



**ARTUR PASSOS DO AMARAL**

**O ALGORITMO FUZZY APLICADO À ROBÓTICA  
MÓVEL: OTIMIZAÇÃO E PROTOTIPAGEM EM CASAS DE  
VEGETAÇÃO**

**LAVRAS – MG  
2019**

**ARTUR PASSOS DO AMARAL**

**APLICAÇÃO DO ALGORITMO FUZZY NA ROBÓTICA MÓVEL:  
OTIMIZAÇÃO E PROTOTIPAGEM EM CASAS DE VEGETAÇÃO**

Monografia apresentada à Universidade Federal de Lavras como parte das exigências do Curso de Engenharia de Controle e Automação para obtenção do título de Bacharel.

Prof. Dr. Leonardo Silveira Paiva  
Orientador

**LAVRAS – MG**

**2019**

**ARTUR PASSOS DO AMARAL**

**APLICAÇÃO DO ALGORITMO FUZZY NA ROBÓTICA MÓVEL:  
OTIMIZAÇÃO E PROTOTIPAGEM EM CASAS DE VEGETAÇÃO  
FUZZY ALGORITHM APPLIED TO MOBILE ROBOTICS: OPTIMIZATION  
AND PROTOTYPING ON GREENHOUSES**

Monografia apresentada à Universidade Federal de Lavras como parte das exigências do Curso de Engenharia de Controle e Automação para obtenção do título de Bacharel.

APROVADA em 20 de novembro de 2019  
Dr. Leonardo Silveira Paiva UFLA  
Dr. Vinicius Miranda Pacheco UFLA  
Fernanda Costa e Silva UFLA

Prof. Dr. Leonardo Silveira Paiva  
Orientador

**LAVRAS – MG  
2019**

Dedico a conclusão deste trabalho à minha mãe, Solange, a quem eu agradeço por cada conquista e por lutar para que eu chegasse até aqui.

## **AGRADECIMENTOS**

Após inúmeros desafios chegou o momento de concluir mais essa etapa em minha formação profissional. A conclusão desse trabalho também significa o fechamento de um ciclo, que foi longo, com muitas noites sem dormir, mas com a sensação de dever cumprido.

Agradeço primeiramente à toda minha família que esteve ao meu lado desde o princípio dessa jornada. Agradeço a minha mãe, Solange, por nunca ter desistido de mim, ou de meus irmãos, e nos mostrar todos os dias que o nosso esforço é capaz de gerar transformação. Agradeço aos meus irmãos, Natália, Najara e Guilherme, que dividiram comigo uma vida. Sou grato também àquela que permitiu que eu seguisse meu sonho e me acolheu como a um filho. Muito obrigado por tudo, Vó Hercília! Junto a ela, não posso deixar de agradecer aos meus tios que também estiveram ao meu lado me apoiando todos os dias durante esses anos.

Aos meus amigos sou grato por me ajudarem a chegar até aqui, depois de tantos desafios na graduação. Agradeço ao Wesley Ribeiro, Fernando Borges, Barbara Andreatta, Andrey Marques, Eduardo Afonso, Bianca Aparecida e Vittoria Casati, com orgulho de citar a cada um deles. Sem eles esse momento não seria o mesmo, pois foi com eles com quem dividi meus sorrisos, minhas lágrimas, todos os momentos de descontração e com quem pude contar diante das infindáveis provas e dos trabalhos. Agradecimento especial à todos aqueles com quem tive a honra de dividir minha jornada no Movimento Empresa Júnior. Os quatro anos que estive na Robótica Jr. me ajudaram a me tornar uma pessoa melhor e mais consciente, além de me permitir conhecer pessoas incríveis e que fizeram história.

Ao Prof. Dr. Leonardo Paiva agradeço por me permitir aprofundar no estudo da Robótica e por toda a atenção e paciência dedicados ao ensino e ao projeto nos últimos anos.

Ao meu namorado, Matheus Mantovani, agradeço por estar comigo em meus momentos mais sinceros e por me acolher tão carinhosamente. Alguém em quem eu me inspiro cada vez mais e que me ajuda a enxergar a vida com bons olhos novamente.

A Deus agradeço por toda a minha existência.

**MUITO OBRIGADO!**

## EPÍGRAFE

*“Só se pode alcançar um grande êxito  
quando nos mantemos fiéis a nós mesmos.”*  
(Friedrich Nietzsche)

## RESUMO

Dentro das áreas de estudo da Robótica existe aquela que se trata por Robótica Móvel, cujo objetivo é o estudo e a otimização de percursos entre um ponto de partida e um ponto final no espaço, podendo ser aplicada em diversos ambientes, seja para substituir ou potencializar a ação humana. O presente trabalho utiliza o algoritmo *Fuzzy* para a geração *off-line* de trajetórias que demandem menor esforço para que o protótipo do robô se locomova dentro de uma casa de vegetação. Essa aplicação propõe a Robótica Móvel como uma substituição à atuação humana na manipulação de fatores nocivos à saúde ou em atividades que sejam de difícil execução para o ser humano, como a aplicação de agentes químicos para fertilização ou controle de pragas. A geração da trajetória *off-line* é feita a partir da representação gráfica da casa de vegetação, com pontos de partida e de chegada predefinidos, permitindo que o *Fuzzy* obtenha a menor distância entre esses pontos. Para que o algoritmo consiga gerar o grafo com o caminho a ser percorrido, inicialmente é feita a discretização do ambiente, a partir de sua representação, utilizando o software Matlab. A etapa subsequente consiste na construção da plataforma robótica, bem como o algoritmo que viabiliza sua locomoção, buscando reproduzir em escala real a trajetória com base no grafo gerado pelo algoritmo *Fuzzy*. Para a construção do robô móvel foi utilizado o microcontrolador Arduino associado a um conjunto de sensores e atuadores que possibilitam a interação entre o robô e o ambiente. Dessa maneira, o presente estudo busca avaliar a robustez do algoritmo proposto na geração da trajetória em aplicações reais, levando em consideração a complexidade do ambiente, apontando uma alternativa para as aplicações em casas de vegetação. Devido à crescente demanda por essas áreas de conhecimentos na sociedade, o presente projeto enxerga a necessidade de iniciativas no ambiente universitário que sejam voltadas à formação de profissionais aptos a aplicar tais tecnologias na resolução de problemas. Para destacar tal relevância, trabalho busca se apresentar como material de apoio no ensino de robótica, respeitando o perfil socioeconômico dos alunos de graduação, visando o ensino mais inclusivo de robótica e suas tecnologias. Alinhado a essa perspectiva educacional, um levantamento socioeconômico do corpo discente da Universidade Federal de Lavras foi feito e levado em consideração nas análises de custos para a realização de práticas didáticas em robótica.

**Palavras chave:** Robótica móvel. *Fuzzy*. Otimização de trajetória. Casas de vegetação.

## ABSTRACT

The study of robotics is divided into many fields of researches and one of them is that one known as mobile robotics, which aims the path optimization considering trajectories where the source and goal points in space are both previously known, showing itself as a new approach to replace or improve the humans efforts. This paper uses the *Fuzzy* algorithm as an *off-line* method to solve the path-planning problem on greenhouses pointing out low effort paths to a mobile robot prototype. This project aims to introduce the mobile robotics as a tool to reduce the impact of the manipulation of chemical products to human health and to replace humans on dangerous tasks. The path planning uses a discrete representation of the greenhouse giving inputs to the algorithm so it can find the best path using the software Matlab. The next stage of the project is consisted by the construction of the robot platform and development of its main code. The prototype is an association of many sensors and actuators responsible to allow the robot dynamics and its interaction to the real world. Thus, the purpose of this paper is to evaluate the *fuzzy* algorithm performance in path planning problem, leading to a reflection about the benefits of the mobile robotics applications in greenhouses. Due to the growing demand for these technologies in society, this paper shows how the academic environment needs to improve its efforts to provide the society with professionals who is capable to apply such technologies in problem solving. To highlight such relevance, this paper seeks to present itself as support material in robotics classes, respecting the socioeconomic profile of undergraduate students, aiming to make the robotics learning process more inclusive. According to this educational perspective, the project considered a socioeconomic survey of the student body of the University of Lavras to analyze the acquisition cost for implementation of didactic practices in robotics.

**Keywords:** Mobile robotics. *Fuzzy* algorithm. Path planning. Greenhouse.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Perfil dos ingressantes na UFLA sob o aspecto de renda .....	19
Figura 2 – Exemplos de aplicações da RM na sociedade.....	21
Figura 3 – Drone e robô autônomo no trato da lavoura .....	22
Figura 4 – Comparação entre o algoritmo A* original e a busca em vários ângulos.....	24
Figura 5 – Exemplo de caminho obtido (em vermelho) após a fase de questionamento face aos vértices e arestas (em azul) da fase de planejamento .....	25
Figura 6 – Exemplo do processo de captura do ambiente .....	26
Figura 7 – Trajetória traçada utilizando o algoritmo <i>fuzzy</i> .....	27
Figura 8 – Modelos de placa do microcontrolador Arduino. a) Arduino Uno Rev3; b) Arduino Mega 2560 Rev3. ....	29
Figura 9 – Exemplos de plataformas. a) Chassi 2WD com estrutura de acrílico e uma roda boba; b) Chassi 4WD com estrutura de acrílico com revestimento adesivo de proteção; c) Chassi 2WD com estrutura circular de acrílico com duas rodas bobas.....	31
Figura 10 – Sensor de distância ultrassônico HC-SR04.....	32
Figura 11 – Driver ponte-H dupla L298N .....	33
Figura 12 – Sensor de velocidade LM393 e disco de encoder. ....	35
Figura 13 – Fotografia do robô móvel.....	35
Figura 14 – Exemplos de estruturas para robôs móveis. a) Ambiente livre de obstáculos; b) Ambiente com obstáculos retangulares; c) Planta baixa de uma casa; d) Labirinto..	38
Figura 15 – Planta baixa da estufa.....	38
Figura 16 – Representações gráficas da estufa em bitmap .....	38
Figura 17 – Reprodução adaptada das instalações da estufa .....	39
Figura 18 – Representação em 3D do ambiente construído .....	39
Figura 19 – Discretização do ambiente em um bitmap .....	40
Figura 20 – Configuração inicial do algoritmo <i>fuzzy</i> .....	40
Figura 21 – Exemplo planejamento de caminho pelo algoritmo <i>fuzzy</i> .....	41
Figura 22 – Escala de cinza representada em 256 bits .....	42
Figura 23 – Resultado da adaptação das cores no algoritmo.....	42
Figura 24 – Conversão da imagem em bitmap em escala de cinza .....	43
Figura 25 – Mapa do ambiente em padrão textual representada no Excel .....	44
Figura 26 – Parametrização de índices e tratamento da rota em código .....	45

Figura 27 – Exemplo do mapa tratado em tempo de execução de código no Serial do Arduino.....	45
Figura 28 – Arquivo de texto com as informações do ambiente para interpretação do Arduino.....	46
Figura 29 – Inserção dos parâmetros do mapa no código do Arduino .....	46
Figura 30 – Exemplo da varredura para escolha do caminho.....	46
Figura 31 – Trecho do código responsável pela interpretação da varredura.....	47
Figura 32 – Trajetórias segundo o algoritmo <i>fuzzy</i> para diferentes ângulos máximos de curva. a) 15°; b)30°; c) 45°; d) 60° .....	49
Figura 33 – Trajetórias para diferentes ângulos máximos de curva. a) 15°, vel. máx. = 5; b) 15°, vel. máx. = 1; c) 45°, vel. máx. = 5; d) 45°, vel. máx. = 1; e) 60°, vel. máx. = 5; f) 60°, vel. máx. = 1 .....	50
Figura 34– Trajetórias segundo o algoritmo <i>fuzzy</i> para diferentes ângulos e velocidades. a) 15°, vel. máx. = 1; b) 15°, vel. máx. = 5; c) 45°, vel. máx. = 1; d) 45°, vel. máx. = 5.	52
Figura 35 – Trajetórias segundo o algoritmo <i>fuzzy</i> para diferentes ângulos em alta complexidade. a) 15°; b) 30°; c) 45°; d) 60°.....	53
Figura 36 – Etapas do planejamento do caminho. a) Configuração do algoritmo <i>fuzzy</i> ; b) Caminho descrito pelo algoritmo. ....	54
Figura 37 – Plataforma robótica e mapa construído.....	55
Figura 38 – Trajeto interpretado pelo código no Arduino.....	56

## LISTA DE TABELAS

Tabela 1: Requisitos elétricos de alimentação dos componentes .....	33
Tabela 2: Requisitos elétricos de alimentação dos componentes.....	36
Tabela 3: Resultados das simulações com o algoritmo <i>fuzzy</i> para diferentes parâmetros .....	48
Tabela 4: Resultados das simulações com o algoritmo <i>fuzzy</i> para diferentes ângulos e velocidades máximas.....	50
Tabela 5: Resultados das simulações com o algoritmo <i>fuzzy</i> frente a obstáculos de formatos diversos.....	51
Tabela 6: Resultados das simulações com o algoritmo <i>fuzzy</i> frente a obstáculos de formatos diversos.....	53
Tabela 7: Custo de aquisição dos equipamentos com base em 2017 e 2019 .....	57

## SUMÁRIO

1 INTRODUÇÃO.....	14
2 OBJETIVOS.....	17
2.1 Objetivo geral.....	17
2.2 Objetivos específicos.....	17
2.2.1 Ensino de robótica na universidade.....	17
2.2.2 Prototipagem e aplicação do robô móvel.....	18
3 REFERENCIAL TEÓRICO.....	18
3.1 Levantamento do perfil dos discentes.....	18
3.1.1 O espaço do discente no ensino de robótica.....	19
3.2 Aplicações da robótica móvel.....	20
3.2.1 Agricultura de precisão e seus benefícios.....	21
3.2.2 Aplicação da robótica móvel em casas de vegetação.....	22
3.3 Planejamento de caminho.....	23
3.3.1 Algoritmo A*.....	23
3.3.2 Algoritmo MRP.....	24
3.3.3 Algoritmo <i>fuzzy</i> .....	26
3.3.4 Escolha do algoritmo.....	28
4 MATERIAIS E MÉTODOS.....	28
4.1 Construção do protótipo.....	29
4.1.1 A escolha do microcontrolador.....	29
4.1.2 Estrutura do robô móvel.....	30
4.1.3 Sensoriamento de obstáculos.....	32
4.1.4 Controle de rotação e direção dos motores.....	32
4.1.5 Alimentação do protótipo.....	33
4.1.6 Sensores de velocidade.....	34
4.1.7 Características do protótipo.....	35
4.2 Descrição do problema e ambiente de atuação.....	36
4.3 Melhor caminho e descrição de rota.....	37
4.3.1 Mapeamento e discretização de ambientes.....	37
4.3.2 Algoritmo <i>fuzzy</i> e planejamento de rota.....	40
4.3.3 Tradução de rota e navegação.....	41
4.3.4 Configuração do robô móvel.....	44
5 RESULTADOS E DISCUSSÃO.....	47

5.1 Análise do algoritmo <i>fuzzy</i> em ambiente livre de obstáculos .....	48
5.2 Análise do algoritmo <i>fuzzy</i> frente a obstáculos paralelos .....	49
5.3 Análise do algoritmo <i>fuzzy</i> em uma planta baixa .....	51
5.4 Análise do algoritmo <i>fuzzy</i> em um labirinto .....	52
5.5 Aplicação do algoritmo <i>fuzzy</i> em ambiente estático.....	54
5.6 Análise de custos e o ensino de robótica .....	57
6 CONCLUSÃO.....	58
6.1 Perspectivas futuras .....	60
REFERÊNCIAS .....	62

## 1 INTRODUÇÃO

Assim como diversas áreas de estudo ligadas à tecnologia, a robótica também se desenvolveu nas últimas décadas buscando atender às demandas de consumo e produção da sociedade moderna. Grandes avanços tecnológicos foram alcançados na criação de sistemas robóticos mais robustos para a realização de tarefas específicas e que, em geral, demandam muita precisão de quem, ou o que, as executa. Por um lado, vê-se a disseminação do uso de robôs, principalmente, no setor industrial refletindo o sucesso de tais avanços. Entretanto, o aumento da demanda por esses sistemas inovadores trouxe também um aumento no valor agregado dos robôs, sejam eles fixos ou móveis. Frente a essa realidade, surge o desafio de trazer as práticas didáticas no ramo da robótica para o ambiente acadêmico, por se tratar de conhecimentos de grande importância na formação de profissionais de controle e automação.

A busca por formas alternativas e acessíveis de introduzir a robótica aos profissionais em formação no curso de Engenharia de Controle e Automação (ECA), foi o que motivou o desenvolvimento de uma atividade extracurricular voltada às práticas em robótica fixa e robótica móvel, possibilitando aos alunos aplicarem os conhecimentos adquiridos em sala de aula no desenvolvimento de projetos que possam ser utilizados na síntese do conteúdo abordado em diversas disciplinas do curso, por meio da robótica. Fomentado pela Universidade Federal de Lavras, o programa de Produção de Material Didático (PROMAD) para alunos de ECA permitiu que diversos alunos se familiarizassem com a robótica sob uma ótica educacional, durante sua vigência entre os anos de 2013 e 2019. Foi por meio do PROMAD que o presente projeto iniciou-se em 2017 e, durante um ano, esteve diretamente associado ao programa seguindo a vertente da robótica móvel.

A ciência da Robótica é um campo relativamente novo na tecnologia moderna e demanda conhecimentos em áreas como engenharias elétrica, mecânica, indústria e ciências da computação (SPONG et al., 2004). O progresso no campo da robótica tem possibilitado que os robôs superem a velocidade de operação dos seres humanos, além de evoluírem também em relação a outras habilidades (TANKE et al., 2012). Nesse contexto, surge a necessidade de classificar os robôs em dois conjuntos de acordo com a sua composição estrutural, sendo eles a robótica industrial e a robótica móvel (NIKU, 2013). Os robôs industriais são compostos, geralmente, de um braço articulado e um elemento final de atuação ligados a uma superfície fixa, já os robôs móveis são capazes de se

movimentar em seu ambiente de trabalho, ampliando sua zona de atuação e tornando sua aplicação muito mais diversificada em relação aos robôs industriais (PAIVA, 2017).

A robótica móvel (RM) consiste em dispositivos, dotados de certa autonomia, capazes de se locomover em um determinado ambiente (ALVES, 2015), considerando um ponto de partida e um ponto final do movimento. Entretanto, observada a aplicação desses robôs em ambientes reais, se faz necessária a especificação de diversos outros fatores que afetam o movimento, além das posições de partida e chegada. Uma forma de lidar com esse nível de detalhamento está na descrição do caminho por meio de pontos intermediários entre o início e o fim do movimento (CRAIG, 2005). O planejamento da trajetória desse movimento pode ser operado *off-line*, conhecendo previamente as informações sobre o ambiente e utilizando o conhecimento do robô sobre tais informações, ou *on-line*, fazendo-se necessário equipar o robô com sensores para monitorar seu movimento permitindo que o sistema de controle consiga reajustar sua trajetória em tempo real (GASARETTO et al., 2015).

A abordagem sobre o planejamento da trajetória pode ser por meio de métodos geométricos ou matemáticos do caminho, evitando colisões. De acordo com as configurações do robô, é necessário definir um espaço de configuração que representa as possíveis configurações do sistema. Feita a definição desse espaço, é possível utilizar diferentes estratégias para se realizar o planejamento da trajetória, como por exemplo o *roadmap*, que identifica caminhos dentro do espaço livre separando-o em nós e as ligações entre eles; campo potencial, que gera uma função matemática no espaço observando a presença de obstáculos; e também a lógica *fuzzy*, adotada nesse trabalho, que considera um sistema de inferências *fuzzy* para construir uma regra base que visa evitar obstáculos e a busca pelo objetivo (LEÇA, 2015; BOUMEHRAZ e BENMAHAMMED, 2014).

Os robôs móveis já vêm sendo utilizados em diversas áreas em nosso cotidiano como na indústria, assegurando o fluxo correto de materiais entre máquinas e postos de trabalho (JUNIOR, 2018). Outra área em que tem crescido a necessidade do aumento de eficiência é o setor agrícola, devido ao aumento demográfico das últimas décadas que implica no aumento na demanda por alimentos (HACKENHAAR et al., 2014). Cerca de 42% da população mundial opta pela agricultura como ocupação principal, fato que gerou interesse em investigar se a introdução de pequenas máquinas autônomas na agricultura seria mais eficiente do que o uso de tratores e da força humana (CHALWA et al., 2014), além de reduzir os riscos da exposição humana às condições adversas de trabalho. Nesse

setor, a inserção de novas tecnologias tem possibilitado a manipulação de fertilizantes minerais e orgânicos, além do melhoramento químico, a fim de obter o máximo rendimento na produção (NGUYEN et al., 2017).

O presente trabalho propôs-se a aplicar e avaliar o uso do algoritmo *fuzzy* para controlar o movimento de robôs móveis em ambientes estáticos. A lógica *fuzzy* foi introduzida por Zadeh na década de 60 para fornecer ferramentas matemáticas para o tratamento de informações de natureza imprecisa e dotada de subjetividade (LEITE, 2013). Sistemas de controle *fuzzy* são baseados em um conjunto de regras ou no conhecimento humano, descrevendo uma relação de reações condicionais, do tipo IF-THEN (ADURKAR; TANKAR, 2017), com base nas funções de pertinência de um dado ou informação a um determinado grupo. Sabendo o grau de pertinência de uma informação quanto aos grupos de informações disponíveis, os algoritmos baseados em lógica *fuzzy* são capazes de gerar saídas, como por exemplo decisões de rota, de forma simples e, em geral, intuitiva ao entendimento humano. Para demonstrar que o algoritmo *fuzzy* é capaz de atender, satisfatoriamente, à demanda de eficiência do setor agrícola, objetivou-se à construção de uma solução voltada a uma casa de vegetação visando demonstrar a precisão do robô na locomoção e posicionamento para a correta manipulação de fertilizantes e agentes químicos, propondo uma alternativa à exposição humana aos riscos gerados por estes à saúde e apontando uma nova abordagem para o aumento da eficiência na produção agrícola.

Além das motivações mais tecnicistas, o presente trabalho também se objetiva a realizar um levantamento sobre o papel do aluno diante do ensino de robótica e como seu contexto pode interferir em sua formação profissional e na síntese dos temas e tecnologias propostos nesse projeto, cuja origem está ligada ao PROMAD, Programa de Produção de Material Didático para alunos de Engenharia de Controle e Automação com foco no ensino de robótica. Tal levantamento se propõe a munir professores e alunos com informações que viabilizem o desenvolvimento e manutenção do caráter inclusivo da educação pública no Brasil. Para tanto, antes do planejamento e execução do projeto proposto, foi necessário compreender o ambiente no qual os discentes estão inseridos, bem como seu local enquanto indivíduos e agentes no ambiente universitário.

## **2 OBJETIVOS**

### **2.1 Objetivo geral**

O presente trabalho teve como objetivo o planejamento da trajetória do protótipo de um robô móvel, aplicado a uma casa de vegetação, buscando um caminho com menor custo para o robô, evitando colisões, utilizando o algoritmo *fuzzy* na geração do caminho. Uma vez validado o potencial de aplicação do protótipo e seus benefícios, este trabalho objetiva afirmar-se perante o ensino de robótica como um projeto adequável às práticas didáticas de tal disciplina robótica, realizando levantamentos sobre o perfil do corpo discente e como este perfil pode apresentar-se como um fator limitante à formação profissionais aptos a solucionar problemas em nossa sociedade por meio das tecnologias abordadas nesse projeto.

### **2.2 Objetivos específicos**

Para atingir os resultados desejados, o objetivo geral foi subdividido em dois grupos de objetivos específicos que contemplam o desenvolvimento deste projeto de acordo com a construção do robô móvel, sua aplicação e a importância de tornar as práticas em robótica mais inclusivas para a formação do engenheiro de controle e automação. Tais objetivos seguem listados abaixo:

#### **2.2.1 Ensino de robótica na universidade**

- a) Realizar o levantamento do perfil socioeconômico dos alunos da Universidade Federal de Lavras;
- b) Planejar a estrutura do protótipo, buscando reduzir os custos de aquisição para torna-lo mais acessível aos alunos;
- c) Realizar uma análise informativa sobre o contexto socioeconômico dos alunos de baixa renda e seu impacto sobre o ensino de robótica, diante da proposta de ensino inclusivo da universidade pública no Brasil.

### 2.2.2 Prototipagem e aplicação do robô móvel

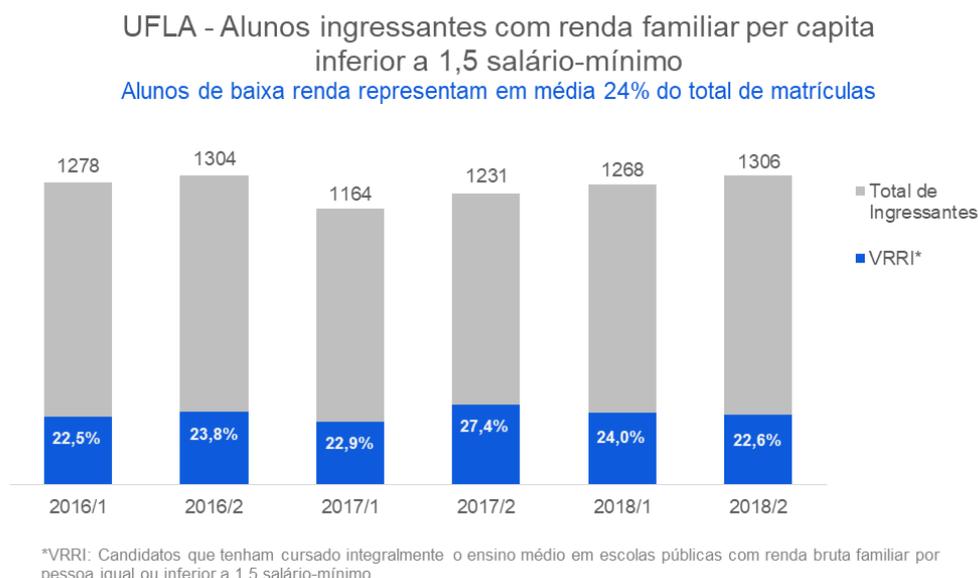
- a) Dimensionar a casa de vegetação e mapear os obstáculos e limites do ambiente;
- b) Gerar uma representação gráfica do ambiente, fiel à etapa de dimensionamento, para realizar sua discretização;
- c) Aplicar o algoritmo *fuzzy*, utilizando o software Matlab, para obter o grafo indicando o melhor caminho para o robô móvel se locomover pela casa de vegetação;
- d) Construir o protótipo do robô móvel, instrumentado com os sensores e atuadores necessários para o reconhecimento do ambiente durante o movimento;
- e) Programar o microcontrolador Arduino para responder ao sensoriamento do robô e percorrer o caminho planejado, considerando os parâmetros definidos pelo grafo da trajetória.

## 3 REFERENCIAL TEÓRICO

### 3.1 Levantamento do perfil dos discentes

Por ter um papel social de inclusão tão importante, a universidade pública no Brasil possui um quadro diverso e plural em seu corpo discente. Sob esse aspecto, em um cenário ideal toda iniciativa didática pensada para o ambiente de sala de aula deve considerar que tal prática precisa ser inclusiva o suficiente para que todos os alunos sejam alcançados em uma formação profissional justa. Segundo dados do plano de desenvolvimento institucional 2016 - 2020 da UFLA, em média 24% dos ingressantes na universidade entre 2016 e 2018 são alunos de baixa renda, cujo rendimento familiar mensal per-capita é igual ou inferior a 1,5 salário-mínimo (FIGURA 1).

Figura 1 – Perfil dos ingressantes na UFLA sob o aspecto de renda



Fonte: Do autor (2019).

Assumindo que o indicador supracitado também representa o perfil de ingressantes no curso de ECA, de acordo com as regras de seleção e distribuição de vagas do Sistema de Seleção Unificada (SISU), tal perfil acadêmico se tornou uma das premissas para o planejamento do projeto.

### 3.1.1 O espaço do discente no ensino de robótica

Ao analisar o corpo discente sob tal ótica, para conseguir contemplar o máximo de alunos nas práticas didáticas, um dos fatores mais relevantes, principalmente para os alunos de baixa renda, é o valor de aquisição dos equipamentos necessários para o desenvolvimento do protótipo e a sua relação com a metodologia de ensino adotada na disciplina de robótica.

Com o perfil dos estudantes já compreendido, outro fator importante é considerar a disciplina de robótica na UFLA e sua ementa. Durante a sua passagem pela disciplina, os alunos absorvem conceitos por meio de aulas teóricas, com o objetivo de introduzir os conteúdos aos alunos; e de aulas práticas, visando sintetizar tais conteúdos. Como parte da avaliação, durante o semestre é proposto aos alunos a realização de um trabalho prático, desenvolvido em grupos, voltado à prototipagem de robôs fixos e/ou móveis.

### 3.2 Aplicações da robótica móvel

Define-se como um robô móvel um mecanismo composto por uma arquitetura de hardware e software, objetivando locomover-se em um ambiente para a execução de uma tarefa final (VIEIRA, 2005). Para tanto, é necessário que o controle utilizado nesses robôs seja robusto o suficiente para considerar distúrbios externos encontrados em ambientes reais, como desníveis, deslizamento de rodas, colisão com obstáculos e incertezas paramétricas (VENTURA, 2011). Esse controle de navegação pode ser realizado de diversas maneiras, como robôs seguidores de linha, robôs exploradores, robôs controlados via *bluetooth* ou *wi-fi* (CARDOSO, 2016). Entretanto, no presente estudo o robô será controlado por meio de um microcontrolador embarcado ao protótipo e associado a um conjunto de sensores e atuadores, responsável de executar uma trajetória predeterminada pelo algoritmo *fuzzy*.

Além de tais fatores, existem outras características que podem ser limitantes em aplicações com robôs móveis, como sua fonte de alimentação, em sua maioria consistente de uma bateria, como é o caso do presente estudo. É necessário que tal fonte de alimentação seja capaz de atender à demanda energética do robô para que a navegação seja executada corretamente. Tal demanda leva em consideração o tipo aplicação específica do robô e a composição de sua arquitetura.

Considerando o cenário descrito anteriormente, existem diversas aplicações da RM na sociedade moderna, demonstrando o potencial do futuro dessa área. Tais aplicações estão presentes em ambientes domésticos, em aspiradores de pó e cortadores de grama robóticos; industriais, no transporte automatizado e em veículos de carga autônomos; segurança e defesa civil e militar, no controle e patrulhamento de ambientes e no resgate e exploração em ambientes hostis (WOLF et al., 2009). Na Figura 2 são mostrados exemplos dessas aplicações: em a) é mostrado um robô aspirador de pó, utilizado em ambiente doméstico; já em b) é mostrado o veículo de carga autônomo da Volvo, visando substituir caminhões com piloto manual no transporte de cargas; e em c) é mostrado o robô antibomba adotado pela Polícia Federal do Brasil para a segurança da Copa do Mundo de Futebol em 2014.

Figura 2 – Exemplos de aplicações da RM na sociedade



Fonte: a) O Globo (2017); b) Estadão (2019); c) ConJur (2010).

Além dessas aplicações, outro setor que têm demandado a inserção de novas tecnologias, como a RM, é a agricultura de precisão, também conhecida por manejo sítio-específico. Estima-se que a população mundial deverá dobrar nas próximas três décadas, fazendo-se necessário o emprego de novas tecnologias para aumentar a produção de alimentos utilizando de forma eficiente os recursos naturais e insumos agrícolas, gerando menor impacto ambiental (ARTUZO et al., 2014).

### 3.2.1 Agricultura de precisão e seus benefícios

O emprego de tecnologias na agricultura pode ser interpretado como a alternativa para que os agricultores atinjam seus objetivos de produção. Tais objetivos incluem a gestão de risco, a qualidade de vida e a rentabilidade, permitindo que eles se mantenham nos negócios (SWINTON et al., 2001). Assim, segundo Pierce e Nowak (1999 citado por MIRANDA; VERÍSSIMO; CEOLIN, 2017, p. 129), “a agricultura de precisão (AP) pode ser definida como o processo pelo qual garante-se a uniformização da eficiência em cada metro quadrado da área de manejo otimizando a produção e diminuindo a variabilidade das áreas da lavoura”. Esse processo depende do gerenciamento de informações da lavoura e só é possível pela integração de novas tecnologias, como o uso de sensores para identificar necessidades do gado ou de culturas (FARM MANAGEMENT, 2018).

Para demonstrar o cenário de adoção de novas tecnologias no setor agrícola, utilizou-se o estudo de Whipker e Akridge (2009), que aponta que naquele ano 85% dos agricultores nos EUA utilizaram alguma técnica de agricultura de precisão. Destes, 52,5% utilizaram tecnologias de navegação por GPS com direção ou controle automáticos, sendo esta a terceira tecnologia mais utilizada. Dentro dessa tecnologia pode-se listar, por exemplo, o uso de tratores completamente autônomos, em que não há cabine de piloto.

Outro exemplo é o uso de *drones* na pulverização de pontos específicos de uma lavoura (FIGURA 3), economizando em defensivos e mão de obra (EXAME, 2018).

Figura 3 – *Drone* e robô autônomo no trato da lavoura



Fonte: *Farm Management* (2018).

Os benefícios da inserção dessas tecnologia no campo são apontados também no Brasil, como mostra o estudo realizado por Filho e Cunha (2015) na região sudoeste do estado de Goiás, onde 68,18% dos agricultores utilizavam piloto automático em suas lavouras. Já no estudo realizado por Artuzo e Jandrey (2014), os autores analisaram e compararam áreas de agricultura convencional (AC) e agricultura de precisão (AP) no estado do RS, chegando à constatação de que o uso de fertilizantes é, em média, 33,1% menor nas áreas de AP em lavouras de soja. Além disso, foi observado um aumento médio de 10,39% na produtividade das lavouras com o uso da AP, o que justifica as motivações do presente estudo e o esforço de pesquisadores no desenvolvimento de novas tecnologias para o campo.

### 3.2.2 Aplicação da robótica móvel em casas de vegetação

O uso de robôs como veículos agrícolas autônomos (também conhecidos como *agribots* ou *agrobots*) é uma valiosa ferramenta para a AP, agregando o uso de teorias em controle robótico já fundamentadas e consolidadas para aplicação em outras áreas (TANGERINO et al., 2011). Alinhando esse conceito com a tendência de ampliação da agricultura de precisão, descrita na seção 3.1.1, o seccionamento de lavouras e estufas permite introduzir os *agrobots* no campo para atuação em regiões específicas do cultivo. Tal seccionamento cria um ambiente em que um conjunto de *agrobots* atuam em conjunto em toda a lavoura, gerando um ambiente mais complexo e dinâmico em que tais robôs trocam informações. O objetivo deste trabalho, entretanto, é demonstrar como um robô

móvel pode ser aplicado à agricultura, sem considerar um trabalho em conjunto com outros robôs, considerando uma seção do ambiente real, nesse caso uma estufa.

Estufas agrícolas são construções que oferecem um ambiente com condições apropriadas para um determinado cultivo, como hortaliças e certas frutas (ALVES, 2015). A atmosfera criada dentro das estufas possibilita o crescimento das plantas mas também causam o surgimento de pestes e outros organismos indesejados, fazendo necessário o uso de pesticidas e outros produtos químicos que devem ser administrados diretamente sobre a planta (ACACCIA et al., 2003). Nessa linha de atuação, estudos como o de Buemi et al. (1994) já propunham uma estrutura robótica composta de um robô móvel, um braço robótico com seis graus de liberdade e uma garra, além de um conjunto de câmeras, servindo de interface para que o ser humano fosse capaz de examinar a planta e atuar sobre ela conforme a necessidade.

### **3.3 Planejamento de caminho**

Para avaliar o desempenho de um robô móvel durante seu percurso, alguns aspectos devem ser considerados, como o tempo de percurso, distância total e também as características físicas do ambiente em que o robô irá se locomover. Por essa razão uma etapa muito importante para avaliar esses critérios é o planejamento do caminho. Na literatura existem diversos algoritmos que auxiliam os pesquisadores a determinar qual é o melhor caminho para o trajeto, como os algoritmos A\*, MRP, algoritmos genéticos e também o *fuzzy*. Cada algoritmo é constituído de um princípio de funcionamento, tornando-o melhor para aplicações distintas.

#### **3.3.1 Algoritmo A\***

O algoritmo A\*, ou A estrela é um dos mais utilizados na resolução do problema de planejamento de caminhos para aplicações em robótica móvel. O algoritmo usa uma combinação de buscas heurísticas e buscas baseadas no menor caminho entre dois pontos (DUCHOŇ et al., 2014). Essa busca é baseada no cálculo do calor da função heurística em cada nó, ou fração, do ambiente de trabalho, além de realizar a checagem de vários nós adjacentes para decidir qual é o melhor caminho buscando zerar a probabilidade de colisão (GURUJI; AGARWAL; PARSEDIYA, 2016). O cálculo, mostrado na equação

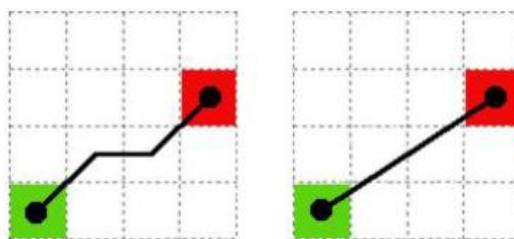
3.1, leva em consideração a distância heurística do objetivo do movimento,  $h(v)$ , e o tamanho do percurso entre o início do movimento e o seu ponto final,  $g(v)$ .

$$f(v) = h(v) + g(v) \quad (3.1)$$

A cada uma das células adjacentes à célula analisada é associado um valor da função heurística. Assim, a célula com o menor valor é escolhida como a próxima da sequência de movimento (DUCHOŇ et al., 2014). Para que o algoritmo consiga realizar esse cálculo, é necessário, a princípio, realizar a modelagem do ambiente de trabalho a fim de discretizá-lo. Tal discretização é caracterizada como um gráfico composto por vértices e arestas, em que cada vértice é descrito como um pixel do mapa de resolução reduzida enquanto as arestas caracterizam as conexões desses vértices, desde que a conexão seja livre de colisão (PAIVA, 2017).

Existem variações do algoritmo A\* que visam ampliar a capacidade do algoritmo original de gerar o melhor caminho. O algoritmo A\* básico se baseia nas células mais próximas às quais a subsequência do caminho é possível, o que restringe a análise do caminho e gera uma trajetória em ziguezague. Algumas adaptações no algoritmo permitem que ele consiga considerar também células mais distantes (FIGURA 4), introduzindo a busca em vários ângulos, tornando a busca muito mais semelhante ao comportamento humano (DUCHOŇ et al., 2014).

Figura 4 – Comparação entre o algoritmo A\* original e a busca em vários ângulos



Fonte: Duchoň (2014).

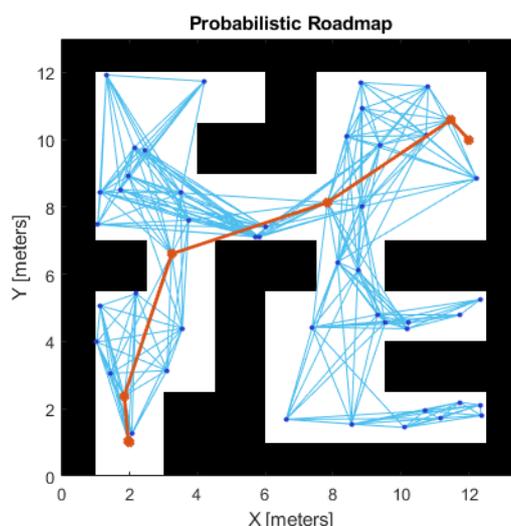
### 3.3.2 Algoritmo MRP

Outro algoritmo muito utilizado no planejamento de caminhos é o mapa de rotas probabilístico, também conhecido como MRP. Inicialmente o algoritmo foi proposto para o planejamento de trajetórias em ambientes estáticos. Entretanto, existem diversos

algoritmos baseados no MRP que têm sido empregados também em ambientes dinâmicos (KAVRAKI et al., 1996).

O MRP pode ser descrito como um grafo  $R = (N,E)$ , em que  $N$  representa os nós do grafo e  $E$  as suas bordas. A conexão é calculada por um planejador local que busca ligar dois nós vizinhos por meio de uma semirreta discretizada (ADÔRNO; BORGES, 2006). Nessa etapa, quanto maior o número de vértices, maiores serão as possibilidades de conexão pelo algoritmo, que ao conectar um par de vértices adiciona a reta que os liga como uma aresta. Por se tratar de um método probabilístico, as rotas são definidas de forma incremental. Ao identificar áreas difíceis no espaço livre de forma heurística, o algoritmo busca aumentar a conectividade nesses pontos do mapa gerando configurações adicionais do robô (PAIVA, 2017). Essa fase é chamada de fase de aprendizado, ou planejamento, que gera um mapa de rotas utilizado na fase seguinte do algoritmo, chamada de fase de questionamento. Na fase de questionamento o planejador local acessa o grafo gerado na etapa de aprendizado e busca um caminho que faça a ligação entre o ponto de partida e o de chegada da trajetória, caracterizando uma busca no grafo. A adoção dessa etapa é justificada pela estratégia de alocar maior esforço computacional na etapa de aprendizado, tornando a definição do caminho, pela fase de questionamento, muito mais rápida em termos de execução (JÚNIOR, 2018). A Figura 5 mostra o funcionamento do algoritmo na escolha do melhor caminho.

Figura 5 – Exemplo de caminho obtido (em vermelho) após a fase de questionamento face aos vértices e arestas (em azul) da fase de planejamento



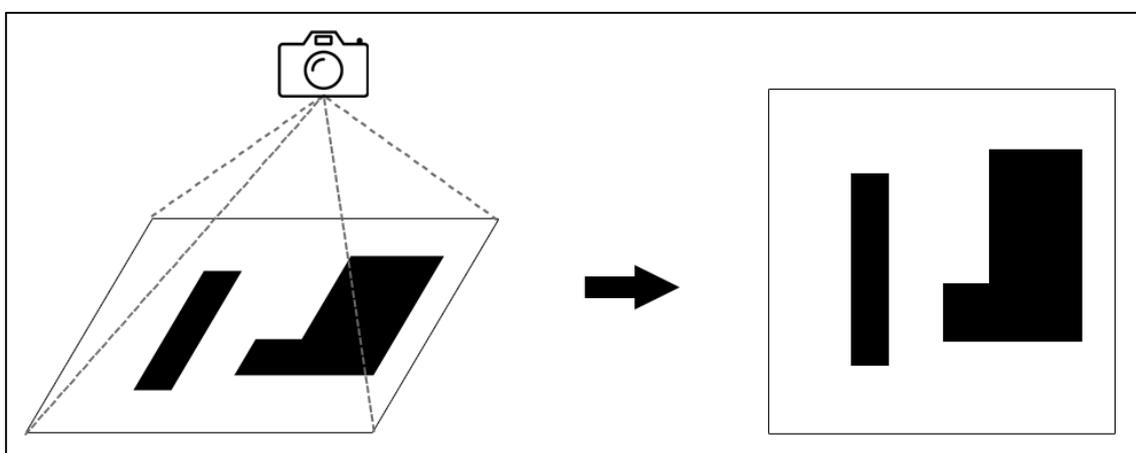
Fonte: <https://www.mathworks.com/help/robotics/examples/path-planning-in-environments-of-difference-complexity.html>.

### 3.3.3 Algoritmo *fuzzy*

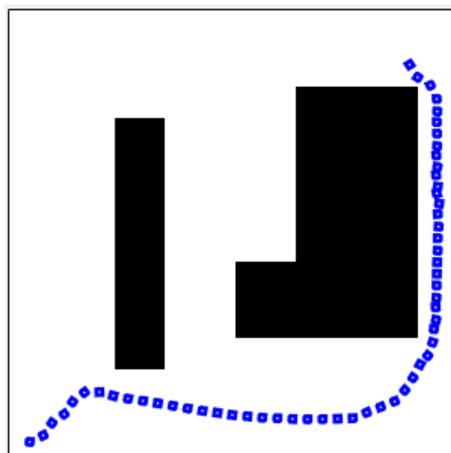
A lógica *fuzzy* pode ser descrita como uma mímica do comportamento humano, permitindo que o sistema consiga trabalhar com incertezas (PANDEY; PARHI; 2017). Sistemas que utilizam controladores *fuzzy*, são baseados em um conjunto de regras composto de uma coleção de ações e reações (se evento 1, então reação 1; se evento 2, então reação 2) baseadas no conhecimento humano sobre o fenômeno que se deseja controlar. Entre as inúmeras possibilidades de aplicação desse algoritmo está a navegação baseada em lógica *fuzzy*. Nesse caso, o planejamento da trajetória é reativo e consiste em um conjunto de regras *fuzzy* capaz de considerar a posição imediata do veículo e as distâncias dos obstáculos do ambiente para determinar o próximo movimento a ser realizado (KALA, 2014).

Para a aplicação do algoritmo *fuzzy* ao planejamento de trajetória, assume-se um ponto de observação suspenso sobre o ambiente e capaz de enxergar a posição do robô e de todos os obstáculos presentes. Tal ponto de observação pode ser interpretado como uma câmera calibrada para capturar o mapa do robô (FIGURA 6). Observando o ambiente, essa câmera também é capaz de observar o ponto de partida e os movimentos do robô. Para calcular a trajetória, seis informações de entrada são necessárias: distância do obstáculo à frente, distância do obstáculo à frente na diagonal esquerda, distância do obstáculo à frente na diagonal direita, ângulo entre a direção atual do robô e a direção do objetivo, distância do robô para o objetivo e o sentido de curva preferido (sentido horário ou anti-horário) (KALA, 2014).

Figura 6 – Exemplo do processo de captura do ambiente



Fonte: Do autor (2019).

Figura 7 – Trajetória traçada utilizando o algoritmo *fuzzy*

Fonte: Do autor (2019).

A Figura 7 mostra o caminho encontrado pelo algoritmo sobre o mesmo ambiente descrito na Figura 3. A tomada de decisão sobre os movimentos do robô é feita de acordo com a percepção da distância dos obstáculos, e produz uma saída que pode ser a direção da curva a ser feita ou a velocidade angular das rodas. O algoritmo *fuzzy* possui três possibilidades de decisões, com base em seu conjunto de regras *fuzzy*, para descrever o caminho:

- a) Obstáculo à frente muito distante: Curva em direção ao objetivo;
- b) Obstáculo à frente está próximo e um novo obstáculo é encontrado: Curva para o lado em que está o objetivo;
- c) Obstáculo à frente está próximo e o mesmo obstáculo encontrado no passo anterior é encontrado: Repetir o mesmo movimento anterior;

Devido à simplicidade desse princípio de funcionamento e sua capacidade de executar inúmeras tarefas sem demandar grande esforço computacional, o uso do controlador *fuzzy* tem se difundido entre pesquisadores na resolução de problemas (ANDURKAR et al., 2017). Assim, alinhado com sua proposta inicial, o presente projeto vislumbra fazer das práticas em robótica uma ponte de contato entre alunos e tecnologias como o algoritmo *fuzzy* na resolução de problemas de engenharia.

### 3.3.4 Escolha do algoritmo

Além dos algoritmos descritos anteriormente, existem outros métodos utilizados no planejamento de caminho, sejam eles adaptações ou algoritmos distintos. Estudos como o de Lamini et al. (2018), por exemplo, destaca o uso de algoritmos genéticos (AG) para a resolução de problemas de planejamento de rotas, trazendo a comparação entre três abordagens distintas com AG em relação aos seus números de iterações e curvas no melhor caminho encontrado. Já segundo Paiva (2017), ao comparar seis diferentes algoritmos de planejamento de caminho, incluindo MRP e A\*, já mencionados anteriormente, viu-se que o esforço cinemático do robô para a realização das rotas foi menor por meio do algoritmo MRP em comparação com A\* e os demais algoritmos, apesar do custo de processamento computacional elevado desse método.

Entendendo que o presente projeto visa realizar o planejamento da trajetória off-line, o tempo de processamento do caminho proposto se torna um parâmetro secundário na análise de desempenho. Entretanto, o projeto buscava deixar portas abertas para que o protótipo seja melhorado, permitindo que o processamento passe a ser realizado online. Por essa razão, a escolha do algoritmo *fuzzy* se baseou no alto tempo de processamento de alguns dos algoritmos presentes na literatura. Com essa escolha, o presente trabalho traz, fora do foco do projeto, informações sobre os tempos de processamento do algoritmo *fuzzy*, provendo informações de fácil acesso para uma comparação com os demais algoritmos em aplicações futuras pelos alunos do curso de ECA na Universidade Federal de Lavras.

## 4 MATERIAIS E MÉTODOS

Entendido o contexto do ensino de robótica na universidade, antes de partir para a etapa de desenvolvimento do robô e o planejamento da trajetória foi necessário realizar um levantamento dos custos de aquisição de kits didáticos básicos para robótica móvel encontrados no mercado e, entendendo os objetivos práticos do projeto e da disciplina, buscar alternativas para baratear o projeto, viabilizando a aquisição dos equipamentos pelos grupos de alunos. Nessa etapa, os seguintes levantamentos foram realizados:

- a) Levantamento dos equipamentos necessários para a construção do protótipo;

- b) Comparação dos custos de aquisição e compra dos equipamentos necessários.

#### 4.1 Construção do protótipo

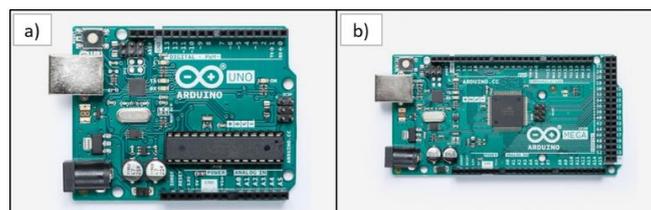
A etapa de levantamento de equipamentos e pesquisa de mercado, levando em consideração o escopo do projeto, teve como resultado o mapa do material necessário para a etapa de prototipagem. Durante a pesquisa, foi necessário compreender qual seria o conjunto de sensores e atuadores que atenderia ao propósito de permitir que o robô cumprisse sua tarefa de locomoção, de acordo com o caminho definido pelo algoritmo *fuzzy*. É importante destacar que todos os equipamentos foram escolhidos levando em consideração não apenas sua capacidade de integrar-se ao projeto do protótipo, mas também o quão acessíveis e disponíveis estes estariam para o perfil de alunos já traçado.

##### 4.1.1 A escolha do microcontrolador

Responsável pela execução do código que rege o movimento do protótipo, o microcontrolador é o componente mais importante no momento da escolha dos equipamentos do projeto. Suas características físicas, computacionais e eletrônicas podem se apresentar como potencialidades ou fatores limitantes do projeto a depender da finalidade.

A opção pelo microcontrolador Arduino se deve, principalmente, às suas inúmeras características que permitem o desenvolvimento de projetos em uma plataforma de código aberto que integra software e hardware (EDUCBA, 2018). A plataforma Arduino conta com um ambiente de desenvolvimento (IDE) gratuita por meio da linguagem C++ que é amplamente ensinada em instituições de ensino superior (IES).

Figura 8 – Modelos de placa do microcontrolador Arduino. a) Arduino Uno Rev3; b) Arduino Mega 2560 Rev3.



Fonte: a) Arduino; b) Arduino.

A placa Arduino Mega, mostrada na Figura 8b, é recomendada para aplicações com impressoras 3D ou sistemas robóticos mais complexos, pois conta com memória flash de 256KB, oito vezes superior à versão Uno; SRAM de 8KB, o quádruplo da placa Uno e por possuir 54 portas I/O, contra 14 do Arduino Uno. Apesar das vantagens em performance, a adoção dessa placa no projeto seria superdimensionada, já que a quantidade de componentes planejados para o projeto consumiriam menos de um quarto das portas I/O disponíveis. Outra consideração importante nessa análise foi a capacidade de armazenamento e processamento de informações das placas. Para uma aplicação como a que se propõe o presente projeto, cujo volume de dados processados e cálculos processados é baixo, a placa Arduino Uno apresenta especificações suficientes ao desempenho do robô móvel.

#### **4.1.2 Estrutura do robô móvel**

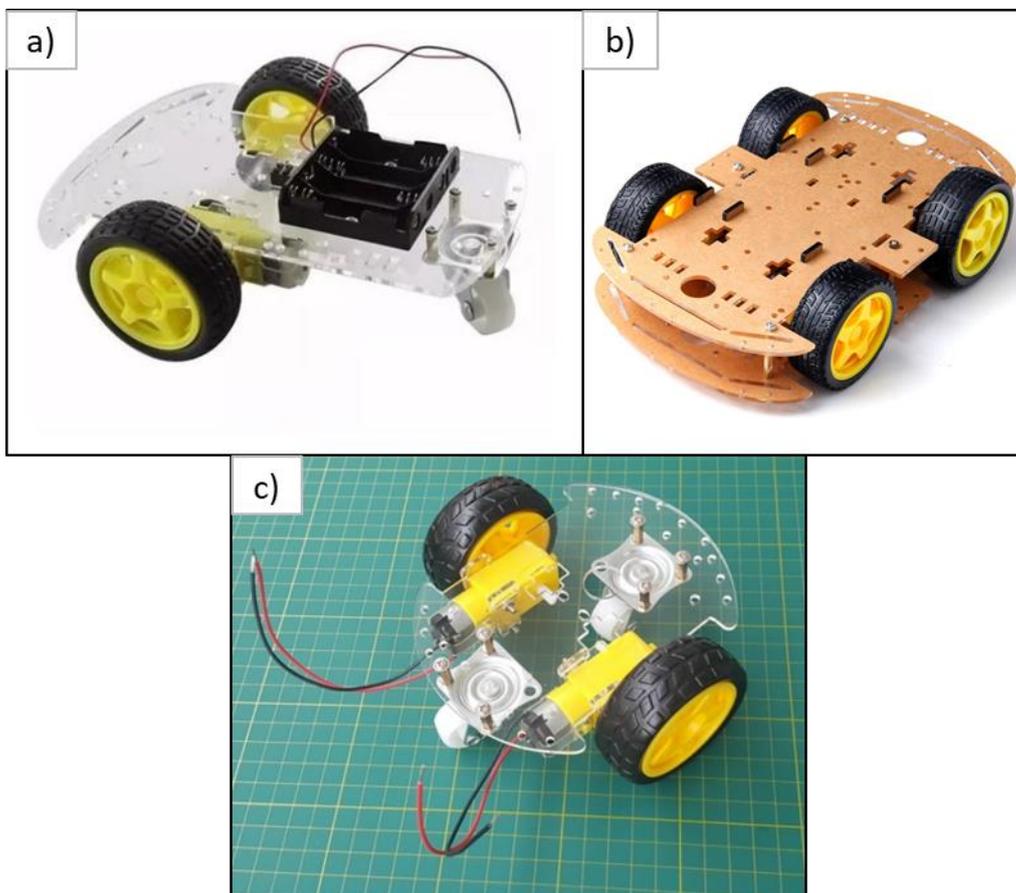
Para a definir o conjunto de sensores e atuadores que seriam incorporados ao robô móvel, foi necessário escolher em primeiro lugar a estrutura física do protótipo. Para fins educacionais e práticos, existem diversas estruturas com diferentes características que são capazes de abrigar todo o aparato eletrônico necessário para movimentá-la. Os tipos de chassis dos robôs móveis variam de acordo com as suas características físicas, como a quantidade de pares de motores DC e rodas; estrutura da base, podendo variar entre um ou mais níveis; desenho da base, com formatos circulares e retangulares; além do próprio material de confecção da estrutura, podendo ser plástica ou de metal.

A escolha da estrutura adequada ao projeto levou em consideração, principalmente, a limitação do número de portas I/O (i.e., portas de entrada e saída) do microcontrolador bem como aspectos físicos dos chassis. Para cada motor DC, responsável pelo movimento das rodas do robô, o driver ponte-H consome duas entradas do Arduino (uma entrada digital e outra analógica para modulação PWM), logo um chassi 4WD demanda 8 portas I/O do microcontrolador. O Arduino Uno dispõe de 14 portas I/O, com as entradas 0 e 1 reservadas para comunicação RT/TX, o que reduz o número de portas a 12. Dessa forma, restariam apenas 4 entradas para realizar a interface com todos os demais sensores.

Devido às limitações do Arduino Uno citadas anteriormente, o chassi 2WD foi escolhido como estrutura para o robô móvel por se adequar ao escopo do projeto.

Entretanto, as rodas bobas que incorporam esse chassi, responsáveis pela estabilidade da estrutura, são mais suscetíveis ao desgaste e mau funcionamento em virtude do possível travamento de suas articulações rotacionais em casos de acúmulo residual em sua estrutura. Tais casos de acúmulo de resíduos devem ser levados em consideração e minimizados devido à proposta de atuar em ambientes como estufas e lavouras, caracterizados como ambientes propícios à eventualidade levantada. Essas restrições de projeto foram fatores determinantes na escolha do chassi 2WD com uma roda boba, exemplificado na figura 9a.

Figura 9 – Exemplos de plataformas. a) Chassi 2WD com estrutura de acrílico e uma roda boba; b) Chassi 4WD com estrutura de acrílico com revestimento adesivo de proteção; c) Chassi 2WD com estrutura circular de acrílico com duas rodas bobas.



Fonte: a) Filipeflop b) Filipeflop; c) Portal do Arduino.

### 4.1.3 Sensoriamento de obstáculos

O sensor de distância HC-SR04 (FIGURA 10) tem como princípio de funcionamento a emissão de ondas sonoras de perfil ultrassônico. O equipamento traz em sua estrutura um conjunto transmissor e receptor de ondas ultrassônicas. Dessa maneira, ao se chocar contra um obstáculo, a onda sonora emitida pelo módulo transmissor é refletida e em seu caminho de volta é detectada pelo módulo receptor do sensor ultrassônico. Considerando a velocidade de propagação do som na atmosfera, e recebendo como informação o tempo decorrido entre a emissão e a recepção, o sensor é capaz de determinar a qual distância do sensor está o obstáculo encontrado.

Figura 10 – Sensor de distância ultrassônico HC-SR04



Fonte: *Cytron Technologies* – HC-SR04 Datasheet (2013).

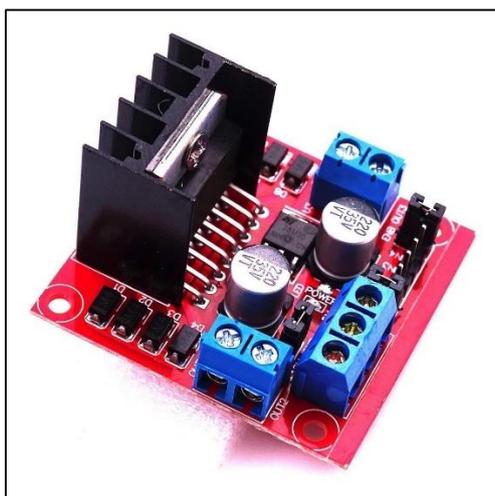
A inserção desse sensor no protótipo garantiu que o robô consiga responder à presença de possíveis obstáculos não mapeados no ambiente, prevenindo colisões durante o percurso.

### 4.1.4 Controle de rotação e direção dos motores

Por ser projetado para operar com alta tensão e alta corrente, o driver de ponte-H dupla L298N (FIGURA 11) aceita níveis lógicos no padrão TTL e cargas indutivas como relés, solenoides, motores DC e motores de passo. Além de permitir controlar o giro do motor por meio da modulação de PWM, o driver também possibilita o controle do sentido de rotação do motor, integrando mais liberdade de movimento ao robô. Outros

componentes eletrônicos, como o CI L293, também permitem o controle PWM do giro de motores DC, mas apenas em um sentido. Ao ser alimentado com uma fonte entre 4V e 35V, o driver cumpre o papel de interface entre os motores DC, responsáveis pelo movimento, e o microcontrolador por meio das portas que permitem modulação PWM.

Figura 11 – Driver ponte-H dupla L298N



Fonte: ST Microelectronics – Driver ponte-H dupla L298N Datasheet (2000).

#### 4.1.5 Alimentação do protótipo

Para garantir maior autonomia ao robô durante o trajeto, foi fundamental garantir que tanto o micricontrolador quanto o driver ponte-H dispusessem de uma fonte de alimentação que atendesse aos seus requisitos de tensão e corrente para o correto funcionamento. Sob tal perspectiva, foi necessário conhecer tanto o microcontrolador quanto o driver ponte-H (TABELA 1).

Tabela 1: Requisitos elétricos de alimentação dos componentes

Componente	Faixa de tensão de alimentação	Faixa de tensão recomendada para alimentação
Arduino Uno	6V a 20V	7V a 12V
Driver L298N	4V a 35V	-

Fonte: ST Microelectronics – Driver ponte-H dupla L298N Datasheet (2000) e Arduino Uno Rev3 – Tech specs (Acessando em 12/10/2019. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>).

A recomendação de alimentação para o microcontrolador Arduino Uno é justificada pela instabilidade da placa quando alimentada com menos de 7V, devido a proximidade com a sua tensão de operação (5.5V), e pelo superaquecimento do regulador de tensão quando alimentada com mais de 12V, podendo danificar a placa.

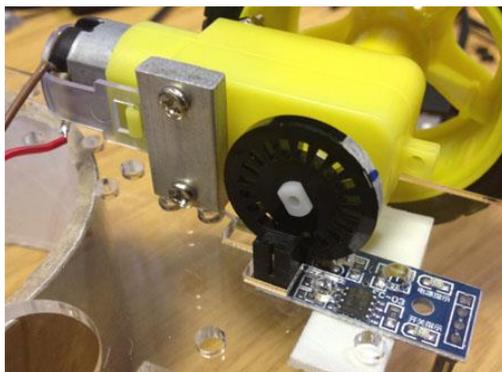
Seguindo tais limitações elétricas de projeto, optou-se por incorporar ao protótipo como fonte de alimentação uma bateria de 9V recarregável, devido a sua disponibilidade no mercado, podendo ser facilmente encontrada tanto em lojas físicas quanto virtuais.

#### **4.1.6 Sensores de velocidade**

Inicialmente projetado para se locomover com base nas iterações do código programado no Arduino, o robô apresentou imprecisão em seus movimentos após os primeiros minutos do início de seu movimento. Conforme a bateria era consumida pelo robô, menor era a tensão de alimentação entregue aos componentes para seu funcionamento pleno, fazendo com que o robô ficasse mais lento em função do tempo. Nesse ponto do projeto, percebeu-se a necessidade de adaptar a lógica de programação do robô, fazendo com que ele passasse a se locomover em função da velocidade de rotação de seus motores e, por consequência, de suas rodas.

Para contornar esse obstáculo, optou-se pelo sensor de velocidade LM393 devido a sua estrutura física compatível e acoplável aos motores DC integrados ao chassi do robô. O LM393 consiste em uma chave óptica composta de uma fenda entre os módulos emissor e receptor. Ao ser alimentado, o módulo emissor dispara um sinal óptico contínuo, através da fenda, até o módulo receptor. Quando um obstáculo é posicionado entre os módulos, o receptor para de receber sinal, caracterizando o funcionamento de uma chave óptica. Para que esse funcionamento seja interpretado como uma medida de velocidade, um disco de *encoder* acoplado à extremidade interna dos motores DC deve realizar seu giro passando pela fenda entre os módulos do sensor, conforme a montagem indicada na Figura 12, fazendo com que o sinal óptico emitido ora passe pelas fissuras do *encoder* ora seja bloqueado por sua estrutura, gerando pulsos ópticos no módulo receptor. A contagem desses pulsos ópticos no LM393 possibilita o cálculo da velocidade angular dos motores do robô, com base nas dimensões físicas das suas rodas.

Figura 12 – Sensor de velocidade LM393 e disco de encoder.



Fonte: Pridopia (Acessado em 15/10/2019. Disponível em: <http://www.pridopia.co.uk/pi-motor-encoder.html>).

#### 4.1.7 Características do protótipo

Após o estudo sobre os componentes necessários ao protótipo, sua montagem foi iniciada inserindo um conjunto de sensores e/ou atuadores por vez. Primeiramente os motores e a ponte-H foram inseridos ao protótipo permitindo estudar o movimento do robô e calibrar o giro das duas rodas. Com os motores calibrados, os demais sensores foram inseridos para compor o protótipo final (FIGURA 13). A etapa de prototipagem também permitiu o dimensionamento do robô, o que possibilitou a correta configuração do algoritmo *fuzzy* (TABELA 2).

Figura 13 – Fotografia do robô móvel



Fonte: Do autor (2019).

Tabela 2: Requisitos elétricos de alimentação dos componentes.

	<b>Dimensionamento</b>
Largura (cm)	15,5 cm
Comprimento (cm)	22 cm
Altura (cm)	14,5 cm

Fonte: Do autor (2019).

#### **4.2 Descrição do problema e ambiente de atuação**

A Universidade Federal de Lavras é tradicionalmente conhecida pela sua relação com o setor agrícola. Projetos desenvolvidos na universidade beneficiam não apenas a população da região de Lavras, mas diversos agricultores e trabalhadores do campo em todo o Brasil. Inseridos nesse ambiente, e observando o amplo potencial de benefício dos projetos do PROMAD à sociedade, construiu-se o ideal de desenvolver uma solução, por meio da robótica móvel, que fosse alinhada ao contexto do referido nicho acadêmico e que se propusesse a sanar dores enfrentadas no dia-a-dia no campo.

Para tanto, optou-se por um ambiente controlado com dimensões inferiores às encontradas em uma aplicação no campo, para estudar o desempenho do veículo na execução de sua função principal: a locomoção. Nesse ambiente, aspectos como a autonomia do protótipo e a contribuição de novos sensores e atuadores para sua locomoção foram observados. O levantamento dessas características do robô móvel foi de extrema importância para avaliar o desempenho do algoritmo *fuzzy* na geração do caminho. O melhor caminho entre dois pontos em uma trajetória está relacionado não apenas à distância entre o ponto de origem e o ponto de chegada, mas também às características do ambiente, como os obstáculos existentes. Partindo desses parâmetros é possível compreender o esforço necessário por parte do protótipo durante o percurso. Esse esforço pode se tornar um fator crítico durante o exercício de locomoção, uma vez que um caminho cujo respectivo esforço requerido seja muito alto, pode apresentar exigências que estejam além das capacidades física e energética do robô. Com tais premissas, foi possível iniciar a análise do desempenho do algoritmo *fuzzy* no planejamento de uma trajetória que atendesse às características e limitações do protótipo e do ambiente.

### 4.3 Melhor caminho e descrição de rota

A realização do planejamento da trajetória foi dividido em duas etapas seguindo a mesma sequência de tarefas, a primeira em um ambiente isolado e a segunda adotando uma estufa como ambiente de trabalho. Para ambas aplicações o primeiro passo foi o mapeamento do ambiente, gerando uma reprodução do local de atuação em uma imagem a ser interpretada pelo algoritmo.

Em sequência, para que o algoritmo fosse capaz de interpretar a imagem gerada, foi preciso submetê-la a um processo de discretização, transformando-a em um mapa de *bits* (*bitmap*). Feito isso, as posições inicial e final da trajetória foram arbitrariamente definidas permitindo que o algoritmo *fuzzy* fosse capaz de interpretar o ambiente e gerar o grafo descrevendo o melhor caminho. Por fim, para que o microcontrolador fosse capaz de entender a rota, o grafo foi submetido a um processo de “tradução” para uma estrutura matricial contendo as coordenadas do percurso desejado.

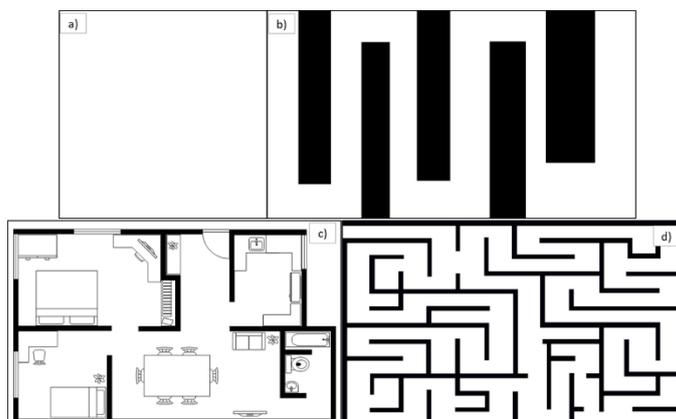
#### 4.3.1 Mapeamento e discretização de ambientes

A fim de testar a robustez do algoritmo *fuzzy*, foi necessário compreender seu desempenho em ambientes distintos, com graus de complexidade variados. Para essa finalidade, representações gráficas de quatro ambientes foram construídas em formato de mapa de bits.

No *bitmap*, os obstáculos são representados pelo *bit* 0 (cor preta) e o caminho livre para locomoção é representado pelo *bit* 1 (cor branca). As representações escolhidas para esse teste foram um ambiente livre de obstáculos, um ambiente com obstáculos retangulares, uma planta baixa de uma casa mobiliada, e finalmente um labirinto (FIGURA 14).

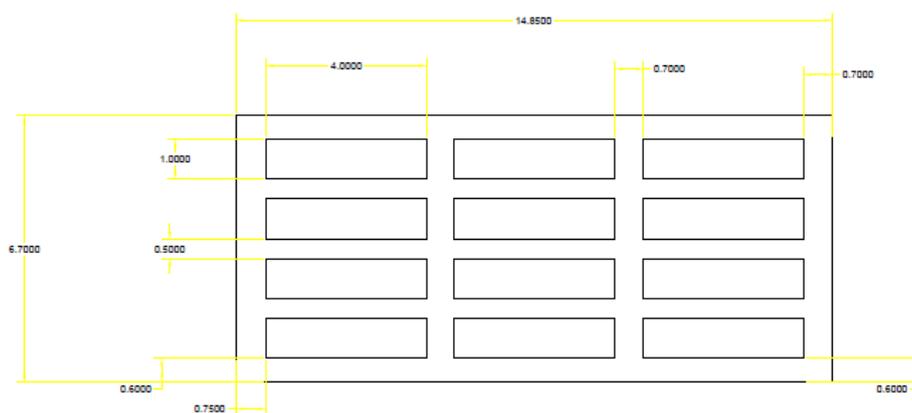
Por fim, alinhando o projeto ao seu objetivo, adotou-se uma estufa localizada no Departamento de Recursos Hídricos da UFLA para estudar o desempenho do robô móvel nessas instalações. Para isso, uma visita ao local foi realizada visando dimensionar a estufa, permitindo a construção de sua planta baixa (FIGURA 15) usada na conversão do ambiente em um *bitmap* (FIGURA 16).

Figura 14 – Exemplos de estruturas para robôs móveis. a) Ambiente livre de obstáculos; b) Ambiente com obstáculos retangulares; c) Planta baixa de uma casa; d) Labirinto



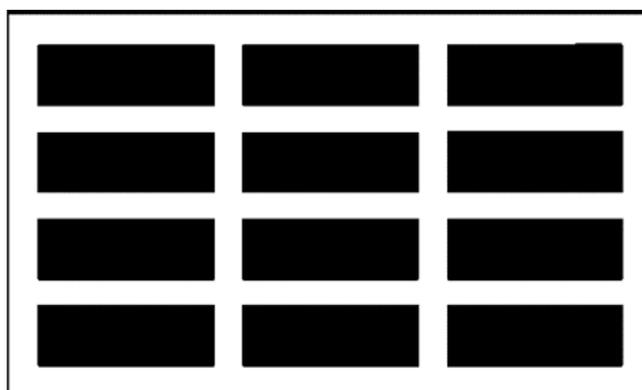
Fonte: a) Do autor (2019); b) Do autor (2019); c) Visual Paradigm; d) Do autor (2019).

Figura 15 – Planta baixa da estufa



Fonte: Bento (2019).

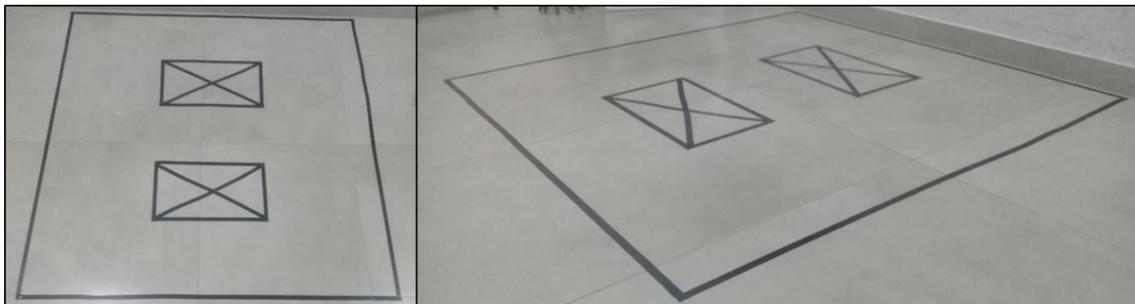
Figura 16 – Representações gráficas da estufa em bitmap



Fonte: Bento (2019).

Devido à indisponibilidade do local para a realização do experimento com a plataforma do robô, foi construída uma representação de uma seção da estufa com dimensões inferiores ao ambiente real (FIGURA 17).

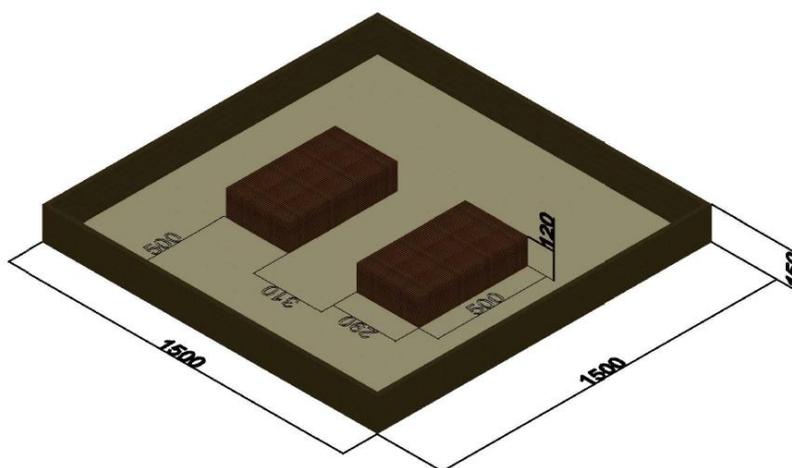
Figura 17 – Reprodução adaptada das instalações da estufa



Fonte: Do autor (2019).

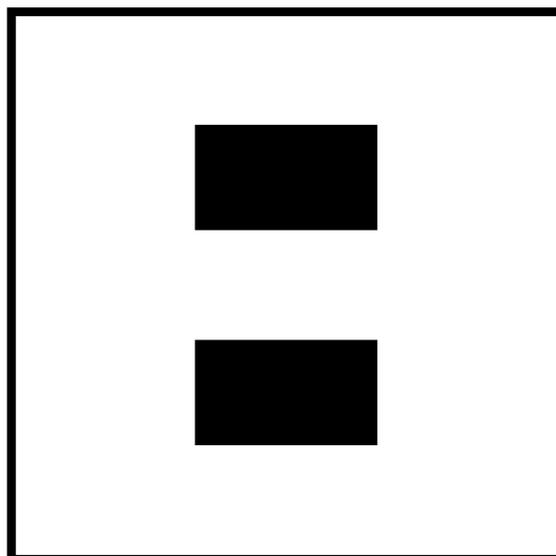
O objetivo desse experimento foi avaliar o desempenho do robô em um ambiente cujas condições físicas fossem semelhantes àsquelas encontradas na estufa. Uma representação gráfica do ambiente foi desenvolvida em 3D com o software Autocad (FIGURA 18) para garantir fidelidade em relação às dimensões reais na representação em *bitmap* após a discretização (FIGURA 19).

Figura 18 – Representação em 3D do ambiente construído



Fonte: Do autor (2019).

Figura 19 – Discretização do ambiente em um bitmap



Fonte: Do autor (2019).

A conversão da representação do ambiente para um *bitmap* caracteriza uma etapa fundamental para a geração do caminho. Após essa etapa, usou-se o algoritmo *fuzzy* proposto por Kala (2014) composto por um conjunto de regras que determina os movimentos do robô frente a um obstáculo.

#### 4.3.2 Algoritmo *fuzzy* e planejamento de rota

Além do mapa de *bits* do ambiente, foi necessário configurar no código do algoritmo algumas características físicas e de movimento da plataforma (FIGURA 20), sendo essas as únicas informações e configurações necessárias para o funcionamento do algoritmo.

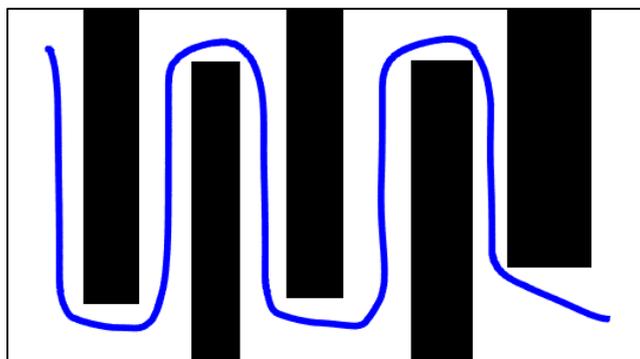
Figura 20 – Configuração inicial do algoritmo *fuzzy*

```
map=int16(im2bw(imread('planta.bmp'))); % planta do ambiente em bmp.
source=[486 249]; % posição inicial em coordenadas Y, X
goal=[87 311]; % posição final em coordenadas Y, X
robotDirection=pi/90; % angulação inicial do robô
robotSize=[3 3]; % largura e comprimento do robô
robotSpeed=500; % velocidade em unidades arbitrárias
maxRobotSpeed=1; % velocidade máxima em unidades arbitrárias
S=1; % distância(margem) de segurança em torno do robô
distanceThreshold=10; % distância limítrofe (pontos dentro desse limite são enxergados como um mesmo ponto.
maxAcceleration=4; % máxima aceleração em unidades arbitrárias
directionScaling=45*pi/90; % angulação máxima em curvas (nesse caso limitando a 45°)
```

Fonte: Adaptado de Kala (2014).

As configurações necessárias são relacionadas às características físicas e de movimento do protótipo como as dimensões e velocidade do robô e também o ângulo de direção inicial do robô. Além disso, o algoritmo também permite restringir o ângulo máximo de rotação do robô nas curvas durante o percurso, permitindo descrever rotas com curvas mais suaves ou mais fechadas. Para garantir a segurança do robô, outro parâmetro configurado para o planejamento do caminho foi uma distância de segurança para prevenir colisões, criando uma representação inflada do protótipo. Esse parâmetro de segurança deve ser ajustado conforme as dimensões do ambiente, uma vez que já se conhece as dimensões do robô. Vale ressaltar que as dimensões do robô que foram informadas ao algoritmo representam o número de pixels, ou pontos no *bitmap*, que representam a estrutura física do robô. Assim, ao considerar a distância de segurança o número de pixels que representam o protótipo passou a ser maior e o algoritmo garantiu que a rota descrita era livre de colisões. Os últimos parâmetros configurados no algoritmo foram os pontos de partida e chegada do percurso. A partir desse ponto, o algoritmo *fuzzy* iniciou seu processo de tomada de decisão se orientando pelo ângulo entre a direção do robô e o ponto de chegada com base em seu conjunto de regras *fuzzy*, gerando o melhor caminho com base em suas regras (FIGURA 21).

Figura 21 – Exemplo planejamento de caminho pelo algoritmo *fuzzy*



Fonte: Do autor (2019).

### 4.3.3 Tradução de rota e navegação

Como o propósito do planejamento do caminho é a sua interpretação e execução por parte do robô, foi necessário programar uma interface de tradução da imagem gerada com o grafo para o microcontrolador embarcado no protótipo, pois o Arduino não possui ferramentas de interpretação ou manipulação de imagens. Entretanto, o mapa de *bits* do

ambiente contendo o grafo pode ser, facilmente, convertido em uma estrutura matricial suportada pelo microcontrolador. O processo de tradução baseou-se no mapa de *bits* de imagens na escala de cinza *grayscale 256 bits* (FIGURA 22), no qual a cada nível de saturação de cinza entre as cores preta e branca é atribuído um valor numérico entre 0 e 255, respectivamente.

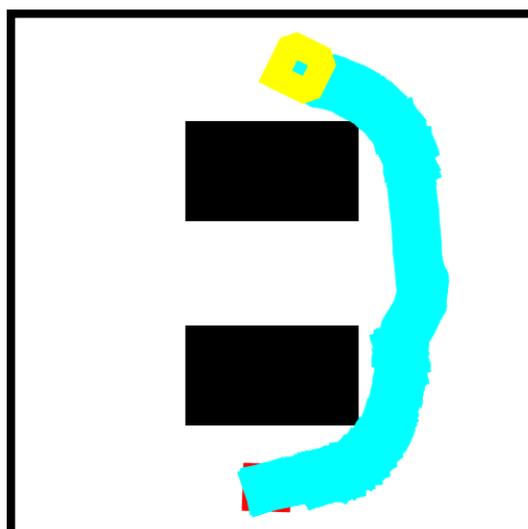
Figura 22 – Escala de cinza representada em 256 bits



Fonte: Adaptado de Leica Microsystems.

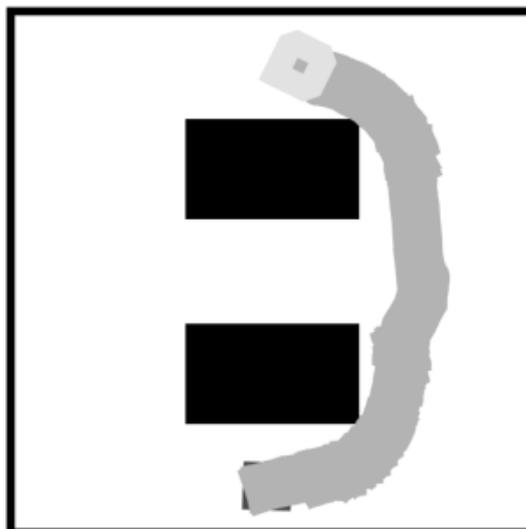
Para o sucesso dessa tradução, algumas alterações foram realizadas no algoritmo original para que o *bitmap* fosse gerado em escala de cinza diferenciando os cinco elementos necessários para a interpretação: o caminho livre, os obstáculos, os pontos de partida e chegada e o trajeto a ser percorrido. Assim, considerando o ambiente construído para o experimento, foi preciso ajustar as cores originais da imagem, de forma que todas apresentassem contraste o suficiente entre si (FIGURA 23) para serem diferenciadas umas das outras em tons de cinza (FIGURA 24).

Figura 23 – Resultado da adaptação das cores no algoritmo



Fonte: Do autor (2019).

Figura 24 – Conversão da imagem em bitmap em escala de cinza



Fonte: Do autor (2019).

Após a conversão da imagem, um segundo código foi desenvolvido para armazenar a informação da imagem em um arquivo de texto para realizar as seguintes adaptações no mapa de *bits*:

- a) Todos os pixels que representavam obstáculos (cor preta, valor 0) foram substituídos pelo valor -1, Esse valor foi escolhido para minimizar sua influência na interpretação do ambiente no código do robô;
- b) Analogamente ao passo anterior, posteriormente todos os pixels que indicavam caminho livre (cor branca e valor 255 na escala de cinza) foram substituídos pelo valor 0;
- c) Ajuste de escala do *bitmap* para o ambiente real: antes os pixels representavam milímetros da escala real, o ajuste corrige o mapa para escala de centímetros em função das dimensões do robô. Essa manipulação reduz as dimensões da estrutura matricial e, por consequência, o processamento dessas informações no microcontrolador;
- d) Armazenamento dos índices da escala de cinza em um arquivo de texto formatado de acordo com a sintaxe de vetores bidimensionais em C++;

O resultado dessa manipulação foi o mapa do ambiente formatado em uma estrutura que o Arduino é capaz de entender e manipular. Por se tratar de um arquivo de texto, o resultado dessa manipulação também pôde ser observado por meio do software Excel que, com o auxílio da ferramenta de Formatação Condicional, permitiu analisar o resultado dessa tradução de forma semelhante a um mapa de calor. Com essa tradução, o desafio da prototipagem foi a interpretação das células que descrevem o caminho, destacado em verde no mapa (FIGURA 25).

Figura 25 – Mapa do ambiente em padrão textual representada no Excel

34	21	21	21	21	21	21	21	21	21	21	31	26	21	21	21	21	21	21	24
21	0	0	0	0	0	0	0	0	0	6	200	197	70	2	0	0	0	0	8
21	0	0	0	0	0	0	0	0	0	83	223	220	191	159	55	0	0	0	8
21	0	0	0	0	0	0	0	0	0	2	80	162	177	179	178	64	0	0	8
21	0	0	0	0	0	9	17	17	17	17	17	37	141	179	169	13	0	0	8
21	0	0	0	0	0	6	-1	-1	-1	-1	-1	-1	35	175	179	57	0	0	8
21	0	0	0	0	0	6	-1	-1	-1	-1	-1	-1	13	144	179	68	0	0	8
21	0	0	0	0	0	6	-1	-1	-1	-1	-1	-1	13	116	179	59	0	0	8
21	0	0	0	0	0	5	18	18	18	18	18	18	7	97	179	81	0	0	8
21	0	0	0	0	0	0	0	0	0	0	0	0	0	84	179	98	0	0	8
21	0	0	0	0	0	0	0	0	0	0	0	0	0	72	179	130	0	0	8
21	0	0	0	0	0	0	0	0	0	0	0	0	0	127	179	107	0	0	8
21	0	0	0	0	0	8	4	4	4	4	4	4	4	63	179	181	22	0	8
21	0	0	0	0	0	6	-1	-1	-1	-1	-1	-1	61	179	180	7	0	0	8
21	0	0	0	0	0	6	-1	-1	-1	-1	-1	-1	78	179	153	0	0	0	8
21	0	0	0	0	0	9	13	13	13	13	13	16	138	179	110	0	0	0	8
21	0	0	0	0	0	0	0	0	0	0	14	62	133	179	179	49	0	0	8
21	0	0	0	0	0	0	0	0	0	34	145	178	179	179	179	119	0	0	8
21	0	0	0	0	0	0	0	0	0	22	174	175	173	151	94	7	0	0	8
24	8	8	8	8	8	8	8	8	11	53	22	11	8	8	8	8	8	8	12

Fonte: Do autor (2019).

#### 4.3.4 Configuração do robô móvel

Para conseguir descrever a trajetória do robô, dois códigos foram desenvolvidos e embarcados no microcontrolador, sendo um código principal e uma biblioteca voltada à movimentação das rodas, configurando um *sketch* do Arduino. No código principal, um conjunto de funções interdependentes foi desenvolvido permitindo que o algoritmo interpretasse o mapa de *bits*. No mapa, além dos índices do caminho, do espaço livre e dos obstáculos, outros índices foram parametrizados durante a realização do percurso (FIGURA 26) para auxiliar o robô a identificar os pontos de partida e chegada e também

as células pelas quais ele já passou, evitando o possível problema de uma célula passada, mas com alto índice na tabela, seja escolhida novamente pelo algoritmo, fazendo o robô andar para trás (FIGURA 27).

Figura 26 – Parametrização de índices e tratamento da rota em código

```
//0 - Caminho livre
//1 - Caminho planejado
//-1 - Obstáculo
//-2 - Caminho já percorrido
//-11 - Ponto de partida
//1111 - Ponto de chegada
//-99 - posição atual do robô
```

Fonte: Do autor (2019).

Figura 27 – Exemplo do mapa tratado em tempo de execução de código no Serial do Arduino

14	14	14	14	14	14	14	14	14	14	14
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	1111	116	8	0	0	0
14	14	14	14	14	14	75	155	0	0	0
-1	-1	-1	-1	-1	-1	5	101	77	0	0
-1	-1	-1	-1	-1	-1	5	1	151	17	0
3	3	3	3	3	3	6	0	105	47	0
0	0	0	0	0	0	0	0	121	26	0
0	0	0	0	0	0	0	0	148	0	0
0	0	0	0	0	0	0	18	137	0	0
19	19	19	19	19	19	5	113	49	0	0
-1	-1	-1	-1	-1	-1	5	159	0	0	0
-1	-1	-1	-1	-1	-1	5	-99	0	0	0
-1	-1	-1	-1	-1	-1	26	-2	0	0	0
16	16	16	16	16	16	-2	70	0	0	0
0	0	0	0	27	86	-2	2	0	0	0
0	0	0	-2	-2	-2	7	0	0	0	0
0	0	32	-2	0	0	0	0	0	0	0
14	14	-11	31	14	14	14	14	14	14	14

Fonte: Do autor (2019).

Com essa estrutura de tratamento, o código também possui variáveis globais declaradas para receber, manualmente, os valores gerados pelo código que foi desenvolvido em Matlab. O código, conforme já foi mencionado anteriormente, é capaz de formatar todo o mapa de *bits* de acordo com a sintaxe de matrizes em C++ (FIGURA 28), permitindo que o usuário do código apenas cole essa informação no código do Arduino antes de carregá-lo à placa (FIGURA 29). Outras informações que o código em Matlab fornece são as coordenadas já redimensionadas dos pontos de partida e chegada do robô na tabela.

Figura 28 – Arquivo de texto com as informações do ambiente para interpretação do Arduino

```
Resolução do ambiente: 20x20ur
i source: 20
j source: 10
i goal: 4
j source: 13
{{12,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,14,20},{4,0,0,0,0,0,
```

Fonte: Do autor (2019).

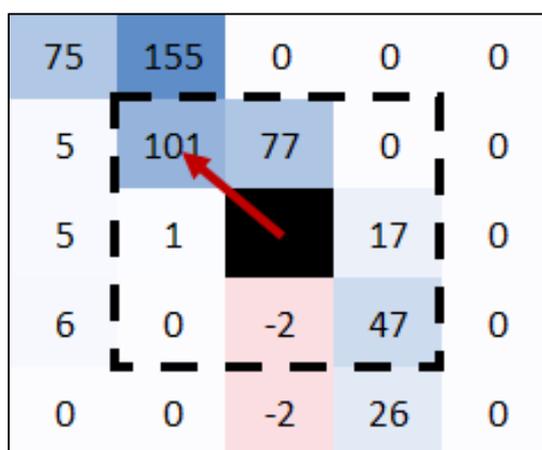
Figura 29 – Inserção dos parâmetros do mapa no código do Arduino

```
//Parâmetros que são dados pelo código conv.m
int matriz[h][l] = {{12, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 14, 20}, {4, 0, 0,
int i_source = 20;
int j_source = 10;
int i_goal = 4;
int j_goal = 13;
```

Fonte: Do autor (2019).

Para interpretar a rota inserida no código e realizar o movimento, o código realiza uma varredura dos valores das células ao seu redor (FIGURA 30), seleciona o maior valor encontrado e retorna o ângulo de curvatura necessário para seguir pelo caminho planejado por meio de funções da biblioteca *rodas.h* (FIGURA 31). Devido ao tratamento dos valores do mapa, é garantido que nessa etapa os valores mais altos no mapa representam o caminho a ser seguido.

Figura 30 – Exemplo da varredura para escolha do caminho



Fonte: Do autor (2019).

Figura 31 – Trecho do código responsável pela interpretação da varredura

```

//Função que seleciona o maior valor nas células adjacentes
//Retorna o ângulo de curvatura a ser feito (0, 45, 90, -45, -90 ou 180)
//Caso o ângulo seja zero, nenhuma ação é feita e o robô segue reto
angulo_tabela = escolha(i_atual, j_atual, angulo_atual);
//Cálculo do ângulo da curva em relação à angulação inicial do movimento
turn = angulo_tabela - angulo_atual;
//Parâmetro de atualização do ângulo entre loops
angulo_atual = angulo_tabela;

if(turn==0)
{ if(angulo_tabela==90 || angulo_tabela==90 || angulo_tabela==0)
  { reto(ref_vel_rodas);
  }
  else if(angulo_tabela==45 || angulo_tabela==45 || angulo_tabela==135 || angulo_tabela==135)
  { diagonal(ref_vel_rodas);}
}
if(turn==45)
{ dir_45();
  if(angulo_tabela==90 || angulo_tabela==90 || angulo_tabela==0)
  { reto(ref_vel_rodas);}
  else if(angulo_tabela==45 || angulo_tabela==45 || angulo_tabela==135 || angulo_tabela==135)
  { diagonal(ref_vel_rodas);}
}
if(turn == -90)
{ dir_90();
  if(angulo_tabela==90 || angulo_tabela==90 || angulo_tabela==0)
  { reto(ref_vel_rodas);}
  else if(angulo_tabela==45 || angulo_tabela==45 || angulo_tabela==135 || angulo_tabela==135)
  { diagonal(ref_vel_rodas);}
}
}

```

Fonte: Do autor (2019).

## 5 RESULTADOS E DISCUSSÃO

Os resultados apresentados nessa seção têm origem nas simulações realizadas com o algoritmo *fuzzy* nos diferentes ambientes propostos. Por se tratar de um planejamento de caminho *off-line* não houve aprofundamento em análises sobre o tempo de processamento do algoritmo. A avaliação foi dividida em duas etapas, sendo que a primeira consiste na avaliação do algoritmo em quatro ambientes sob as mesmas configurações iniciais. Nessa etapa foi possível tirar algumas conclusões sobre o método de decisão do algoritmo *fuzzy*, como por exemplo a opção do algoritmo em descrever caminhos mais suaves e com menos desvios de rota sob configurações mais cautelosas. Já a segunda consiste na simulação com base em um ambiente real, configurando o algoritmo de acordo com as características físicas do robô móvel, objetivando a realização do percurso planejado.

Para analisar o planejamento do caminho, durante a etapa de simulações o algoritmo foi analisado sob diferentes condições. Aplicou-se, como limitação, valores distintos para o ângulo máximo de curva do robô, além da variação da velocidade máxima de movimento.

### 5.1 Análise do algoritmo *fuzzy* em ambiente livre de obstáculos

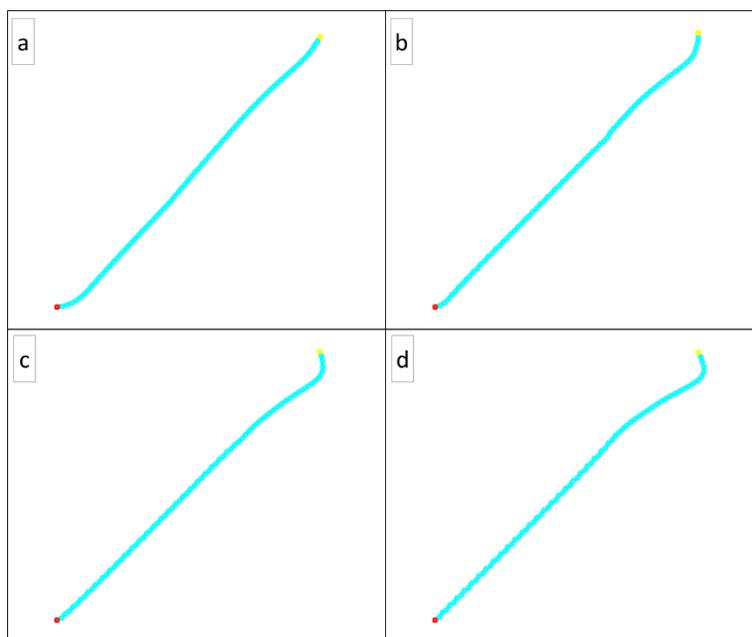
A princípio, para mostrar a influência da variação do ângulo máximo de curva, o algoritmo foi configurado para esse cenário mantendo todas as demais características constantes e velocidade máxima igual a 1, em unidades arbitrárias. O primeiro ângulo testado foi o de 15°, que caracteriza um movimento com curvas mais abertas e suaves. Ao comparar os demais ângulos adotados (30°, 45° e 60°) ao primeiro (FIGURA 32) percebeu-se que quanto maior o ângulo, menos retilíneo era o caminho planejado. Isso se deve ao fato de haver maior grau de liberdade ao movimento. Para ângulos menores, nesse cenário, se torna mais difícil recuperar o caminho em caso de desvio de trajeto. Além disso, quando configurado com ângulos maiores, o tempo de processamento do algoritmo chega a ser 16% menor. Esse comportamento pode ser atribuído à sensibilidade do algoritmo *fuzzy* diante das limitações, evitando realizar curvas ou tentando se desviar menos da rota, consumindo mais tempo de processamento para garantir o sucesso do planejamento. Já com ângulos maiores, há maior liberdade do algoritmo na escolha do traçado do caminho, fazendo com que o processamento seja mais rápido, pois as escolhas perante as regras *fuzzy* são mais simples e com mais liberdade dinâmica. Entretanto, se por um lado o tempo de processamento se torna menor para ângulos maiores, o mesmo não ocorre com o comprimento do caminho. Para ângulos de curva menores, o caminho traçado se aproxima mais de uma reta, devido às limitações de movimento, fato que pode ser descrito segundo Pitágoras. Os resultados dessa primeira simulação estão expostos na Tabela 3.

Tabela 3: Resultados das simulações com o algoritmo *fuzzy* para diferentes parâmetros

<b>Ângulo máximo de curva</b>	<b>Tempo de processamento (s)</b>	<b>Comprimento do caminho</b>	<b>Resolução do mapa</b>
15°	3,7249	39,7225	300x300
30°	3,4748	58,4749	300x300
45°	3,0329	82,3936	300x300
60°	3,1417	111,4818	300x300

Fonte: Do autor (2019).

Figura 32 – Trajetórias segundo o algoritmo *fuzzy* para diferentes ângulos máximos de curva. a) 15°; b) 30°; c) 45°; d) 60°



Fonte: Do autor (2019).

## 5.2 Análise do algoritmo *fuzzy* frente a obstáculos paralelos

Nesse ambiente, foram inseridos cinco obstáculos retangulares dispostos paralelamente. Existe apenas um caminho possível para realizar o percurso contornando os obstáculos. Assim como na avaliação anterior, aqui também variou-se o ângulo máximo de curva. Entretanto, inicialmente os testes foram feitos com velocidade máxima igual a 5. No primeiro teste, com ângulo igual a 60° o algoritmo não conseguiu encontrar um caminho e parou para evitar uma colisão (FIGURA 33a), isso se deve ao fato de que ao encontrar o primeiro obstáculo à frente, o algoritmo optou por fazer uma curva brusca, devido ao seu alto grau de liberdade, e se desviou do único caminho válido, parando próximo à borda do mapa. Algo semelhante ocorreu quando o ângulo configurado foi de 15°. Diferente do caso anterior, aqui o algoritmo não tinha liberdade suficiente para reagir rapidamente a um obstáculo próximo, por essa razão, esse foi o único cenário nesse mapa em que uma colisão foi reportada (FIGURA 33c). Nesse cenário, o único ângulo que apresentou sucesso no planejamento foi o de 45° (FIGURA 33b), o que indica que a correta restrição do ângulo máximo de curva é um fator fundamental para o sucesso do planejamento. Em seguida, para verificar a relação da velocidade máxima com o sucesso

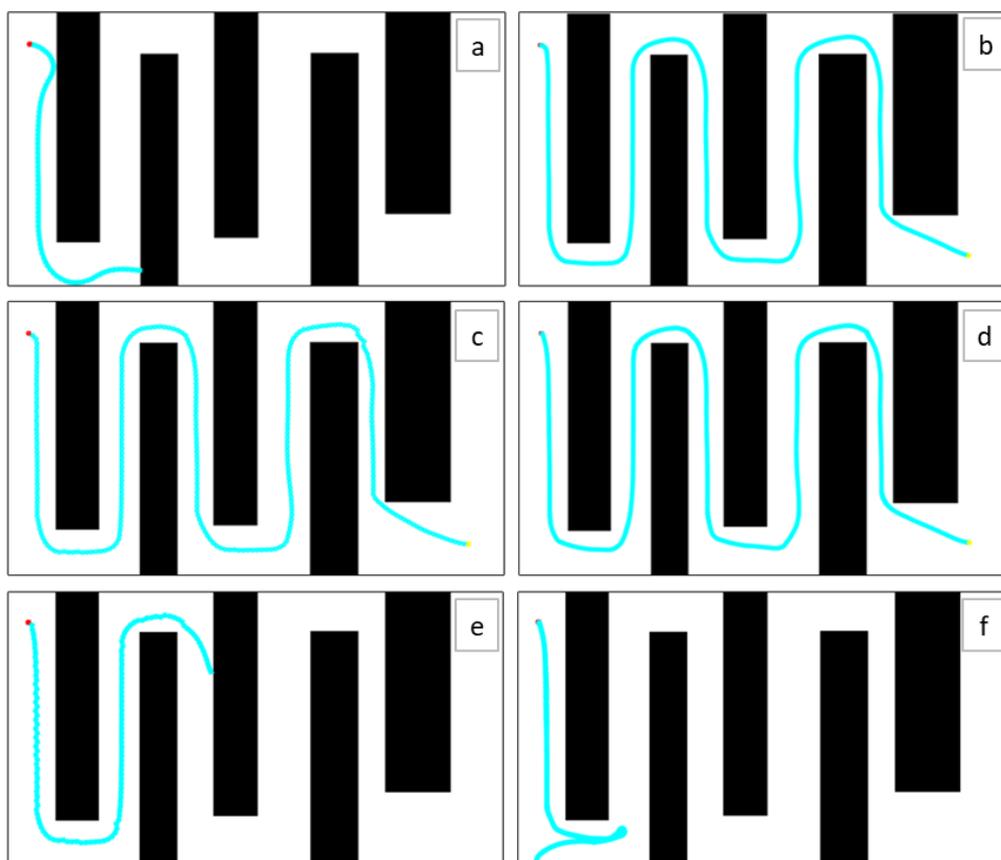
do planejamento, os mesmos testes foram replicados utilizando a velocidade máxima igual a 1. Os resultados dessa análise em comparação com a anterior estão na Tabela 4.

Tabela 4: Resultados das simulações com o algoritmo *fuzzy* para diferentes ângulos e velocidades máximas

Ângulo máximo de curva	Velocidade máxima	Tempo de processamento (s)	Comprimento do caminho	Resolução do mapa
15°	1	121,0190	1528,6720	530x300
15°	5	Colisão reportada	-	530x300
45°	1	121,7989	1504,4900	530x300
45°	5	18,5898	1569,9240	530x300
60°	1	Parou para evitar colisão	-	530x300
60°	5	Parou para evitar colisão	-	530x300

Fonte: Do autor (2019).

Figura 33 – Trajetórias para diferentes ângulos máximos de curva. a) 15°, vel. máx. = 5; b) 15°, vel. máx. = 1; c) 45°, vel. máx. = 5; d) 45°, vel. máx. = 1; e) 60°, vel. máx. = 5; f) 60°, vel. máx. = 1



Fonte: Do autor (2019).

### 5.3 Análise do algoritmo *fuzzy* em uma planta baixa

Ao testar o algoritmo utilizando a representação de uma planta baixa como mapa, foi possível testar o seu desempenho em um cenário com obstáculos de diferentes formatos. No teste anterior foi possível perceber que frente a muitos obstáculos, ângulos maiores tendem a não encontrar um caminho. Por essa razão, nesse cenário as análises ficaram restritas aos ângulos de 15° e 45°, mantendo a variação de velocidade máxima. A Figura 34 reafirma que para ângulos menores, nesse caso 15°, o algoritmo *fuzzy* se torna mais cauteloso nas suas decisões, realizando curvas mais suaves devido a sua limitação em curvas. Ao ser testado com o ângulo de 45° o algoritmo apresentou colisão quando a velocidade máxima era maior. Isso se deve ao fato de que ângulos maiores permitem curvas mais bruscas que, em alta velocidade, são mais difíceis de controlar ou de se recuperar após um desvio. Foi possível observar também, como mostra a Tabela 5, que velocidades mais baixas estão relacionadas a caminhos mais curtos, apesar do tempo de processamento, esse também foi um resultado observado nos testes anteriores. Outra observação feita nessa análise foi sobre o traçado do percurso indicado. Ao comparar os resultados obtidos com o ângulo de 15° foi possível constatar que com a velocidade igual a 1 o algoritmo se comportou de forma mais estável, gerando um percurso com menos curvas que chega a ser 6% mais curto do que o caminho gerado para a velocidade máxima igual a 5.

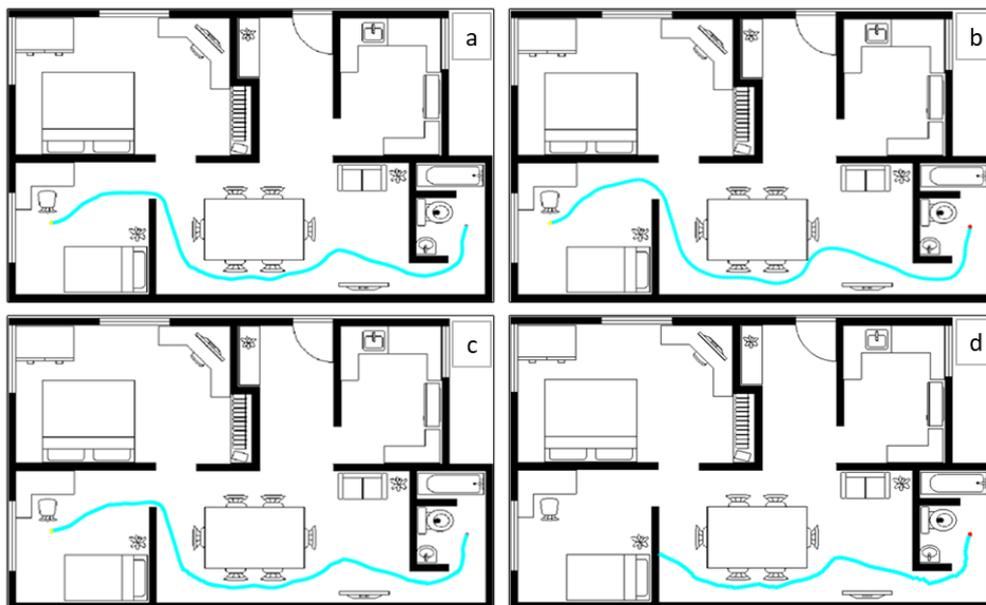
Tabela 5: Resultados das simulações com o algoritmo *fuzzy* frente a obstáculos de formatos diversos

Ângulo máximo de curva	Velocidade máxima	Tempo de processamento (s)	Comprimento do caminho	Resolução do mapa
15°	1	31,4658	674,1902	707x439
15°	5	5,4479	717,2179	707x439
45°	1	30,2867	674,6175	707x439
45°	5	Colisão reportada	-	707x439

Fonte: Do autor (2019).

Figura 34– Trajetórias segundo o algoritmo *fuzzy* para diferentes ângulos e velocidades.

a) 15°, vel. máx. = 1; b) 15°, vel. máx. = 5; c) 45°, vel. máx. = 1; d) 45°, vel. máx. = 5.



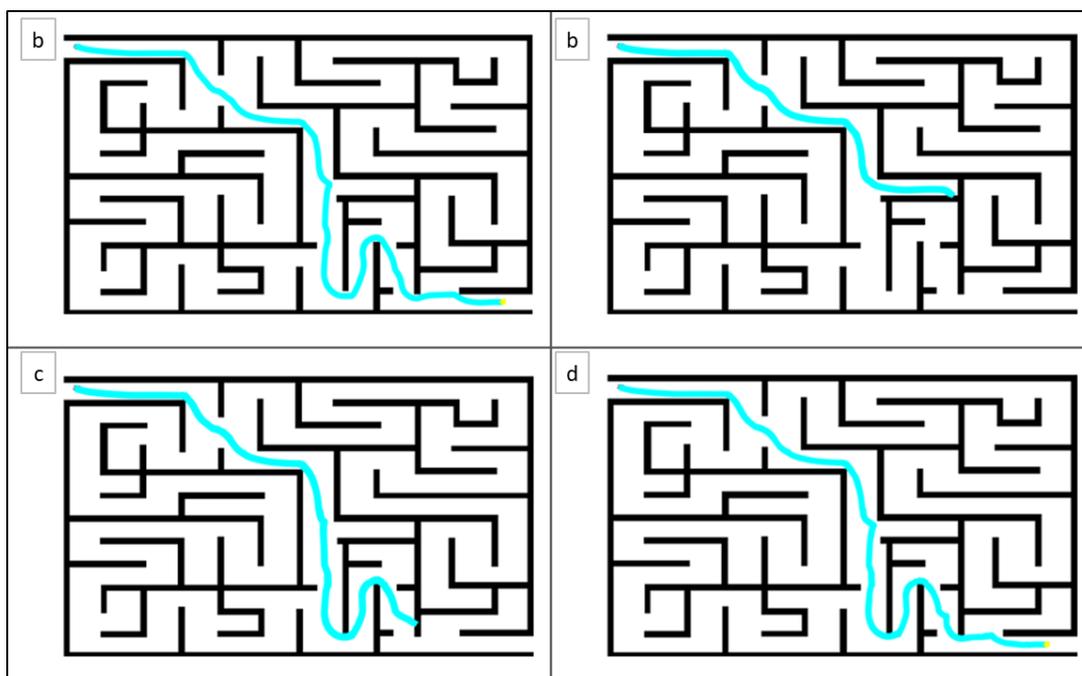
Fonte: Do autor (2019).

#### 5.4 Análise do algoritmo *fuzzy* em um labirinto

Nessa última análise, por se tratar de um labirinto, portanto um mapa que apresenta maior dificuldade que os anteriores, uma análise mais voltada a encontrar o melhor caminho foi conduzida. Já conhecida a relação da velocidade máxima do robô com o comportamento do algoritmo, para esse cenário foi utilizada apenas a velocidade igual a 1. Isso porque nos testes anteriores, os resultados sob essa configuração apresentaram menor comprimento, além de serem visualmente mais retilíneos. Dessa forma, ao variar o ângulo máximo permitido em curvas, foi observado que para os ângulos mais intermediários (30° e 45°) o algoritmo não conseguiu realizar o planejamento do caminho, parando para evitar colisão. Isso pode ser explicado pela ausência dos comportamentos extremos, como é o caso dos ângulos de 60° e 15°. No caso do ângulo de 60°, o que se observou é que o algoritmo é capaz de responder melhor à complexidade do ambiente por ser capaz de mudar o sentido do trajeto com mais facilidade. Já no caso do ângulo de 15°, exatamente por não possuir a mesma facilidade que o caso anterior, o algoritmo foi capaz de tomar decisões com cautela devido a sua limitação em curvas. A Figura 35 mostra o resultado obtido no planejamento de caminho no labirinto. Para avaliar o comprimento do percurso, a Tabela 6 traz a comparação entre os resultados obtidos

nessa análise. O que se constatou foi que mesmo o algoritmo tendo reagido bem à configuração de 15°, o resultado obtido ao configurar o ângulo em 60° foi melhor tanto no comprimento do caminho quanto no tempo de processamento, o que mostra que em ambientes muito complexos a capacidade de reagir mais rapidamente a um obstáculo contribui mais para o resultado do que a capacidade de gerar caminhos mais cautelosos e com curvas mais suaves.

Figura 35 – Trajetórias segundo o algoritmo *fuzzy* para diferentes ângulos em alta complexidade. a) 15°; b) 30°; c) 45°; d) 60°



Fonte: Do autor (2019).

Tabela 6: Resultados das simulações com o algoritmo *fuzzy* frente a obstáculos de formatos diversos

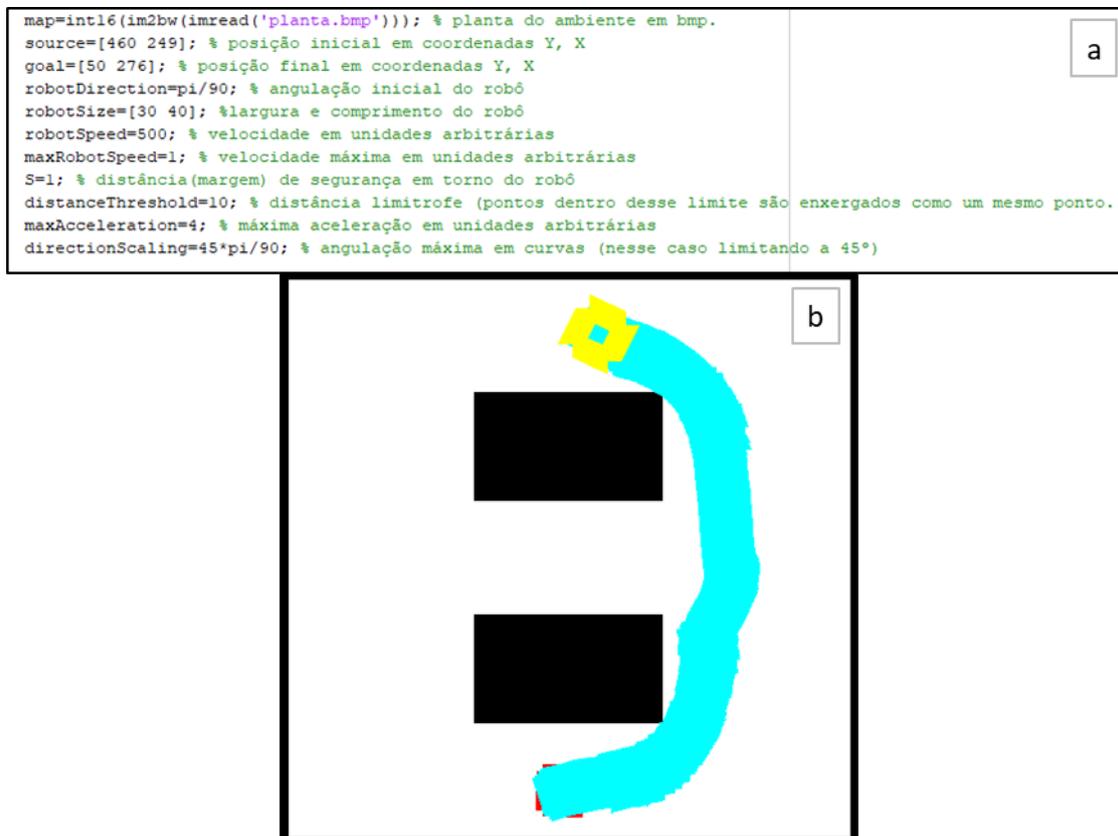
Ângulo máximo de curva	Tempo de processamento (s)	Comprimento do caminho	Resolução do mapa
15°	257,0905	2076,9700	1280x768
30°	Parou para evitar colisão	-	1280x768
45°	Parou para evitar colisão	-	1280x768
60°	197,1482	2066,6390	1280x768

Fonte: Do autor (2019).

## 5.5 Aplicação do algoritmo *fuzzy* em ambiente estático

Diferentemente das análises anteriores, nessa etapa o projeto se objetivou a aplicar a trajetória planejada ao robô móvel e observar a capacidade do código embarcado no microcontrolador de interpretar e reproduzir a rota proposta. Para tanto, aplicou-se o algoritmo *fuzzy* para o planejamento do caminho, configurando-o de acordo com as características do robô móvel. A Figura 36a mostra o código em Matlab com as características já adequadas à unidade de pixels, enquanto a Figura 36b mostra o caminho derivado dessa configuração. Devido às dimensões do robô em relação ao mapa original, no código optou-se por reduzir a margem de segurança em torno do robô para não limitar a possibilidade do algoritmo explorar caminhos em sua tomada de decisão. As análises realizadas anteriormente contribuíram para a escolha dos valores de parametrização do algoritmo. O planejamento durou 52,0313 segundos e gerou um percurso de comprimento igual a 502,844 em pixels.

Figura 36 – Etapas do planejamento do caminho. a) Configuração do algoritmo *fuzzy*; b) Caminho descrito pelo algoritmo.

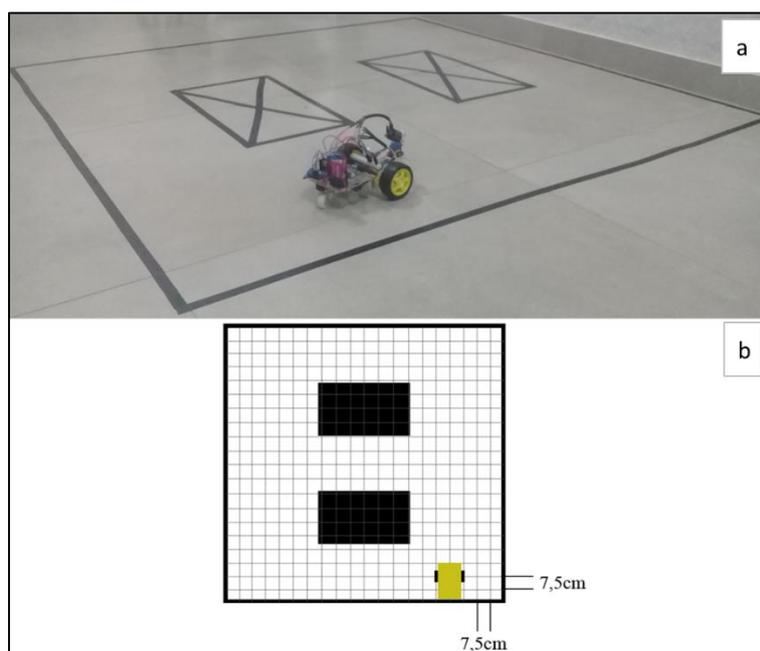


Fonte: Do autor (2019).

Ao analisar a trajetória descrita na Figura 36b foi possível constatar que, diferente das simulação anteriores, nesse cenário o caminho traçado pelo algoritmo se aproxima muito dos obstáculos nos instantes em que foi necessário realizar um curva. Isso se deve a baixa distância de segurança configurada no código. Apesar de se tratar do mesmo valor absoluto, no código  $S=1$ , nas simulações considerava-se um robô cujas dimensões gráficas eram de 3x3 pixels, assim, a distância de segurança era equivalente a um terço das dimensões do robô. Já no caso da aplicação em questão, a mesma margem de segurança é muito menos representativa em relação à plataforma, logo tais parâmetros de projeto resultaram em mais liberdade nas escolhas do algoritmo. Esse comportamento também está associado à escolha do ângulo máximo permitido em curvas, pois ângulos menores tendem a gerar caminhos com curvas mais suaves e mais cautelosas em relação aos obstáculos. Nesse cenário, entretanto, optou-se pelo ângulo de  $45^\circ$  para permitir que o robô se movimente melhor, uma vez que as dimensões do ambiente são um fator limitante se comparadas às dimensões da plataforma.

Em sequência, os códigos desenvolvidos em Matlab foram responsáveis por redimensionar a representação do ambiente, ajustando-o para uma resolução mais fácil de ser tratada pelo Arduino. A conversão realizada ajusta as dimensões do mapa de 1500x1500mm (equivalentes a 507x507 pixels no *bitmap*) para 20x20 células, cada célula com altura e largura equivalentes a 7,5cm do mapa real, como mostra a Figura 37.

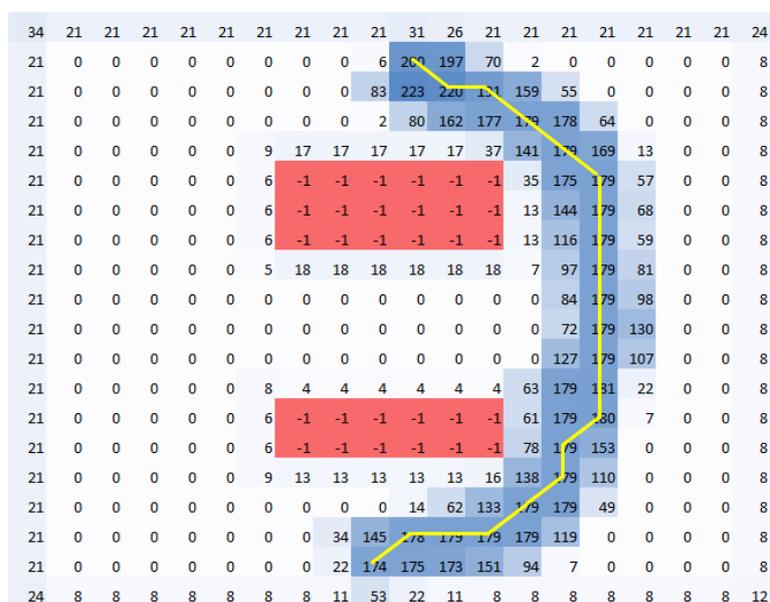
Figura 37 – Plataforma robótica e mapa construído



Fonte: Do autor (2019).

Logo, uma interpretação do mapa foi implementada no código do microcontrolador, objetivando-se a buscar pela célula de maior valor dentre as células adjacentes à posição do robô. A cada célula escolhida, o código instrui o robô sobre a angulação necessária para realizar curvas e, a cada iteração do código, permite que a plataforma avance a distância necessária para que ela atinja a próxima posição do mapa. Após essa interpretação, a plataforma executou um trajeto de 174,8cm de comprimento, esboçado na Figura 38, em uma média de 11,304 segundos.

Figura 38 – Trajeto interpretado pelo código no Arduino



Fonte: Do autor (2019).

O robô executou repetições do trajeto para que fosse possível observar seu desempenho. Dessa repetição de execuções foi possível perceber que, devido à superfície muito lisa do ambiente construído, nas primeira execuções as rodas do robô apresentaram deslizamento na partida, o que comprometia sua trajetória impedindo-o de executá-la com sucesso. Após a remoção de poeira da superfície e limpeza das rodas, esse problema foi mitigado permitindo que o robô percorresse o caminho conforme o esperado. Outro evento identificado foi a interferência da roda boba no alinhamento da plataforma durante o percurso. Após realizar uma curva, a inclinação da roda boba exercia influência sobre o movimento, influência essa que foi desprezada após o bloqueio da articulação rotacional da roda boba auxiliar.

## 5.6 Análise de custos e o ensino de robótica

Retomando a discussão sob a ótica educacional acerca do projeto, foi possível observar que trata-se de um projeto com grande potencial de contribuição para o ensino de robótica. Por se tratar de uma disciplina ministrada ao final do currículo do curso de ECA, projetos como esse auxiliam na sintetização dos conhecimentos adquiridos ao longo do curso estabelecendo uma conexão entre diferentes disciplinas. Conteúdos como programação, eletrônica, desenho técnico, instrumentação e sistemas embarcados foram todos integrados na concepção de um protótipo que atenda seu propósito, de permitir o contato com novos conteúdos na área de robótica e também no planejamento de caminho. Essas já se destacam como razões suficientes para ressaltar a importância de levantar esforços para garantir que todos os alunos tenham acesso às práticas didáticas da disciplina. Atualmente, *kits* didáticos para o ensino de robótica oscilam em torno da faixa de preço de R\$350,00 sem considerar o preço de transporte quando adquiridos em lojas virtuais. Entretanto, grande parte desses kits não atendem às necessidades da disciplina, fazendo-se necessário buscar por outras alternativas. A Tabela 7 traz o levantamento realizado no ano de 2017, quando o projeto se iniciou, além da atualização desses valores viabilizando uma análise de custos para o ensino de robótica.

Tabela 7: Custo de aquisição dos equipamentos com base em 2017 e 2019

<b>Componente</b>	<b>Custo de aquisição (2017)</b>	<b>Custo de aquisição (2019)</b>
Chassi 2 rodas	R\$ 69,90	R\$ 65,00
Kit Arduino	R\$ 65,80	R\$ 54,90
Sensor ultrassônico	R\$ 12,80	R\$ 12,69
Servomotor	R\$ 16,00	R\$ 14,90
Sensor de velocidade (2x)	R\$ 23,90	R\$ 22,00
Bateria de 9V e carregador	R\$ 40,00	R\$ 48,89
Driver ponte-H	R\$ 14,49	R\$ 15,30
<b>TOTAL</b>	<b>R\$ 242,89</b>	<b>R\$ 233,68</b>

Fonte: Mercado Livre.

Semestralmente, o curso de ECA da UFLA oferta 50 vagas para alunos ingressantes. Partindo da premissa de que o número de vagas nas disciplinas do curso atendam, pelo menos, o número de alunos ingressantes no curso e considerando o método avaliativo em grupos utilizado na disciplina, uma divisão simples em grupos de 5 alunos geraria demanda de 10 protótipos como o que foi utilizado nesse trabalho. Alinhando esse cenário, com o custo de aquisição mais atualizado da Tabela 7, semestralmente seriam investidos pelos alunos R\$2336,80 no ensino de robótica. Ao projetar tal análise para os alunos enquanto indivíduos, para cada aluno em um grupo nesse cenário seria necessário um investimento de R\$46,74.

Quando trazido para a análise individual, o custo de aquisição a princípio se torna menos impactante. Entretanto, esse pode se tornar um desafio para uma parte dos alunos do curso se seu contexto socioeconômico for levado em consideração. Conforme mostrado na seção 4.1.1, cerca de 24% dos alunos ingressantes no curso são alunos de baixa renda familiar. Muitos desses alunos contam com programas institucionais que permitem sua permanência na universidade contando com bolsas no valor de R\$300,00. Ao considerar a realidade desses alunos que dependem de bolsas, a sua contribuição para a aquisição dos equipamentos necessários à disciplina representa cerca de 16% de sua renda mensal. Esse se torna um indicador preocupante quando se considera que outras disciplinas também demandam investimentos semelhantes.

Essa análise de custos possuiu um caráter informativo, de entendimento do papel do aluno e a realidade encontrada no ensino nos cursos de graduação. A realidade dos alunos pode ser um fator limitante em sua trajetória dentro da universidade e, observando o caráter inclusivo da educação pública, o acesso às informações supracitadas permite elaborar medidas de inclusão e difusão de conhecimento sem a interferência de fatores sociais e econômicos.

## **6 CONCLUSÃO**

O presente trabalho utilizou o algoritmo *fuzzy* na resolução do problema de planejamento de caminho para um robô móvel. Inicialmente alguns testes e simulações foram necessários para avaliar a robustez e a capacidade do algoritmo gerar caminhos com percursos menores, exigindo menos esforço dinâmico por parte do robô. Foi observado que a variação de alguns parâmetros de configuração de projeto influenciam na resposta do algoritmo, chegando a determinar se o algoritmo conseguiria planejar o

caminho com sucesso. A natureza dos ambientes, alinhada com a correta configuração dos parâmetros possibilitou entender como o algoritmo toma decisões, com base em seu conjunto de regras *fuzzy*, para definir o trajeto. A configuração do algoritmo é a etapa que determina o tratamento que o algoritmo *fuzzy* terá ao tomar decisões, podendo ser mais cauteloso ou mais flexível em sua tomada de decisões. Isso foi constatado em cada um dos ambientes adotados e, em geral, em condições de maior cautela o algoritmo se mostrou capaz de planejar um caminho a ser executado. Sob essas condições, os caminhos encontrados pelo algoritmo apresentaram curvas mais suaves, e menos desvios de rota, que exigem menos esforço do robô, além de manter distâncias seguras dos obstáculos tanto em curvas quanto em seções de trajeto paralelo aos obstáculos. Já em condições mais flexíveis ou mais arriscadas, o sucesso do algoritmo dependia também da influência que tais configurações exerceriam na relação entre o robô e o ambiente. Uma característica dessa configuração foi a permissividade em relação aos desvios no trajeto. O algoritmo se tornou mais sensível aos obstáculos, desviando de um trajeto retilíneo com mais frequência, o que na maioria dos casos gerou um percurso maior.

Devido à indisponibilidade do ambiente de estudo, uma casa de vegetação, foi necessário construir um ambiente de trabalho inspirado nas dependências da estufa, porém em dimensões inferiores. Tais dimensões se apresentaram como um desafio ao projeto, uma vez que a área livre para o trajeto, composta por corredores ao redor do obstáculos, foi reduzida. Para dar mais flexibilidade ao algoritmo diante dessa limitação, parâmetros foram configurados para permitir que o algoritmo tivesse mais liberdade e tempo de reação diante dos obstáculos. Nesse caso, o caminho gerado não possuía curvas bruscas, entretanto as curvas foram caracterizadas por sua proximidade em relação aos vértices dos obstáculos. O caminho determinado pelo algoritmo passou por uma etapa de tratamento e redimensionamento antes de ser informado ao robô móvel, que foi programado para ser versátil e capaz de interpretar novos mapas sem grandes esforços. O desenvolvimento do código adaptável no Arduino foi fundamental para o sucesso do robô em sua locomoção, apesar de restringir o movimento a trechos retos e intervalos discretos de ângulos de curvas. Dessa forma, em futuras aplicações do código pode-se expandir a capacidade de avaliação das células vizinhas à posição do robô, resultando em mais flexibilidade na movimentação da plataforma robótica.

Os resultados obtidos demonstraram que o algoritmo adotado se apresenta como uma ferramenta simples e eficaz no planejamento de caminhos. Dessa forma, sua aplicação em casas de vegetação abre caminhos para implantação dessa tecnologia na

execução de tarefas como a ministração de agentes químicos, que podem ser nocivas à saúde. Alinhando também ao contexto de agricultura de precisão, robôs móveis guiados pelo algoritmo *fuzzy* também podem auxiliar o ser humano no trato de seções de uma estufa ou lavoura, sendo mais preciso em relação às necessidades locais do cultivo, reduzindo desperdícios e permitindo um tratamento mais direcionado. A etapa de aplicação do robô em um ambiente estático reduzido, proposta e concluída nesse trabalho, trouxe um alinhamento com a característica de seccionamento de lavouras da agricultura de precisão, mostrando seu potencial de contribuição para a sociedade.

Todas as conclusões trazidas até aqui, levantam a reflexão sobre a importância do ensino de robótica para a formação de profissionais que serão aptos a utilizar essa área de conhecimento para sanar dores da nossa sociedade, como os riscos à saúde humana, economia e redução de desperdícios. Por esses motivos, é de extrema importância garantir que mais alunos possam ter contato com práticas que permitam seu desenvolvimento intelectual e profissional para que eles sejam dotados das ferramentas e conhecimentos necessários. Sob tal perspectiva de formação profissional, este trabalho também se propõe a informar e refletir sobre a posição do aluno perante a estrutura de ensino, e como esta posição pode se manifestar como um fator limitante em sua formação. Diante das informações apresentadas sobre o perfil de alunos, este trabalho se propõe a munir a comunidade acadêmica com informações pertinentes à manutenção de uma educação inclusiva e uma formação acadêmica despidida de limitações socioeconômicas. Face à tal exposição, o presente estudo se coloca como um projeto embasado tecnicamente, planejado sob uma ótica de produção didática inclusiva e com a validação de seus princípios educacionais por meio da proposição e desenvolvimento de uma alternativa tecnológica que se coloca à disposição da sociedade na resolução de problemas, ressaltando a importância do ensino de robótica e o contato com novas tecnologias.

### **6.1 Perspectivas futuras**

O presente projeto deixa portas abertas para inúmeras aplicações e melhorias futuras. A integração de módulos de comunicação remota à plataforma tanto para o planejamento de caminho *online* quanto para georreferenciamento, abrem portas para aplicação deste trabalho em ambientes de dimensões muito superiores, como lavouras. Além disso, o uso de painéis solares na plataforma também surge no horizonte como uma alternativa à limitação energética do protótipo. Entretanto, como o presente projeto visa

a continuidade de seu desenvolvimento por parte de outros alunos no futuro, algumas boas práticas nesse trabalho foram adotadas para facilitar algumas das possibilidades já levantadas:

- a) O código em Matlab desenvolvido para o tratamento da trajetória gera um arquivo de texto formatado de acordo com a sintaxe da linguagem de programação C++, pois vislumbra atuar como um arquivo de leitura para comunicação remota. Assim todo o planejamento de caminho permanece *off-line*, porém sem a necessidade de uma interface humana para atualizar o código no Arduino, inserindo ao projeto conceitos como IOT.
- b) Conceitos de visão computacional podem ser integrados ao projeto, permitindo que o planejamento do caminho passe a ser *online*;
- c) Incorporação de um segundo nível à plataforma, permitindo acoplar ao robô um braço robótico para realizar manipulações em plantas ou outras estruturas responsáveis pela ministração de agentes químicos;

## REFERÊNCIAS

ACACCIA, G. M. et al. Mobile robots in greenhouse cultivation: inspection and treatment of plants. In: **Proc. of ASER 2003, 1st International Workshop on Advances in Service Robotics**. Bardolino, 2003.

ADÔRNO, B. V.; BORGES, G. A. Um método de planejamento de trajetória para robôs móveis através de passeios aleatórios adaptativos e mapa de rotas. In: AUTOMÁTICA, C. B. D. **Anais do Congresso Brasileiro de Automática**. Brasília, 2006.

ALVES, R. J. F. **Rastreamento de agrobots em estufas agrícolas usando modelos ocultos de Markov**: Comparação do desempenho e da correção dos algoritmos de Viterbi e Viterbi com janela de observações deslizante. Ponta Grossa, 2015. 80 p. Dissertação (Mestrado em Computação Aplicada) – Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, 2015.

ANDURKAR, A. G.; TANKAR, R. Fuzzy Logic Based Path Navigation for Robot using Matlab. **International Research Journal of Engineering and Technology (IRJET)**, v. 4, Junho 2017.

ARTUZO, F. D.; JANDREY, W. F. Possibilidade de Redução de Impacto Ambiental Através do Uso de Fertilizantes na Agricultura de Precisão. In: **4º Congresso Internacional de Tecnologias para o Meio Ambiente**. Bento Gonçalves, 2014.

BENTO, F. A. P. **Determinação de rotas otimizadas para robótica móvel em uma casa de vegetação**. Lavras, 2019. Monografia (Bacharel em Engenharia de Controle e Automação) – Universidade Federal de Lavras, Lavras, MG, 2019.

BOUMEHRAZ, M.; BENMAHAMMED, K. **Tuning Fuzzy Inference Systems by Q-Learning**. 2014.

BUEMI, F. et al. The Vision System For The Agrobot Project. In: **Proc. 5th ASAE International Conference on Computers in Agriculture**. 1994.

CARDOSO, B. . M. . C. **Desenvolvimento de um robô móvel com trajetória programável utilizando a plataforma arduino**. Ponta Grossa, 2016. Monografia (Bacharel em Ciência da Computação) – Universidade Tecnológica Federal do Paraná, PR, 2016.

CHALWA, V. N.; GUNDAGI, S. S. Mechatronics based remote controlled agricultural robot. **International Journal of Emerging Trends in Engineering Research**, v. 2, p. 33-43, Julho 2014.

CRAIG, J. J. **Introduction to Robotics: Mechanics and control**. 3ª. ed. Upper Saddle River: Pearson, 2004. 408 p.

DINO. Os impactos da tecnologia no campo, com a agricultura 4.0. **EXAME**, 2018. Disponível em: <<https://exame.abril.com.br/negocios/dino/os-impactos-da-tecnologia-no-campo-com-a-agricultura-40/>>. Acesso em: 23 jun. 2019.

DUCHONĚ, F. et al. Path planning with modified A star algorithm for a mobile robot. **Procedia Engineering**, v. 96, p. 59-69, 2014.

EDUCBA. Raspberry Pi 3 vs Arduino. **EDUCBA**, 2018. Disponível em: <<https://www.educba.com/raspberry-pi-3-vs-arduino/>>. Acesso em: 11 Setembro 2019.

FILHO, R. S.; CUNHA, J. P. A. R. Agricultura de precisão: particularidades de sua adoção no sudoeste de Goiás – Brasil. **Engenharia Agrícola**, Jaboticabal, v. 35, n. 4, p. 689-698, Jul./Ago. 2015.

GASPARETTO, A. et al. Path Planning and Trajectory Planning Algorithms: A General Overview. In: G., C.; F., G.-B. **Motion and Operation Planning of Robotic Systems: Background and Practical Approaches**. Springer, v. 29, 2015. Cap. 1, p. 3-27.

GURUJI, A. K.; AGARWAL, H.; PARSEDIYA, D. K. Time-Efficient A\* Algorithm for Robot Path Planning. **Procedia Technology**, v. 23, p. 144-149, 2016.

HACKENHAAR, N. M.; HACKENHAAR, C.; ABREU, Y. V. Robótica na agricultura. **Interações**, Campo Grande, v. 16, p. 119-129, jan./jun. 2015.

JÚNIOR, A. M. M. **O uso do algoritmo mapa de rotas probabilístico na robótica móvel**: Planejamento de caminhos para ambientes estáticos com foco em casas de vegetação. Lavras, 2018. 53 p. Monografia (Bacharel em Engenharia de Controle e Automação) – Universidade Federal de Lavras, MG, 2018.

KALA, R. Code for Robot Path Planning using Fuzzy Logic, Indian Institute of Information Technology, Allahabad. **Rahul Kala**, Junho 2014. Disponível em: <<http://rkala.in/codes.php>>. Acesso em: 21 Junho 2019.

KARVAKI, L. et al. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. **IEEE Transactions on Robotics and Automation**, v. 12, p. 566-580, Setembro 1996.

LAMINI, C.; BENHLIMA, S.; ELBEKRI, A. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning. **Procedia Computer Science**, v. 127, p. 180-189, 2018.

LEÇA, L. M. S. F. **Implementação de Algoritmos de Planejamento de Trajetórias de Robôs Móveis**. Porto, 2015. 67 p. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) -Faculdade de Engenharia da Universidade do Porto, 1998.

LEITE, D. S. **Controle autônomo de robô móvel baseado em lógica fuzzy**. Rio de Janeiro, 2013. Monografia (Bacharel em Engenharia de Controle e Automação) – Pontifícia Universidade Católica, RJ, 2013.

MANAGEMENT, F. O impacto da agricultura de precisão. **Farm Management**, 2018. Disponível em: <<https://www.farmmanagement.pro/pt/o-impacto-da-agricultura-de-precisao/>>. Acesso em: 23 jun. 2019.

MIRANDA, A. C. C.; VERÍSSIMO, A. M.; CEOLIN, A. C. Agricultura De Precisão: Um Mapeamento Da Base Da Scielo. **Revista Gestão.Org**, v. 15, n. Especial, p. 129-137, Dezembro 2017.

NGUYEN, V. et al. Analysis of main tasks of precision farming solved with the use of robotic means. In: **12th International Scientific-Technical Conference on Electromechanics and Robotics "Zavalishin's Readings" - 2017**. São Petresburgo: EDP Sciences, v. 113, 2017.]

NIKU, S. B. **Introdução à robótica**: análise, controle, aplicações. 2ª. ed. Rio de Janeiro: LTC, 2013. 402 p.

PAIVA, L. S. **Análise e Geração de Caminho Mínimo Aplicado à Navegação de Veículos Terrestres**. Lavras, 2017. 141 p. Tese (Doutorado em Engenharia Agrícola) – Universidade Federal de Lavras.

PANDEY, A.; PARHI, D. R. MATLAB Simulation for Mobile Robot Navigation with Hurdles in Cluttered Environment Using Minimum Rule Based Fuzzy Logic Controller. In: ENGINEERING, 2. I. C. O. I. I. A. A. M. **Procedia Technology**, v. 14, 2014. p. 28-34.

SPONG, M. W.; VIDYASAGAR, M. **Robot dynamics and control**. 1ª. ed. John Wiley and Sons, 1989.

SWINTON, S. M.; DEBOER, J. L. Global adoption of precision agriculture technologies: who, when and why? **Proceedings of the 3rd European Conference on Precision Agriculture**, 2001. 557-562.

TANGERINO, G. T. et al. Controle de esterçamento de robô agrícola móvel de quatro rodas guiáveis. **X SBAI – Simpósio Brasileiro de Automação Inteligente**, São João del Rei, v. 10, p. 797-802, Setembro 2011.

TANKE, N. F. et al. **Automation of Hydroponic Installations using a**. Proceedings of the International Conference of Agricultural Engineering CIGR-Ageng 2012. Valencia: CIGR-EurAgEng. 2012. p. 6.

VENTURA, J. M. V. **Estudo de coordenação de robôs móveis com obstáculos**. São Carlos. 2011. Dissertação (Mestrado em Ciências) – Universidade de São Paulo, 2011.

VIEIRA, F. C. **Controle dinâmico de robôs móveis com acionamento diferencial**. Natal, 2005. 106 p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Norte, 2005.

WHIPKER, L. D.; AKRIDGE, A. T. **2009 Precision Agricultural Services Dealership Survey Results**, 2009. Working Papers 56111, Purdue University, Department of Agricultural Economics.

WOLF, D. F. et al. Robótica Móvel Inteligente: Da Simulação às Aplicações no Mundo Real. In: **Atualizações em informática 2009**. Rio de Janeiro: PUC-Rio, 2009. Cap. 1.