

## **DESENVOLVIMENTO DE ROBÔS DE RESGATE**

Beatriz Gomes Medeiros<sup>1</sup>; Fernanda Gomes Medeiros<sup>2</sup>; Iago Bruno Gonçalves Rocha<sup>3</sup>; Isabelly Moreira Machado de Souza<sup>4</sup>; Samuel Marques<sup>5</sup>; Elias José de Rezende Freitas<sup>6</sup>; Luiz Olmes Carvalho<sup>7</sup>

1 Beatriz Gomes Medeiros, Bolsista (IFMG), Curso Técnico em Automação Industrial, IFMG, Campus Avançado de Itabirito – MG

2 Fernanda Gomes Medeiros, Bolsista (IFMG), Curso Técnico em Automação Industrial, IFMG, Campus Avançado de Itabirito – MG

3 Iago Bruno G. Rocha, Voluntário (IFMG), Curso Técnico em Automação Industrial, IFMG, Campus Avançado de Itabirito – MG

4 Isabelly M. Machado de Souza, Bolsista (IFMG), Curso Técnico em Automação Industrial, IFMG, Campus Avançado de Itabirito – MG

5 Samuel Marques, Voluntário (IFMG), Curso Técnico em Automação Industrial, IFMG, Campus Avançado de Itabirito – MG

6 Coorientador Elias José de Rezende Freitas: Pesquisador do IFMG, Campus Ibité – MG

7 Orientador Luiz Olmes Carvalho: Pesquisador do IFMG, Campus Avançado de Itabirito – MG

### **RESUMO**

A área da robótica tem evoluído significativamente nos últimos anos. Prova disso é o emprego, cada vez mais difundido, de robôs que auxiliam ou mesmo substituem a presença humana em diversos contextos. Uma das tarefas nas quais o emprego de robôs pode auxiliar a sociedade relaciona-se às situações de resgate. Robôs de resgate são projetados para dar suporte em uma possível calamidade ou desastre, sendo capazes de mapear espaços e identificar potenciais sobreviventes. Motivado por este cenário, este projeto de pesquisa visa a construção de um protótipo de robô terrestre capaz de se guiar por uma trajetória. Como principal resultado obtido, destaca-se o protótipo de um robô seguidor de trajetória formado por quatro motores e sensores de deslocamento em um ambiente simulado.

### **INTRODUÇÃO**

Em um contexto histórico, no século XVIII, no auge da Revolução Industrial, a necessidade da inserção de máquinas possibilitou um melhor entendimento de sua importância para a humanidade e a partir daí houve uma notória evolução dos meios disponíveis para esse fim, que hoje resultam em robôs que se utilizam de mecanismos autônomos e inteligência artificial (SCHWAB, 2019).

O assunto que tange a aplicabilidade de robôs em um âmbito mais abrangente permite refletir sobre os recursos tecnológicos disponíveis e amplamente utilizados em prol do ser humano (OLIVEIRA et al., 2017). A real necessidade de se conseguir meios de explorar ambientes não permissivos ao ser humano levou a criação e adequação de tecnologias que trabalham com essa finalidade.

O emprego de tecnologias, além de extremamente relevante, permite que sejam agregadas prática e teorias que são comumente utilizadas em metodologias experimentais. A possibilidade de trabalhar em áreas de risco iminente, como em casos de resgate, por intermédio de robôs, garante segurança física e comodidade, além de conhecimento a ser adquirido e aprimorado.

A importância de estudos que visem avanços tecnológicos e sua aplicabilidade em diversos contextos são positivos na busca por meios que facilitem o “trabalho braçal” dando autonomia aos seus usuários (CHUNG et al., 2017; MIHELJ et al., 2019).

Este trabalho teve a finalidade de desenvolver um protótipo de robô idealizado para realizar um diagnóstico descritivo de rotas pré-definidas em um ambiente simulado. Para tal fez-se a utilização de kits acadêmicos conhecidos como LEGO® MINDSTORMS® Education, além do framework ev3dev (EV3DDEV, 2021) e a linguagem de programação Python.

### **METODOLOGIA**

A metodologia que foi utilizada para o desenvolvimento do presente trabalho é embasada na experimentação científica, importante instrumento na coleta de dados e de informações na obtenção dos resultados esperados. O método é indutivo e qualitativo, ou seja, conclusões são geradas a partir da experimentação e são vistas como verdade universal e as informações obtidas não podem ser quantificáveis (COSTA, 2007). O protótipo inicial busca descrever rotas pré-definidas em um ambiente simulado, para servir de suporte em uma possível situação de resgate.

## Ambiente Simulado

Para simular o ambiente de uma possível situação de resgate, foi utilizada uma manta de borracha com as dimensões 100 cm x 157 cm. Nela foram feitas marcações em branco e uma rosa dos ventos, como pode ser visto na Figura 1. Para ajudar na orientação do robô, alguns vértices foram definidos e numerados de 1 a 15, como pode ser visto na Figura 2.

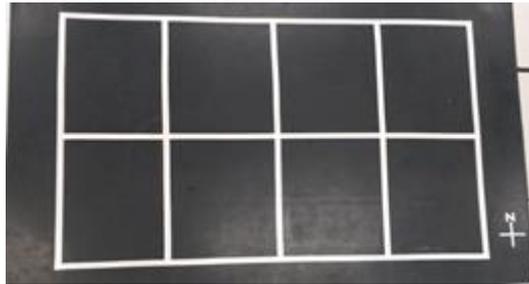


Figura 1 - Ambiente simulado.

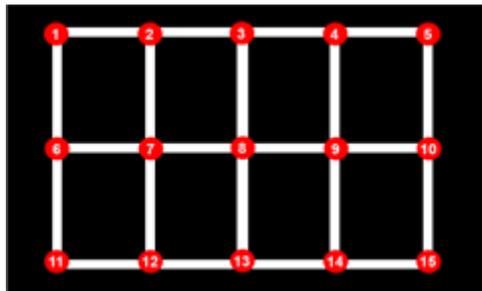


Figura 2 - Vértices de orientação do ambiente simulado.

## Sensor de Cor e Motores

O sensor de cor é um dos componentes mais relevantes deste trabalho, pois é utilizado para distinguir as cores preto e branco. Para isso, é utilizada a função “reflected\_light\_intensity”, que faz com que a intensidade de luz captada pelo sensor volte em uma escala que varia de 0 a 100. Após realizar alguns testes, abrindo uma margem para erro, foi constatado que a cor preta é tudo aquilo que reflete um valor menor que 20 nessa escala. Foram utilizados quatro sensores, ligados ao módulo EV3. Além disso, para o deslocamento do robô foram utilizados quatro motores do tipo Large Motor. Os motores foram nomeados conforme as saídas do EV3.

## CÓDIGO FONTE

O código-fonte, escrito na linguagem Python, é a parte mais importante desse estudo. O código se inicia incluindo todas as bibliotecas necessárias, declarando motores, sensores, e setando valores de variáveis como “sp\_max” (velocidade máxima), “cont” (contador iniciado com 0 (zero)) e “black” (valor máximo que a leitura do Sensor de Cor pode assumir para que objeto seja considerado preto), além de informar o vértice de partida e o de destino. Com todas as variáveis setadas, algumas funções auxiliares de movimentação são definidas, sendo elas:

## Função stop()

Na função “stop()” todos os motores são parados com a função “stop” do EV3.

### Função ture\_1()

A função “ture\_1” é utilizada para girar o robô para a direita. Nela é utilizada a função “run\_direct” do EV3. Os motores A e C recebem velocidade máxima e os motores B e D velocidade nula. Observando a Figura 3, é possível notar que essa disposição de acionamento dos motores faz com que o robô vire à direita.

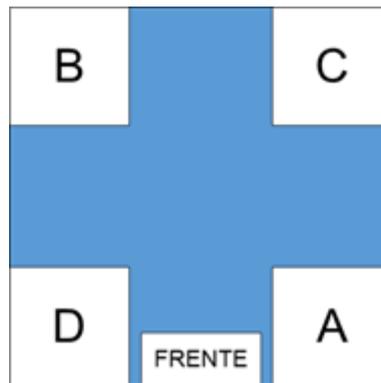


Figura 3 - Disposição dos motores

### Função ture\_2()

De modo análogo à função “ture\_1”, a função “ture\_2” é utilizada para girar o robô para a esquerda. Nela também é utilizada a função “run\_direct”. Os motores B e D recebem velocidade máxima e os motores A e C velocidade nula.

### Função line()

A função “line()” é a responsável por manter o robô seguindo uma linha reta. Para isso são utilizados dois sensores de cor, um de cada lado da linha. Ela começa testando se o robô se encontra em um cruzamento; isso serve para não gerar conflito com as funções de rota. O robô só faz o controle de seguir linha se não estiver em um cruzamento. Caso o sensor 2 identifique um valor abaixo de black, variável com o valor 20, o robô vira para a direita, com a função “ture\_1”, até que o sensor volte a identificar um valor maior que black. O mesmo vale para o sensor 3, só que nesse caso o robô vira para a esquerda, com a função “ture\_2”. Se os dois sensores estiverem visualizando preto, o robô segue com as velocidades dos quatro motores iguais. A função “line()” é dada por:

```
def line():
    if(cruzamento == OFF):
        if(sensor_color2.reflected_light_intensity > black):
            ture_1()
        elif(sensor_color3.reflected_light_intensity > black):
            ture_2()
        else:
            ma.run_direct(duty_cycle_sp = sp_front)
            mb.run_direct(duty_cycle_sp = sp_front)
            mc.run_direct(duty_cycle_sp = sp_front)
            md.run_direct(duty_cycle_sp = sp_front)
```

### Função front()

A função “front()” é a responsável por mover o robô para frente em caso de cruzamento e chamar a função “line()” se o robô não estiver em um cruzamento. Para isso é utilizada a função “run\_direct”, sendo que, nela, todos os motores recebem o mesmo valor de velocidade. A função “front()” pode ser vista a seguir:

```
def front():  
    if(cruzamento == ON):  
        ma.run_direct(duty_cycle_sp = sp_front)  
        mb.run_direct(duty_cycle_sp = sp_front)  
        mc.run_direct(duty_cycle_sp = sp_front)  
        md.run_direct(duty_cycle_sp = sp_front)  
    else:  
        line()
```

### Funções de rota

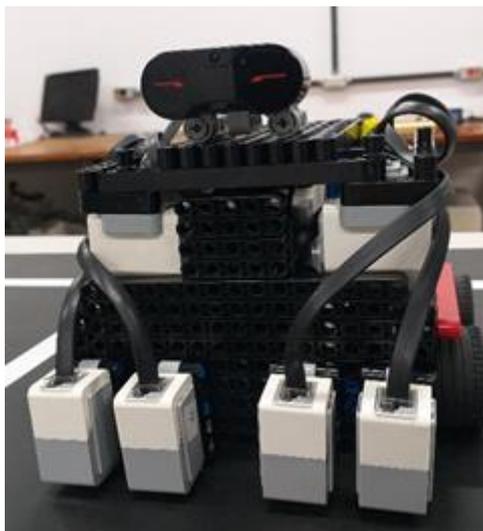
Neste primeiro protótipo, existe uma função para cada rota existente. O controle das rotas é feito com contadores pré-definidos, que contam cruzamentos, e através de estudos praticados pelos autores desse projeto, eles podem ser exemplificados através de:

**Rota\_1\_8: cont = 3 + ture\_1() + cont = 1.**

O nome descreve a rota desejada, como, por exemplo a rota\_1\_8 é a rota de 1 a 8. A função importa as variáveis “cont” e “cruzamento” e é formada por pequenos loops. Por exemplo, a rota\_1\_8 é formada por um “cont = 3”, um “ture\_1()” e um “cont = 1”, ou seja, o primeiro loop roda enquanto o “cont” é menor que 3, o segundo loop gira o robô para a direita até atingir a linha e o terceiro loop testa se “cont” é igual a quatro, se for, a função “stop()” é chamada e o robô pronuncia a mensagem “*Hello, I am your rescue assistant!*”.

### RESULTADOS E DISCUSSÕES

Na busca pelos resultados que foram obtidos, foi desenvolvido um protótipo com quatro motores, quatro sensores de cor e um EV3, além de outras peças, sendo todos componentes LEGO®. Logo após foi realizada a instalação do framework ev3dev no EV3. Todos os componentes foram cedidos pela Equipe EPIIBOTS. O protótipo final pode ser visto na Figura 4.



**Figura 4 - Protótipo final**

O ambiente simulado foi desenvolvido utilizando uma manta de borracha de 100 cm x 157 cm e fita isolante na cor branca, ambas foram cedidas pela EPIIBOTS.

Um algoritmo responsável por traçar rotas foi desenvolvido e implementado em Python. O robô descreve positivamente todas as rotas possíveis e emite uma mensagem sonora ao chegar ao fim da trajetória. A principal dificuldade deste trabalho foi a criação das funções de mapeamento de rotas. Considerando a existência entre diversas possíveis rotas entre duas referências inicial e final, o mapeamento realizado considera o caminho mais direto, porém, não necessariamente a rota mais otimizada, o que abre uma possibilidade para investigações futuras.

Em se tratando de um projeto de nível médio/técnico, nota-se que os objetivos esperados foram atingidos e os resultados mais relevantes sob a ótica discente foram o conhecimento mais aprofundado da linguagem Python e noções de leitura e lógica envolvendo sensores e motores ampliadas.

## **CONCLUSÕES**

Um dos objetivos deste trabalho foi desenvolver um robô que possa ser igualmente construído com possibilidade de melhoria por qualquer pessoa com o mínimo entendimento específico para ser usado em situações de resgate. É importante destacar que o robô é um protótipo que atua em um ambiente simulado que deve ser aprimorado caso seja utilizado em uma situação de resgate real. Além disso, pode ser necessário a mudança da estrutura considerando o tipo de relevo do terreno onde será empregado.

## **Proposta de Continuidade**

Mesmo ciente que o protótipo alcançou os objetivos esperados e obteve resultados satisfatórios, é necessário mencionar que adequações podem ser adicionadas na intenção de um total aproveitamento da ideia, dentre eles, integração de uma câmera para transmissão em tempo real e armazenamento de imagens; integração de um transmissor de áudio para comunicação em tempo real; personalização da interface.

## **REFERÊNCIAS BIBLIOGRÁFICAS:**

CHUNG, W. K.; YANG, Y.; CUI, N.; QIAN, H.; XU, Y. Design of a rescue robot with a wearable suit augmenting high payloads rescue missions. International Conference on Advanced Robotics and Mechatronics. 2017.

COSTA, W. R. Metodologia Científica. Paracambi. 2007.

EV3DEV. ev3dev is your EV3 re-imagined. 2021. Disponível em: <<https://www.ev3dev.org/>>. Acesso: 15/06/2021.

MIHELJ, M.; BAJD, T.; UDE, A.; LENARCIC, J.; STANOVNIK, A.; MUNIH, M.; REJC, J.; SLAJPAH, S. Robotics. Springer. 2019.

OLIVEIRA, L. B. R.; LEROUX, E.; FELIZARDO, K. R.; OQUENDO, F.; NAKAGAWA, E. Y. ArchSORS: A Software Process for Designing Software Architectures of Service Oriented Robotic Systems. The Computer Journal, 60 (9), 2017.

SCHWAB, K. A quarta Revolução Industrial. Edipro. 2019.