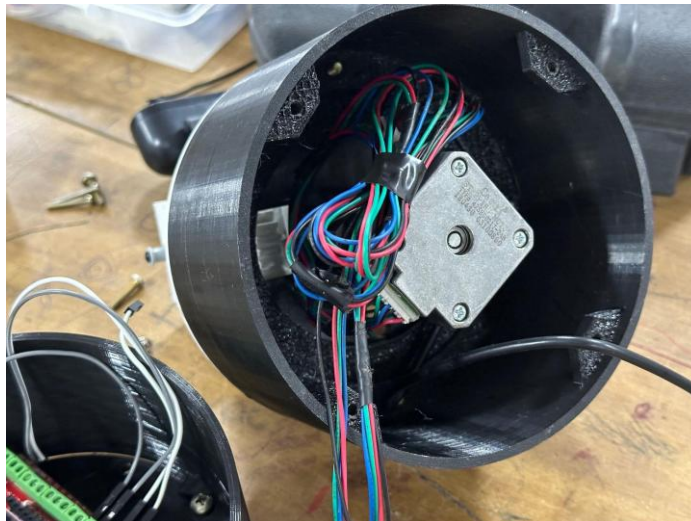
Fonte: Autores, 2025.

Figura 3 – Vista superior do segundo compartimento



Fonte: Autores, 2025.

Figura 4 – Vista inferior do primeiro compartimento



Controle e Programação

O sistema de controle foi implementado em Python, utilizando o Visual Studio Code, sendo responsável pela visão computacional e pelos cálculos de rastreamento. Foram configurados dois modos de operação: rastreamento de mão, por meio da biblioteca MediaPipe, e rastreamento de rosto, utilizando HaarCascade. Para cada quadro capturado pela câmera, o programa identifica a posição central do objeto de interesse, representada pelas variáveis (cx, cy) , e calcula o erro em relação ao centro da tela, definido por (cx_img, cy_img) . A diferença entre essas coordenadas gera os erros em X e Y, que são processados por controladores PID

independentes. Esses controladores fornecem as velocidades de correção (v_x , v_y), que serão aplicadas aos motores para ajustar a posição do sistema.

Algoritmo 1: Rastreamento de Mão e Rosto em Python

```
Iniciar comunicação com Arduino
Iniciar câmera
Definir centro da tela ( $cx\_img, cy\_img$ )
Configurar PID para X e Y
Criar janela gráfica com botões e gráficos
while programa rodando do
    Capturar imagem da câmera
    if modo == "mão" then
        Detectar mão com MediaPipe
        if mão detectada then
            Calcular centro da mão ( $cx, cy$ )
            Calcular erro ( $cx\_img - cx, cy\_img - cy$ )
            Aplicar PID → gerar ( $v_x, v_y$ )
            Enviar ( $v_x, v_y$ ) ao Arduino
        else
            Enviar (0, 0)
        end
    else
        Detectar rosto com HaarCascade
        if rosto detectado then
            Calcular centro do rosto ( $cx, cy$ )
            Calcular erro ( $cx\_img - cx, cy\_img - cy$ )
            Aplicar PID → gerar ( $v_x, v_y$ )
            Enviar ( $v_x, v_y$ ) ao Arduino
        else
            Enviar (0, 0)
        end
    end
    Atualizar gráficos de erro ( $err_x, err_y$ ) e de velocidade ( $v_x, v_y$ )
end
Encerrar programa
```

Comunicação e Interface

As informações de velocidade (v_x , v_y) calculadas pelo Python são enviadas via comunicação serial para o Arduino, que interpreta os comandos e ajusta gradualmente a velocidade dos motores, garantindo movimentos suaves e evitando variações bruscas que possam comprometer a estabilidade do sistema. Além disso, o software inclui uma interface gráfica desenvolvida em Tkinter, permitindo ao usuário selecionar o modo de rastreamento e acompanhar o desempenho do protótipo em tempo real. Dois gráficos são apresentados na interface: o primeiro exhibe a evolução do erro de rastreamento nos eixos X e Y (err_x , err_y),

mostrando a precisão do acompanhamento do objeto; o segundo apresenta as velocidades de correção (v_x , v_y) aplicadas aos motores, evidenciando como o sistema reage para minimizar os erros.

```
Algoritmo 2: Controle de motores no Arduino  
Definir pinos de direção, passo e enable para os dois motores  
Configurar comunicação serial  
Habilitar os drivers dos motores  
Definir velocidade máxima permitida  
while programa rodando do  
  if mensagem na serial then  
    Ler string no formato "vx,vy"  
    Separar valores de velocidade de X e Y  
    Aplicar inversões se necessário  
    Guardar como alvo de velocidade  
  end  
  Calcular tempo decorrido desde a última atualização (dt)  
  Para cada motor (X e Y): Comparar velocidade atual com alvo  
  Ajustar suavemente (limitando variação por dt)  
  Atualizar velocidade atual  
  Definir velocidades ajustadas nos motores  
  Executar movimento contínuo com runSpeed()  
end
```

Desempenho Final

Com a integração entre hardware, software e controle, o protótipo demonstrou eficiência no rastreamento de objetos. Ele foi capaz de acompanhar tanto mãos quanto rostos em movimento, realizando ajustes suaves e mantendo estabilidade mesmo diante de variações rápidas de posição.

Os resultados confirmam que os objetivos do projeto foram atingidos, com um sistema funcional, organizado e confiável, pronto para ser utilizado em aplicações de monitoramento e automação.

4 CONCLUSÃO

O projeto foi concluído com sucesso, integrando de forma funcional os aspectos mecânicos, eletrônicos e de controle. Os motores NEMA 17 e os drivers A4988 garantiram precisão e estabilidade nos movimentos dos eixos, enquanto a organização dos componentes eletrônicos facilitou ajustes, manutenção e operação do protótipo.

O sistema de controle, combinando Python para visão computacional e cálculo do erro, e Arduino para execução dos comandos nos motores, demonstrou eficiência. O uso de controladores PID permitiu rastreamento contínuo e preciso de mão ou rosto, e a interface gráfica forneceu feedback em tempo real, auxiliando na análise e ajuste do desempenho.

O desenvolvimento do projeto proporcionou experiência prática relevante para a área de automação industrial, permitindo aplicar conhecimentos em mecânica, eletrônica, programação e controle, além de lidar com problemas reais de integração de sistemas. A execução do protótipo consolidou habilidades técnicas e mostrou a importância da experimentação e do planejamento no desenvolvimento de sistemas automatizados.

A experiência adquirida reforça a importância do trabalho estruturado e integrado em projetos de automação, contribuindo para a formação profissional e a compreensão prática de sistemas mecatrônicos aplicados à indústria.

AGRADECIMENTOS

Agradeço ao IFMA pelo apoio financeiro essencial e ao IFMA Campus Açailândia pela infraestrutura disponibilizada, que foram fundamentais para a realização deste projeto. Registro também meus sinceros agradecimentos a todas as pessoas que, de alguma forma, contribuíram para o seu êxito.

REFERÊNCIAS

AMARAL, V.; GASPAROTTO, C. Automação e tecnologia na indústria 4.0: desafios e oportunidades. **Revista de Engenharia e Tecnologia**, v. 9, n. 1, p. 45–56, 2021.

GUDWIN, R. Novas fronteiras na inteligência artificial e na robótica. *In: dincon'2005*. [S.l.: s.n.], 2005. p. 01–18.

LOBO, L. Inteligência artificial, o futuro da medicina e a educação médica. **Revista Brasileira de Educação Médica**, v. 42, p. 3–8, 2018.

MILANO, J. D.; HONORATO, L. Visão computacional aplicada: uma abordagem prática. **Revista Brasileira de Tecnologia**, v. 8, n. 2, p. 30–40, 2010.

NEVES, L.; NETO, H.; GONZAGA, A. **Avanços em visão computacional**. Curitiba: Omnipax, 2012. v. 27.

TEIXEIRA, J. D. F. **O cérebro e o robô: inteligência artificial, biotecnologia e a nova ética**. [S.l.: s.n.]: Pia Sociedade de São Paulo-Editora Paulus, 2016.