

## **Título: SiRoVIA - Sistema Robótico Virtual com Inteligência Artificial: implementação de Lógica Fuzzy**

**Lucas Calefe Pereira**

Instituto Federal(IFSP), São Carlos, SP, Brasil

**Fabríciu Alarcão Veiga Benini**

Instituto Federal(IFSP), São Carlos, SP, Brasil

**Resumo:** A cada dia mais a robótica está em ascensão e todo esse avanço possibilitou a autonomia em tecnologias nesse novo contexto da Indústria 4.0. Momento de inovação que engloba diversas tecnologias e permite uma maior aproximação do homem com a máquina. Nesse contexto, o projeto intitulado “SiRoVIA - Sistema Robótico Virtual com Inteligência Artificial: implementação de Lógica Fuzzy” pode contribuir nesse novo momento da indústria. O projeto SiRoVIA, por se tratar de inteligência artificial, pode favorecer uma grande gama de sistemas de produção, assim como a indústria automotiva na concepção de veículos autônomos. O propósito deste projeto é usar a Lógica Fuzzy como principal base para tomada de decisões rápidas. O objetivo é guiar, dentro de um ambiente virtual, uma Unidade Robótica Móvel (URM), a tomada de decisões autônomas para desvio de obstáculos. Ao se encontrar com algum obstáculo à sua frente, a URM utiliza o sistema de decisões Fuzzy para contorná-lo utilizando a leitura dos sensores ultrassônicos distribuídos ao redor do robô, o sistema de sensores frontais são utilizados para identificar um obstáculo à frente, já os sensores laterais e traseiros são utilizados como anti-colisão e decisão de rota. No ambiente de simulação do robô existem obstáculos fixos e móveis que interferem na sua rota de movimento, a URM foi elaborada para funcionar tanto em um ambiente virtual desenvolvido especificamente para esse fim, quanto para funcionar na vida real.

**Palavras-chave:** Ambiente Virtual. Simulação. CoppelliaSim.

**Abstract:**

Robotics is increasingly on the rise and all this advance has enabled autonomy in technologies in this new context of Industry 4.0. Moment of innovation that encompasses technologies and allows for a closer relationship between man and machine. In this context, the project entitled "SiRoVIA - Virtual Robotic System with Artificial Intelligence: implementation of Fuzzy Logic" can contribute at this moment in the industry. The SiRoVIA project, as it is artificial intelligence, can favor a wide range of production systems, as well as the automotive industry in the design of autonomous vehicles. The purpose of this project aims to use Fuzzy Logic as the main basis for quick decision making. The objective is to guide a Mobile Robotic Unit within a virtual environment to make autonomous decisions to avoid obstacles. When encountering an obstacle in front of him, he uses the Fuzzy decision system to get around it using the reading of the ultrasonic sensors distributed throughout the robot, the system of front sensors are used to identify an obstacle ahead, the side and rear sensors are used as anti-collision and route choices. In the robot simulation environment there are fixed and mobile obstacles that interfere with its movement path, the URM was designed to work so much in a virtual environment that it was specifically designed for that purpose, how much to work in real life.

**Keywords:** *Virtual environment. Simulation. CoppelliaSim.*

## APRESENTAÇÃO DO TRABALHO

A lógica Fuzzy passou a ter influência no meio científico no século XX através da publicação do artigo “Fuzzy Sets” (RIGNEL; CHENCI e LUCAS, 2011). “A lógica Fuzzy é uma técnica de inteligência artificial baseada na teoria dos conjuntos Fuzzy (nebulosos)” (ROGES et al, 2017, p. 99), diferente da lógica Clássica que tem como valores apenas verdadeiro e falso, a lógica Fuzzy tem uma gama de opções muito maior, baseado em informações onde não existe apenas respostas extremas como verdadeiro e falso, 0 e 1, mas sim um amplo grau para escolher uma melhor opção de desvio (AGUADO; CATANHEDE, 2010). Utilizando um sensor ultrassônico, modelo HC-SR04, a URM, ao se aproximar de um obstáculo, emite ondas sônicas que chegam ao objeto e voltam para o sensor, fazendo com que o robô tome outra decisão de rota por conta própria. A implementação da lógica Fuzzy foi feita semelhante ao de Benini (2016), só que desta vez para o ambiente CoppeliaSim Edu e implementado em Python, por ser uma linguagem de programação gratuita e de simples aprendizado, além de ser suportada por microcontroladores tal como ESP32 (FERREIRA e BENINI, 2020).

A cada dia mais a robótica está em ascensão e todo esse avanço possibilitou a autonomia em tecnologias nesse novo contexto da Indústria 4.0. Momento de inovação que engloba diversas tecnologias e permite uma maior aproximação do homem com a máquina (FERREIRA e BENINI, 2020). Nesse contexto, o projeto intitulado “SiRoVIA - Sistema Robótico Virtual com Inteligência Artificial: implementação de lógica Fuzzy” pode contribuir nesse novo momento da indústria, onde fábricas precisam ser automatizadas com uso da inteligência artificial (IA), permitindo que máquinas tenham respostas cada vez mais rápidas. O projeto SiRoVIA, por se tratar de inteligência artificial, pode favorecer uma grande gama de sistemas de produção, assim como a indústria automotiva na concepção de veículos autônomos.

O desenho da URM foi projetado utilizando o SketchUp, um software de criação de modelagem 3D, sendo esse um programa de trabalho e aprendizado fácil e versátil, além de ser uma plataforma gratuita (LOSS e BENINI, 2018).

Também foi utilizado o software CoppeliaSim Edu, novo nome dado ao antigo V-REP, que é uma plataforma de simulação gratuita com desenvolvimento e

aprendizado rápido e fácil comparado a outras, a ideia principal de utilização desse programa para o projeto SiRoVIA, veio de Lawich, Benini e Guimarães (2019) onde foi pensado um projeto exclusivo utilizando esta plataforma de forma eficiente e prática.

Este trabalho tem como base projetos anteriores a ele, o estudo atual é o quinto na sequência SiRoVIA. O primeiro projeto intitulado, “Um Estudo Sobre Plataformas Gratuitas de CADs Orientado à Prototipagem Robótica em Impressora 3D”; trata-se do estudo que realizou um levantamento geral de todos softwares gratuitos, foi realizado uma comparação de desempenho e utilidade entre eles. Essa comparação se deu reproduzindo um conjunto de engrenagens destinado a um Veículo Aéreo Não Tripulado (VANT). (LOSS e BENINI, 2018).

O segundo projeto: “UM ESTUDO SOBRE PLATAFORMAS VIRTUAIS DE SIMULAÇÃO ROBÓTICA”; se resume a elaborar um estudo sobre os diversos ambientes virtuais de simulação orientado à robótica existentes no mercado (LAWICH, BENINI e GUIMARÃES, 2019) e (BENINI e GUIMARÃES, 2019).

O terceiro projeto: “Projeto de Enxame de Unidades Robóticas móveis virtuais”; tem objetivo de fazer um modelo 3D em CAD, para ser implementado em ambiente de simulação robótica, com finalidade de se obter um enxame de robôs em movimento para executar tarefas simples em um mesmo ambiente (PATERNOSTRO e BENINI 2019).

O quarto projeto chamado de: “Projeto de uma Unidade Robótica Móvel para a Indústria 4.0”: “Neste projeto é proposta a modelagem em 3D, na plataforma SketchUp, de uma URM com o máximo de detalhes, que consiga atuar no chão de fábrica, ou seja, que consiga detectar obstáculos, e que execute a sua tarefa”. (FERREIRA e BENINI, 2020, p.1).

Durante o processo de criação da URM foram utilizados alguns tutoriais, para fazer a importação do SketchUp para o CoppeliaSim, foi utilizado o documento de (FERREIRA e BENINI, 2020, p.1), e para a ativação dos motores da URM foi utilizado o tutorial do próprio site do CoppeliaSim (<https://www.coppeliarobotics.com/helpFiles/en/bubbleRobTutorial.htm>).

Com o avanço da tecnologia e a chegada da indústria 4.0 é necessária a implementação de robôs em um ambiente autônomo para realização de diversas

funções. A modelagem virtual e a simulação são de extrema importância para economia de gastos e tempo, e para sua excelente autonomia este projeto apresenta a lógica Fuzzy como principal técnica de programação.

Após revisar a URM virtual modelada no projeto feito por Ferreira e Benini (2020) foi necessário modelar uma nova URM, baseando-se em uma URM já modelada e alguns componentes no “3D Warehouse”. Foi confeccionado cada peça da URM utilizando o programa de modelagem "SketchUp Pro 2021"; deu-se início ao desenvolvimento do chassi no formato retangular que é o espaço onde se instala a bateria DC de 12V, que é ligado nos motores, o microcontrolador ESP-32 e os sensores infravermelho inferiores que detectam desníveis, os sensores de ultrassom dianteiros, traseiros e laterais que servem para detectar obstáculos no caminho, além dos parafusos para prender os equipamentos e da roda dianteira, na Tabela 1 é possível observar os componentes, a quantidade necessária e uma descrição dos mesmos para a criação do robô.

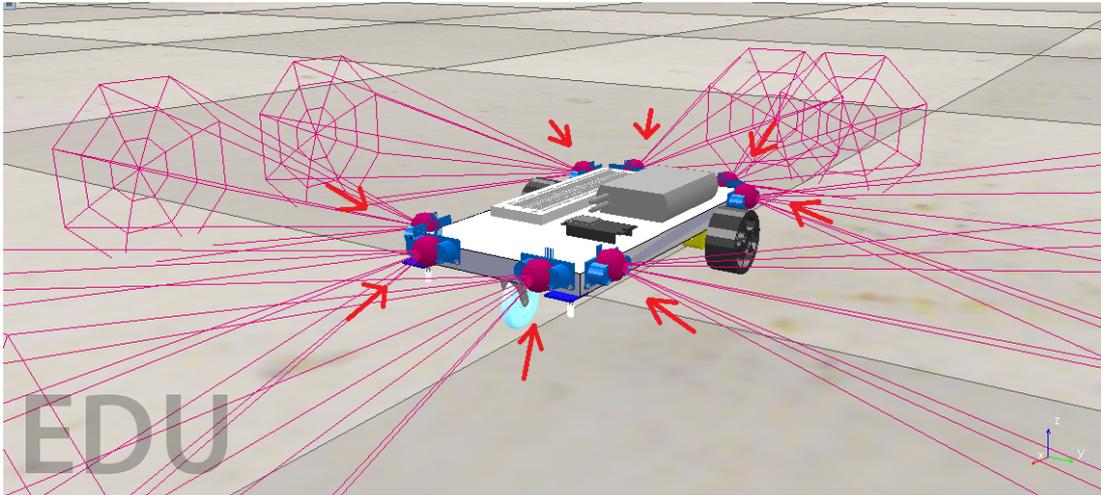
**Tabela 1 - Tabela de componentes da unidade robótica móvel e suas descrições.**

NOME	QUANTIDADE	DESCRIÇÃO
Motor DC	2	37 mm diâmetro, potência de saída: 6w pesa aprox. 30 g
Bateria 12v	1	Dimensões: 100mm x 60mm x 25mm, 190g, recarregável. pesa aprox. 45 g
Sensor de ultrassom	8	Tensão: dc 5v Distância de detecção: mínima de 2 cm e máxima de 450 cm pesa aprox. 10 g
Sensor de obstáculos infravermelho	4	Distância de detecção: 2-30cm, Ângulo de detecção:

		35º pesa aprox. 3g
Kit com regulador de tensão pro microcontrolador + protoboard + 65 jumpers	1	Dimensões: 16,5cm x 5,5cm pesa aprox. 50 g
Rodas da frente	2	Diâmetro: 68mm Largura: 26mm. Peso: 50g pesa aprox. 32 g
ESP32	1	Microcontrolador pesa aprox. 10 g
Controlador PWM	2	Dimensões: 3.4cm x 1.7cm x 1.2cm, Tensão de funcionamento: 5v-36v pesa aprox. 60 g
Encoder	2	Número de linhas: 100 linhas, Diâmetro externo: 22mm, Diâmetro do furo: 3.5mm pesa aprox. 10 g

Após a completa modelagem da URM, foi necessário utilizar todos os sensores modelados anteriormente no SketchUp e exportá-los no formato STL para o programa de simulação Coppelia, em seguida, houve a necessidade de posicionar cada sensor de forma que os mesmos fiquem alinhados para que ele consiga identificar objetos em todas as direções conforme ilustrado na Figura 1, indicados pelas setas vermelhas, depois configurá-los com distância e peso, e então fazer uma conexão entre eles e entre os motores da URM.

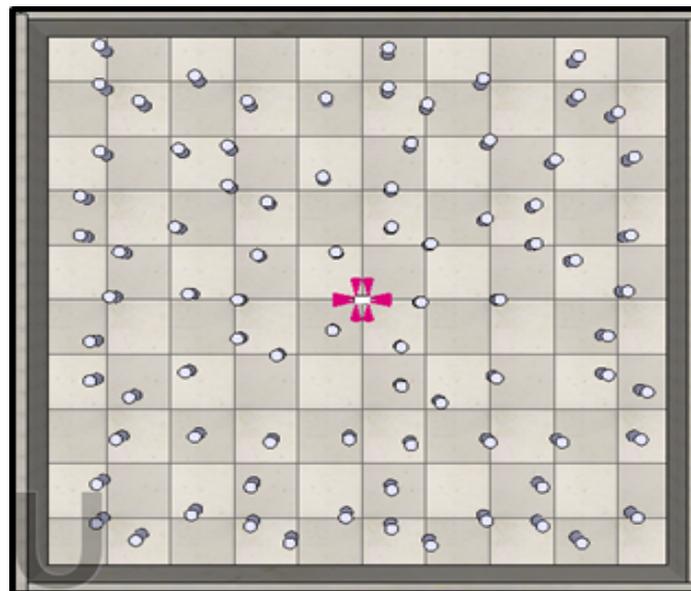
Figura 1 - Posicionamento dos sensores da URM.



Modelo da URM integrado no ambiente de simulação do CopeelliaSim

Com o desenvolvimento do ambiente virtual contemplando 5 categorias: ambiente amplo com bastante espaço, com passagem estreita, com obstáculos fixos, com obstáculos móveis, com obstáculos mistos (fixo e móvel). Passou a ser possível fazer o teste de peso e ação da URM.

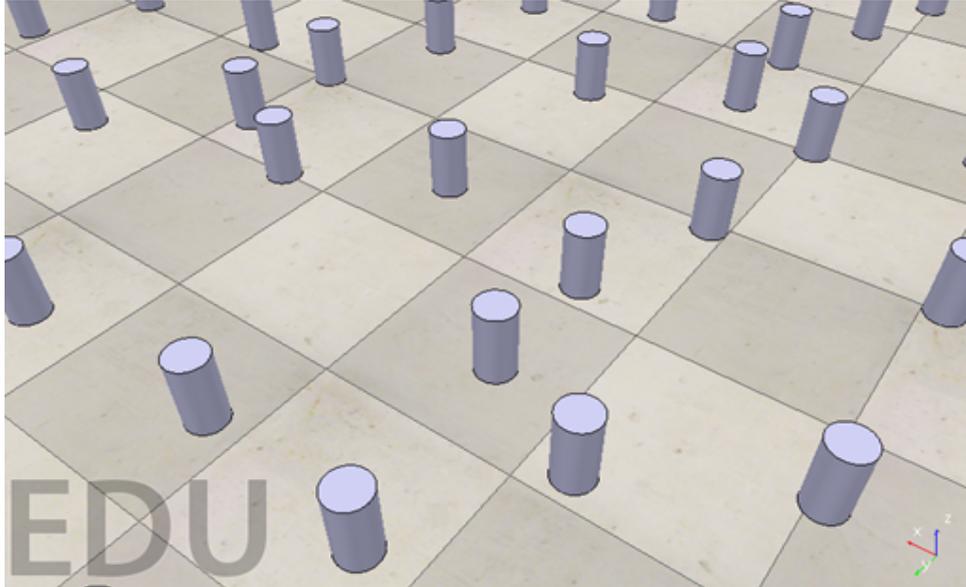
Figura 2 - Vista superior do ambiente virtual



Ambiente virtual com obstáculos fixos: Vista superior do ambiente amplo, criado para simulação no Coppeliasim, demonstrando a URM (com alcance do ultrassom nos quatro lados, em

vermelho) no centro e obstáculos fixos em seu entorno, utilizando esse ambiente para fazer os testes com sensores visando a integração dos mesmos com os motores.

Figura 3 - Vista diagonal do ambiente virtual com obstáculos fixos.

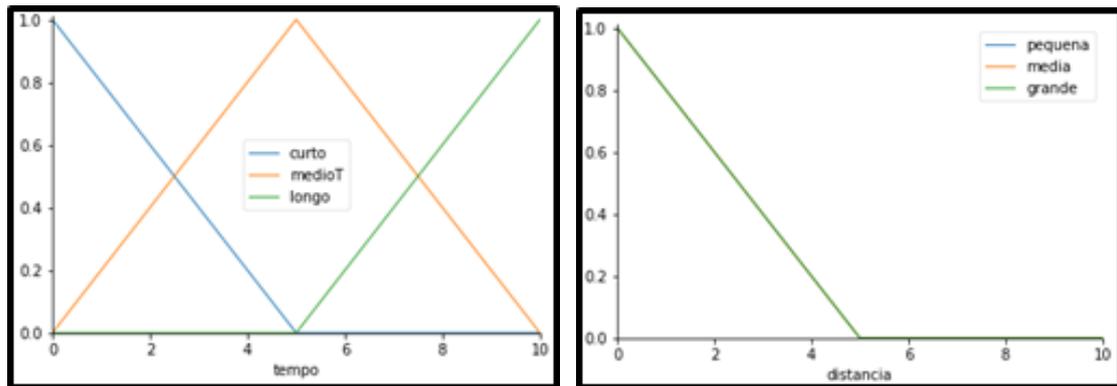


Ambiente de teste para URM implementado no ambiente virtual de simulação.

Foi então desenvolvido um programa, conforme ilustrado na Figura 4, utilizando a linguagem interna utilizada pelo CoppeliaSim chamada de “LUA” para configurar os sensores e os motores que permite que o robô ande para frente, em linha reta e quando o sensor encontrar um objeto a sua frente é necessário que a URM pare de se mover, então com umas modificações na programação base, o robô foi capaz de, além de parar, retroceder em alguma das direções para poder sair do caminho do objeto que estava a sua frente.



Figura 5 - Gráfico das funções de pertinências

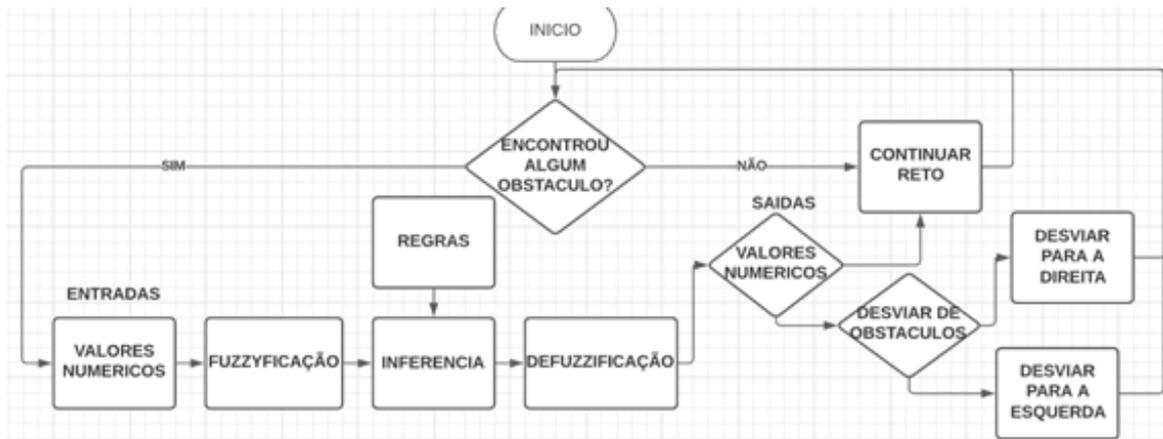


Funções de pertinência baseado nos sensores frontais da URM.

Os gráficos demonstram as funções de pertinência baseadas nos sensores frontais da URM o primeiro gráfico das funções de pertinência, baseado no tempo de resposta da URM gerado automaticamente utilizando regras de fuzzificação no Python e o segundo gráfico das funções de pertinência baseado na distância da URM em relação a algum obstáculo gerado automaticamente utilizando regras de fuzzificação no Python. Com os resultados obtidos das funções de pertinência foi desenvolvido a tabela verdade e o método de defuzzificação que controla os sensores e os motores da URM, foi utilizado o vídeo logica Fuzzy com Python (<https://www.youtube.com/watch?v=M5DrTSE87UQ>), para entender a funcionalidade dessa etapa, após feita no Python a programação de defuzzificação, foi necessário utilizar como base para a programação final implementado no CoppeliaSim.

Para se entender melhor o funcionamento da URM foi desenvolvido um fluxograma como ilustrado na Figura 6 para ser usado como base para a finalização do algoritmo implementado, que deu funcionalidade à URM através da implementação da lógica Fuzzy, visando a implementação da fuzzificação e defuzzificação que foram implementados no Python com função de gerar as escolhas de rotas de forma autônoma.

Figura 6 - Fluxograma da lógica Fuzzy

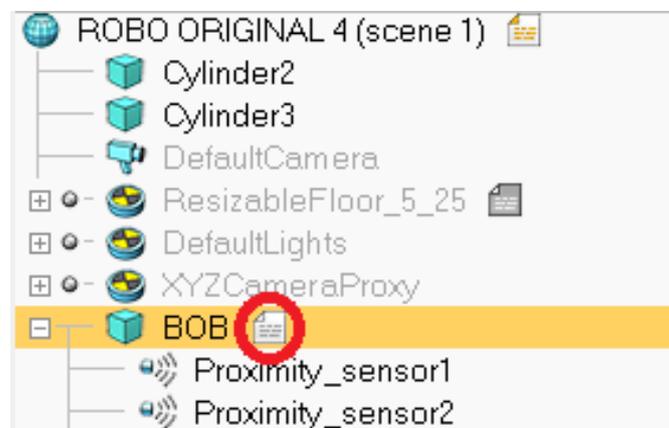


Fluxograma demonstrando a forma de funcionamento da Lógica de funcionamento da URM.

O algoritmo final em linguagem “LUA” foi desenvolvido na seção de *script* do CoppeliaSim. O código está disponível na Figura 8 e 9 e demonstra a programação utilizada para a autonomia da URM dentro do ambiente virtual.

A parte do algoritmo é o que comanda a URM por isso é muito importante a sua utilização. Para acessar a parte de *script* no CoppeliaSim, há a opção especificada na Figura 7, que é localizada ao lado do nome da URM que foi nomeada nesse caso como BOB.

Figura 7 - Ícone de configuração para acessar a página de *script*



Ícone para acesso da programação do CoppeliaSim

Figura 8 - Aplicação do *script* utilizado para autonomia da URM, primeira parte

```
Child script (BOB)
function sysCall_init()
    local robot=sim.getObjectHandle(sim.handle_self)
    local obstacles=sim.createCollection(0)
    sim.addItemToCollection(obstacles,sim.handle_all,-1,0)
    sim.addItemToCollection(obstacles,sim.handle_tree,robot,0)
    usensors={}
    for i=1,8,1 do
        usensors[i]=sim.getObjectHandle("Proximity_sensor"..i)
        sim.setObjectInt32Param(usensors[i],sim.proxintparam_entity_to_detect,obstacles)
    end

    motorLeft=sim.getObjectHandle("Motor_Esquerdo")
    motorRight=sim.getObjectHandle("Motor_Direito")
    noDetectionDist=0.5
    maxDetectionDist=0.2

    detect={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0}
    braitenbergL={-0.2,-0.4,-0.6,-0.8,-1,-1.2,-1.4,-1.6,-2, 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
    braitenbergR={-2,-1.6,-1.4,-1.2,-1,-0.8,-0.6,-0.4,-0.2, 0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0}
    v0=2
end
```

*Script*, primeira parte

Figura 9 - Aplicação do *script* utilizado para autonomia da URM, segunda parte

```
Child script (BOB)
function sysCall_cleanup()
end

function sysCall_actuation()
    for i=1,8,1 do
        res,dist=sim.readProximitySensor(usensors[i])
        if (res>0) and (dist<noDetectionDist) then
            if (dist<maxDetectionDist) then
                dist=maxDetectionDist
            end
            detect[i]=1-((dist-maxDetectionDist)/(noDetectionDist-maxDetectionDist))
        else
            detect[i]=0
        end
    end

    vLeft=v0
    vRight=v0

    for i=1,8,1 do
        vLeft=vLeft+braitenbergL[i]*detect[i]
        vRight=vRight+braitenbergR[i]*detect[i]
    end

    sim.setJointTargetVelocity(motorLeft,vLeft)
    sim.setJointTargetVelocity(motorRight,vRight)
end
```

*Script*, segunda parte

## REFERÊNCIAS

AGUADO, A. G; CANTANHEDE, M. A. **Lógica Fuzzy** in: Faculdade de Tecnologia – Universidade Estadual de Campinas (UNICAMP). Limeira, SP, 2010, p. 1-2

BENINI, F. A. V. Um sistema Fuzzy para desvio de obstáculos em ambiente virtual conhecido. In: 1º Congresso de Pós-Graduação do IFSP, 2016, Matão, SP, **Anais...** Matão: IFSP, 2016, p. 1-4.

GUIMARÃES, J. P. E.; BENINI, F. A. V. Um estudo sobre plataformas virtuais de simulação robótica. In: “V Semana Do Brincar: O Brincar Que Abraça a Diferença” e “II Encontro Dos Cursos Técnicos Em Lazer Dos Institutos Federais”, 2019, Rio Grande do Norte, **Anais...** Rio Grande do Norte: IFSP, 2019.

FERREIRA, M. C. E; BENINI, F. A. V. Projeto de uma unidade robótica móvel para a indústria 4.0. In: 11º Congresso de inovação, ciência e tecnologia IFSP (CONICT), 2020, Presidente Epitácio, SP, **Anais...** Presidente Epitácio: IFSP, 2020, p. 1-5.

LAWICH, G; BENINI, F. A. V; GUIMARÃES, J. P. E. Um estudo sobre plataformas virtuais de simulação robótica. In: 10º Congresso de Inovação, Ciência e Tecnologia do IFSP, 2019, Sorocaba, SP, **Anais...** Sorocaba: IFSP, 2019, p. 1-4.

LOSS, A. L. P; BENINI, F. A. V. Análise de Plataforma Gratuitas de Cads Orientadas à Prototipagem. In: 4º Workshop de inovação, 2018, São Carlos, SP, **Anais...** São Carlos: IFSP, 2018, p. 1-4

PATERNOSTRO, W. G. M; BENINI, F. A. V. **Projeto de exame de unidades robóticas móveis virtuais**. 2019. 7 f. Projeto de pesquisa, IFSP, São Carlos, 2019.

RIGNEL, D. G. S; CHENCI, G. P; LUCAS, C. A. Uma Introdução à lógica Fuzzy. In: **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, v. 1, p. 17-20, 2011. Disponível em: <http://periodicos.unifacfe.com.br/index.php/resiget/article/view/153>