



**FILIPPE ALVES PEREIRA BENTO**

**DETERMINAÇÃO DE ROTAS OTIMIZADAS PARA  
ROBÓTICA MÓVEL EM UMA CASA DE VEGETAÇÃO**

**LAVRAS-MG**

**2019**

**FILIPPE ALVES PEREIRA BENTO**

**DETERMINAÇÃO DE ROTAS OTIMIZADAS PARA ROBÓTICA MÓVEL EM  
UMA CASA 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**

**2018**

*Dedicatória*

*Este trabalho é dedicado aos meus pais, Sandra e Joaquim, por todo amor, suporte e carinho conferidos, incondicionalmente, a mim. Também dedico a todos os trabalhadores do mundo que passaram a vida subjugados às vontades do Capital...*

## **AGRADECIMENTOS**

Agradeço, primeiramente, aos meus pais pela oportunidade de uma vida com muito carinho, amor, aprendizado e alegrias. Agradeço também a minha namorada, Graziella, por todo amor e apoio. Ao professor Leonardo Paiva pelos conhecimentos compartilhados comigo, pelo apoio para resolução de problemas e pela paciência para me guiar pelo caminho maravilhoso da Robótica.

**MUITO OBRIGADO!**

## EPÍGRAFE

*“Um dos paradoxos dolorosos do nosso tempo reside no fato de serem os estúpidos os que têm a certeza, enquanto os que possuem imaginação e inteligência se debatem em dúvidas e indecisões. ”*

**Bertrand Russell**

*“O homem como cientista é amoral. Só é moral como homem, quando se preocupa com o que descobre, se será usado para o bem ou para o mal. Como toda descoberta científica dá mais poderes sobre a natureza, ela pode aumentar o bem ou o mal. “*

**César Lattes**

*"Saber muito não lhe torna inteligente. A inteligência se traduz na forma que você recolhe, julga, maneja e, sobretudo, onde e como aplica esta informação."*

**Carl Sagan**

## RESUMO

O mundo atual possui uma demanda crescente pela produção de alimentos. Com níveis altos de crescimento demográfico, limitação de recursos naturais, novas pragas e doenças que atacam cultivos, novas tecnologias estão sendo desenvolvidas para a otimização da produção agrícola. Nesse contexto, o presente trabalho busca propor uma metodologia para determinar um caminho otimizado dentro de uma casa de vegetação para a o deslocamento de uma plataforma robótica. A planta do ambiente utilizado representa uma estufa do Departamento de Recursos Hídricos da Universidade Federal de Lavras – UFLA. O algoritmo para determinar o caminho é o Mapas de Rotas Probabilístico (MRP), o qual traça o caminho mais curto entre dois pontos informados. A plataforma robótica utilizada foi construída a partir do microcontrolador Arduino. O processo consiste em obter uma foto do ambiente, aplicar o MRP e gerar o caminho. Com posse do caminho gerado, o robô foi programado para executar as manobras de movimentação na planta seguindo o mesmo. Assim, foi realizada, com resultados satisfatórios, a aplicação de um robô móvel em um ambiente agrícola, otimizando os deslocamentos dentro da casa de vegetação, construída em escala menor, visando a economia de tempo e recursos. O presente trabalho teve também o objetivo de instituir um padrão de metodologia prática para ser usado em trabalhos futuros.

**Palavras-chave:** Robótica Móvel. MRP. Casas de Vegetação.

## ABSTRACT

The world today has a growing demand for food production. With high levels of population growth, limited natural resources, new pests and diseases that attack crops, new technologies are being developed to optimize agricultural production. In this context, the present work seeks to propose a methodology to determine an optimized path inside a greenhouse for the displacement of a robotic platform. The plant of the environment used represents a greenhouse of the Departamento de Recursos Hídricos da Universidade Federal de Lavras – UFLA. The algorithm to determine the path is Probabilistic Roadmap (PRM) planner, which traces the shortest path between two points reported. The robotic platform used was built using an Arduino microcontroller. The process is to get a picture of the environment, apply the PRM, and generate the path. From this, the robot was programmed to execute the maneuvers of movement in the plant following the generated path. Therefore, the application of a mobile robot in an agricultural environment was done successfully, optimizing the displacements inside the greenhouse, built on a smaller scale, aiming to save time and resources. The present work also had the objective of establishing a standard of practical methodology to be used in future works.

**Keywords:** Mobile Robotics. PRM. Greenhouses.

## LISTA DE FIGURAS

Figura 1 – Robô Unimate no programa The Tonight Show.....	16
Figura 2 - Robô Shackey realizando manobras .....	16
Figura 3 – Linha de robôs apresentada pela Honda em 2018.....	17
Figura 4 – Robô Mark II Talon da marinha americana .....	18
Figura 5 – Exemplo de um robô empregado na colheita de morangos .....	20
Figura 6 – Robô Aurora, Málaga, Espanha .....	21
Figura 7 – Robô móvel para colheita de tomates. ....	21
Figura 8 – Exemplo de caminho gerado pelo MRP .....	26
Figura 9 - Simulação 3D da casa de vegetação do Departamento de Fitopatologia da UFLA . .....	27
Figura 10 - Aplicação do MRP no mapa da casa de vegetação do Departamento de Fitopatologia da Ufla. a) Mapa de rotas gerado; b) Caminho a ser percorrido pelo robô.....	28
Figura 11 - Caminho percorrido (em azul) pelo robô .....	28
Figura 12 - Grades de Ocupação Binária. a) Grade de ocupação binária da casa de vegetação; b) Grade de ocupação binária da casa de vegetação com obstáculos inflados.....	30
Figura 13 - Mapa de Rotas Probabilístico para a casa de vegetação do Departamento de Fitopatologia da Ufla.....	30
Figura 14 - Exemplo de declaração dos parâmetros de posição inicial, final, número de nós e imagem. ....	31
Figura 15 - Fases de “bombardeamento” e aprendizado, respectivamente, do caso 1.....	33
Figura 16 - Fases de “bombardeamento” e aprendizado, respectivamente, do caso 2. ....	33
Figura 17 - Caminho final – a) Caso 1 b) Caso2.....	34
Figura 18 - Planta com obstáculos, representados por traços que impedem o trajeto.....	36
Figura 19 - Fases de “bombardeamento” e aprendizado do Caso 1, respectivamente.....	36
Figura 20 - Fases de “bombardeamento” e aprendizado do Caso 2, respectivamente . ....	37
Figura 21 - Caminho final a) Caso 1 b) Caso 2.....	28
Figura 22 - Planta com todos os caminhos bloqueados por obstáculos.....	39
Figura 23 - Aplicação do algoritmo.....	40
Figura 24 - Mensagem de erro "Caminho não encontrado" .....	41
Figura 25 - Robô utilizado no projeto.....	41
Figura 26 - Exemplo de uma placa Arduino Uno R3, como a usada no robô.....	42

Figura 27 - Exemplificação do funcionamento de um circuito Ponte H.....	43
Figura 28 - Sentidos da corrente e do giro do motor variam conforme as chaves são acionadas.....	43
Figura 29 - Circuito similar ao da placa, com transistores e diodos realizando o chaveamento.....	44
Figura 30 - Motor DC 3-6V.....	45
Figura 31 - Ligação dos motores na placa e código na IDE do Arduino.....	46
Figura 32 - Exemplos da variação do Duty Cycle e ondas produzidas.....	47
Figura 33 - Velocidade sendo passada como parâmetro, em destaque, no código.....	47
Figura 34 - Exemplo de movimentação para o robô.....	48
Figura 35 - Planta em escala menor da estufa.....	49
Figura 36 - Pontos de partida e chegada do robô na planta contruída.....	50
Figura 37 - Fase de bombardeamento do Teste 1.....	51
Figura 38 - Fase de aprendizado do Teste 1.....	52
Figura 39 - Caminho gerado.....	52
Figura 40 - Relações de triângulo retângulo para encontrar a extensão dos segmentos de reta do caminho.....	53
Figura 41 - Código implementado no Arduino para o deslocamento do robô.....	54
Figura 42 - Fase de bombardeamento do Teste 2.....	54
Figura 43 - Fase de aprendizado do Teste 2.....	55
Figura 44 - Caminho gerado (vermelho), subdivisão dos segmentos (a até f) e exemplo da aplicação do Teorema de Pitágoras no segmento c.....	56
Figura 45 - Código de movimentação do robô para o Teste 2.....	56
Figura 46 - Código do Teste 1 com correções.....	58
Figura 47 - No Teste 1 o robô invadiu alguns limites dos obstáculos .....	60

## **LISTA DE TABELAS**

Tabela 1 – Parâmetros e resultados para aplicação do MRP no cenário livre de obstáculos. ...	32
Tabela 2 – Parâmetros e resultados da aplicação do MRP no cenário com obstáculos geométricos.....	35

## LISTA DE SIGLAS

A*	A-Estrela
AGs	Algoritmos Genéticos
MAPRM	Medial Axis Probabilistic Roadmap
MRP	Mapas de Rotas Probabilística
RRT	Rapidly Exploring Random Tree
UFLA	Universidade Federal de Lavras
MIT	Massachusetts Institute of Technology
PWM	Pulse Width Modulation

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>12</b>
<b>2 OBJETIVOS .....</b>	<b>14</b>
2.1 Objetivo geral.....	14
2.2 Objetivos específicos.....	14
<b>3 REFERENCIAL TEÓRICO .....</b>	<b>15</b>
3.1 Robótica, inteligência artificial e robôs móveis.....	15
3.1.1 Aplicações da robótica móvel no contexto de casas de vegetação .....	18
3.2 Planejamento de caminhos e trajetórias.....	23
3.3 Algoritmos usados para geração de caminhos e o MRP .....	25
<b>4. MATERIAIS E MÉTODOS .....</b>	<b>29</b>
4.1 Descrição dos ambientes .....	30
4.2 O algoritmo MRP .....	31
4.1.2 Testes de validação do MRP .....	32
4.1.2.a) Teste 1.....	32
4.1.2.b) Teste 2 .....	35
4.1.2.c) Teste 3 .....	39
4.3 Plataforma móvel.....	41
4.3.1 Descrição dos componentes .....	42
4.4 Código de programação do robô .....	46
4.5 Testes com o robô .....	49
4.5.1 – Pista de testes .....	49
<b>5. RESULTADOS E DISCUSSÕES .....</b>	<b>57</b>
<b>6 CONCLUSÃO.....</b>	<b>61</b>
6.1 Perspectivas futuras .....	62
6.2 Perspectivas éticas .....	62
<b>7 REFERÊNCIAS .....</b>	<b>65</b>

## 1 INTRODUÇÃO

Nos últimos 60 anos o mundo científico sofreu grandes mudanças com o advento da robótica e inteligência artificial (RUSSEL, 2001). Esses campos de estudo trouxeram a possibilidade de criar e programar máquinas que participassem e ajudassem os humanos nas mais diversas atividades, fossem relacionadas à pesquisa e desenvolvimento científico, indústria, medicina e, até mesmo, como visto nas últimas décadas, alcançando o cotidiano das pessoas em suas tarefas diárias (NIKU, 2013). Motivados pela grande economia de recursos, alta produtividade, qualidade, diminuição de acidentes e riscos e, conseqüentemente, otimização dos lucros que o emprego de robôs trouxe para os processos de produção industrial, esse setor vem investindo progressivamente na inserção dessas máquinas em suas linhas. No Japão, por exemplo, de cada dez trabalhadores do setor, um é robô. Sendo uma especialização da robótica, a robótica móvel também é significativamente presente nas atividades atuais, tendo sido extensivamente estudada e desenvolvida em pesquisas e trabalhos científicos nas últimas décadas, ganhando destaque nas novas tecnologias e demonstrando grande potencial para inovação (WOLF et al., 2009). Os robôs móveis possuem vantagens sobre os robôs que trabalham fixados em células por terem menos partes móveis e articulações e podendo ser empregados em atividades em vários locais diferentes com amplo alcance geográfico, fazendo uso de pernas, esteiras ou rodas (PIO; CASTRO; CASTRO JÚNIOR, 2006; SECCHI, 2008). Essas características, aliadas à sua resistência a ambientes hostis, grande precisão e exatidão de movimentos permitem o emprego para várias finalidades como desarmar bombas, explorar ambientes perigosos e difíceis para locomoção humana, transporte, limpeza de grandes e pequenas áreas e aplicações na agricultura (SECCHI, 2008; VICTORINO; RIVES; SLAM, 2006; WOLF et al., 2009). Pesquisas voltadas para as áreas de mecanização e automação do campo vem abordando a utilização de robôs móveis para colheita, irrigação, pulverização e plantio, a fim de otimizar a produção e melhorar a sua qualidade (ABRAHÃO; MEGDA; BECKER, 2010). Plataformas móveis podem, inclusive, atuar no controle de pragas em casas de vegetação (MANDOW et al., 1996).

A principal função de um robô móvel é realizar sua locomoção de um ponto especificado até outro. O planejamento de sua trajetória é crucial para evitar colisões em possíveis obstáculos (ADÔRNO; BORGES, 2006). Os métodos existentes para planejar uma trajetória são classificados como probabilísticos ou determinísticos (PAIVA, 2017). Os probabilísticos usam

amostras do ambiente para representa-lo com grande aproximação, tornando o custo com recursos computacionais menor e podendo ser aplicado em lugares de grande dimensão (MARTINS et al., 2003). Já os determinísticos requerem que todas as configurações possíveis do robô sejam conhecidas, aumentando o consumo de gasto computacional e inviabilizando a sua aplicação em áreas extensas (LATOMBE, 1991; SIGWART; NOURBAKHS, 2004; SILVEIRA, 2016). O algoritmo Mapas de Rotas Probabilístico possui baixo custo computacional e obteve resultados significativos quando aplicado em ambientes estáticos (KAVRAKI et al., 1996). Exemplos de suas aplicações incluem realização de solda ponto a ponto na indústria, limpeza de ductos de usinas nucleares e soldas em fuselagem de aviões. O planejamento da trajetória visa fornecer a rota otimizada para que o robô gaste o menor tempo possível no seu trajeto, evitando desgastes desnecessários e economizando energia e recursos. Assim, o estudo desse campo se torna muito relevante para o setor industrial, onde o tempo está diretamente relacionado à produtividade e ao lucro, como também o aumento da qualidade de condições de trabalho.

## 2 OBJETIVOS

Motivado pelo contexto exposto acima, o presente trabalho pretende:

### 2.1 Objetivo geral

Realizar a locomoção de uma plataforma robótica móvel de um ponto determinado a outro, em um ambiente representando uma casa de vegetação, planejando o caminho com o algoritmo MRP.

### 2.2 Objetivos específicos

Os objetivos específicos foram separados de acordo com a ordem de execução das tarefas:

- a) Fazer uma revisão bibliográfica sobre Robótica, Robótica Móvel e suas aplicações no contexto de casas de vegetação.
- b) Obter uma imagem da planta e discretizá-la.
- c) Aplicar o algoritmo MRP, utilizando a *toolbox* de sistemas robóticos do Matlab, obtendo o grafo de conexões do ambiente.
- d) Obter o caminho de menor custo
- e) Configurar o robô de acordo com os parâmetros para execução do caminho.
- f) Realizar as manobras para que o robô alcance o ponto desejado.

### 3 REFERENCIAL TEÓRICO

#### 3.1 Robótica, inteligência artificial e robôs móveis

Nos últimos 60 anos o mundo científico sofreu grandes mudanças com o advento da robótica e Inteligência Artificial (IA) (RUSSEL, 2013). A IA tem sido apontada atualmente por cientistas de todo mundo (RUSSEL 2013), de diferentes ramos do conhecimento, como o campo de pesquisa que mais gostariam de atuar. Paralelamente ao desenvolvimento da IA, a robótica também evolui exponencialmente. Desde a publicação da história fictícia *Rossum's Universal Robots*, em 1922, pelo autor tcheco Carel Kapek, a palavra “robô” habita o imaginário das pessoas em todo o mundo (CAPEK, 2004, HOCKSTEIN et al, 2007). Mas foram necessários anos de estudos e pesquisas, principalmente em matemática, física e eletrônica, para que somente em 1946 ambas ciências fossem de fato inicializadas concretamente através da construção do primeiro computador digital eletrônico *Electronic Numerical Integrator and Computer – ENIAC*, pelos cientistas Eckert e Mauchley (BRITO, 2008; BATIK, 2013), e do desenvolvimento do controlador magnético por George Devol, um dos maiores cientistas da robótica (DEVOL,1952; NIKU,2013). Em 1952 a primeira máquina de comando numérico foi construída no Instituto de Tecnologia de Massachusetts. Em 1954, Devol cria o primeiro robô programável, abrindo portas para o seu uso funcional na indústria (BEKEY, 1998). Em 1955 os cientistas Jacques Denavit e Richard S. Hartenberg desenvolveram as matrizes de transformação homogênea, ferramenta crucial para os robôs manipuladores e móveis (HARTENGERG et al, 1955; NIKU,2013). Os primeiros robôs industriais apareceram em 1961, com a fundação da Unimation, a primeira empresa de robótica fundada por Devol e Joseph Endelberger, em Danbury nos Estados Unidos (EUA) (MUNSON, 2010). A General Motors (GM), no mesmo ano, instalou o primeiro robô em sua produção, proveniente da Unimation. Já em 1966, como mostra a Figura 1, Endelberger apresentou o robô Unimate para uma grande audiência em rede nacional americana, no popular programa televisivo *The Tonight Show*. Durante a apresentação, o robô realizou números servindo cerveja em um copo, fazendo uma tacada de golfe e até mesmo regendo banda musical do programa. Este fato foi uma grande exposição e publicidade não só para a empresa Unimation, mas para toda a robótica, que passou de mera ficção para realidade.

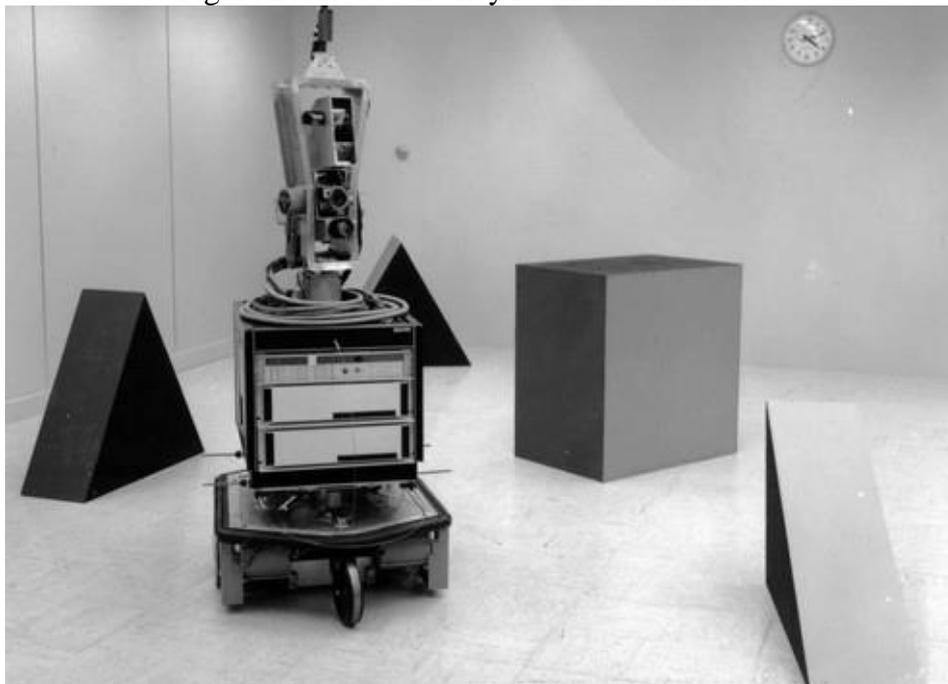
Figura 1 – Robô Unimate no programa *The Tonight Show*.



Fonte: Robotics Organization – A Tribute to Joseph Engelberger

Em 1967 ocorre a primeira importação de um robô na história, feita pelo Japão, ao comprar o Mark II, também da Unimation, para ser usado na pintura por pulverização da indústria automobilística. O robô Shakey, o primeiro inteligente da história, foi construído pelo *Stanford Research Institute* no ano de 1968 (NILSSON, 1984). Mesclando trabalhos de robótica e IA, ele foi projetado com a capacidade de reconhecer e modelar o ambiente ao seu redor, planejando caminhos e movimentos anticolisão. Na Figura 2 é mostrado o robô Shakey durante a realização de alguns movimentos.

Figura 2 – Robô Shakey realizando manobras



Fonte: SRI International

Nas décadas de 70 e 80, os avanços na robótica foram numerosos e se espalharam para o mundo (NIKU, 2013). O robô PUMA e o H0, primeiro humanoide, foram criados nesses anos. EUA e Japão se destacaram como os maiores desenvolvedores de tecnologias no campo, detentores de várias indústrias como a International Business Machines (IBM), Cincinnati Milacron, Honda, Fanuc, Unimation, nova divisão de robótica da GM dentre muitas outras. O mundo corporativo, então, absorveu em larga escala a IA e a robótica, as quais, muitas vezes, se tornaram indissociáveis uma da outra em sua aplicação, como o caso do já mencionado robô Shackey, o primeiro robô móvel que utilizava técnicas de IA aliadas a sensores (DE SOUSA PISSARDINI, 2013). A expansão dessas tecnologias evoluiu exponencialmente nas décadas posteriores à medida que processadores mais poderosos foram produzidos, possibilitando o surgimento dos carros autônomos e consolidando o desenvolvimento da robótica móvel (MAHADEVAN; NARENDRAN, 1990). Na Figura 3 é mostrado a mais recente linha de robôs móveis da Honda.

Figura 3 – Linha de robôs móveis apresentada pela Honda em 2018.



Fonte: Honda

Como vantagens dos robôs móveis, podemos citar o aumento do grau de automação e flexibilidade de um sistema, otimizando e integrando o seu controle, através de uma rede de computadores, otimização do fluxo de materiais e aumento da produtividade e diminuição do risco do processo por substituir a presença humana em ambientes nocivos (ROCHA, 2000). As desvantagens estão relacionadas ao uso de baterias, sua durabilidade e complexidade envolvida

para o desenvolvimento de tecnologias automáticas voltadas para a navegação desses robôs nas vias.

Atualmente, os robôs móveis desempenham funções em setores variados pelo mundo, indo desde aplicações menores no uso doméstico e comercial, até em atividades mais complexas e maiores relacionadas à indústria, agricultura e área militar (KAVRAKI; KOLOUNTZAKIS; LATOMBE, 1998; SECCHI, 2008; VICTORINO; RIVES; SLAM, 2006; WOLF et al., 2009). No setor industrial e comercial, são utilizados para transporte de cargas e materiais em depósitos, armazéns, entre postos de trabalhos e dentro de linhas de produção (ROCHA, 1998). O setor militar foi responsável por um grande desenvolvimento deste campo da robótica, tendo em vista a dinâmica e variedade das tarefas destinadas aos robôs móveis, tais como vigilância de instalações, desarme de bombas, transporte de materiais, ataques com armas de fogo, espionagem e reconhecimento de ambientes hostis (VOTH, 2004). O robô militar *Mark II Talon*, mostrado na Figura 4, é um exemplo desse tipo de máquina com essas aplicações.

Figura 4 – Robô militar *Mark II Talon* da marinha americana.



Fontes: FoxNews

### 3.1.1 Aplicações da robótica móvel no contexto de casas de vegetação

Nas últimas décadas, a demanda global por alimentos aumentou constantemente devido à explosão demográfica mundial, ao incremento da renda per capita na maioria dos países e à diversificação das dietas. Mas a partir de 2004, esse cenário resultou na escassez e encarecimento de alimentos, esgotamento de estoques, subida de preços de fertilizantes e, conseqüentemente, diminuição da produção mundial de alimentos (ONU, 2008). Uma grande

crise internacional se instalou, afetando, principalmente, o mercado de grãos e cereais. Sob esse contexto, o desenvolvimento de tecnologias voltadas à produção e otimização dos processos agrícolas se faz muito importante. Uma das alternativas para o aumento da produtividade, é o uso de estufas. O cultivo realizado sob essas condições, propicia o desenvolvimento da cultura em regiões pouco favoráveis, com irrigação controlada, reduzindo as contaminações por pragas, doenças e predadores naturais (GASPARETO, 2013; MANDOW et al., 1996; LENTEREN; WOETS, 1988). Outras vantagens que o cultivo em estufas permite é o controle de umidade (solo e ar) e temperatura (VIDA et al., 2004). Segundo Yang et al. (2013), essas duas variáveis, juntamente com a luminosidade, são os principais fatores que devem ser controlados em uma casa de vegetação. Segundo Gasparetto (2013), a sua construção segue formas geométricas bem definidas, sendo as retangulares com cobertura de plástico de polietileno as mais utilizadas. A organização interna depende de fatores ligados aos tipos de culturas e instalações, mas geralmente ficam organizadas em linhas, deixando corredores estreitos para locomoção dos trabalhadores que atuam no seu manejo (MANDOW et al., 1996).

No contexto mundial, o cultivo em estufas já é consolidado, sendo uma atividade praticada em todo o mundo e, em alguns países, responsável por maior parte da produção agrícola (SANCHEZ-HERMOSILLA et al., 2010). É amplamente utilizado por mercados, mesclando alta tecnologia e produtos com altos valores aditivos, contribuindo para o desenvolvimento da plantação através do controle de qualidade e resistência ao clima, insetos e doenças (YANG et al., 2013). Embora esse sistema seja susceptível de auto grau de automação e controle, a maior parte das operações ainda são manuais, como por exemplo colheita, pulverização e plantio (GASPARETO, 2013). Concomitante com a situação de ambiente fechado, pouca ventilação, espaço reduzido, uso de agentes químicos e biológicos, o trabalho no interior das estufas pode se tornar nocivo para a saúde dos trabalhadores (ACACCIA et al., 2003; GONZALEZ et al., 2009; MANDOW et al., 1996; MEHTA; BURKS; DIXON; 2008; SAMMONS; FURUKAWA; BULGIN, 2005).

Um estudo conduzido em uma província espanhola com mais de 20 mil hectares ocupados por casas de vegetação, avaliou a saúde de uma população de trabalhadores de suas instalações. Foram encontrados níveis altos de doenças de pele, conjuntivite e mucosas alteradas (MANDOW, 1996). Embora a automação de operações como a de pulverização, irrigação, colheita e polinização impactem positivamente nos custos econômicos do processo, ela ainda não elimina a exposição às substâncias nocivas. A robótica móvel pode ser alternativa para essa questão, uma vez que reduz significativamente o contato humano no ambiente de trabalho ao mesmo tempo em que aumenta a produtividade.

Em todo mundo, o setor agrícola vem investindo muitos recursos em robótica móvel para aplicações na agricultura (ABRAHÃO; MEGDA; BECKER, 2010). Agricultores vêm inserindo robôs inteligentes nas lavouras a fim de diminuir trabalho manual e aumentar a sua produtividade e qualidade (KO et al., 2015; SUPREM et al., 2013). Funções como o controle de pragas e doenças, irrigação e monitoramento em tempo real são exemplos de atividades importantes que um robô móvel pode desempenhar numa plantação (ALVES, 2015; MANDOW et al., 1996). Wolf et al. (2009) expõe que devido a maior precisão dos robôs em relação ao trabalho manual, a aplicação de pesticidas é feita em menor quantidade, trazendo menos resultados nocivos para o meio ambiente. Existem vários exemplos de aplicação de robótica móvel na agricultura. Qingchun et al. (2012) projetaram um robô inteligente capaz de colher morangos em casas de vegetação. Um braço robótico foi desenvolvido e configurado para realizar o corte propício para a colheita do fruto, a qual foi realizada com êxito. Testado dentro do ambiente de uma casa de vegetação com 100 morangos maduros, o robô alcançou o rendimento 86%. O sistema de navegação é composto por câmera de visão 3D, com resolução de 1024 x 768 pixels e lentes de 6mm para localização dos frutos. Um algoritmo de processamento digital de imagem planeja o corte do morango identificando apenas as áreas onde eles se encontram e, posteriormente, identificando o seu centro e realizando a manobra. Na Figura 5, é mostrado um robô empregado na colheita de morangos.

Figura 5 – Exemplo de um robô empregado na colheita de morangos



Fonte: Fresh Fruit Portal

Outra aplicação no mesmo contexto, descrita por Mandow et al. (1996), é o robô Aurora, representado na Figura 6, que foi desenvolvido por pesquisadores da Universidade de Málaga na Espanha. Este robô pode se deslocar de maneira autônoma pelos corredores de uma casa de vegetação, fazendo uso de sensores de baixo custo. Além disso, pode pulverizar agentes químicos na plantação pois possui um pulverizador acoplado em sua estrutura.

Figura 6 - Robô Aurora – Universidade de Málaga, Espanha



Fonte: Universidade de Málaga

O trabalho de Nezhad et al. (2011) desenvolveu um protótipo de robô para ser empregado na colheita de tomates. A Figura 7 mostra esse protótipo. Ele é formado por um braço robótico com uma mão mecânica na extremidade para realizar a colheita, e sistema de navegação para navegação pela casa de vegetação. Seus sensores detectam não só a posição de frutos vermelhos, mas também a quantidade, através do processamento digital de imagens. Assim, a mão mecânica, com o auxílio de um sensor ultrassônico que calcula a distância, se move em direção ao fruto, o engloba em seu interior e realiza torção para desprendê-lo da planta. Os resultados obtidos foram satisfatórios, ressaltando como vantagens desse sistema o fato de poder evitar perdas na colheita, sendo capaz de cobrir áreas mais vastas em menores períodos de tempo e com maior velocidade se comparado aos humanos

Figura 7 – Robô móvel para colheita de tomates



Fonte: Nezhad (2011)

Bac et. al (2014) desenvolveram um robô móvel para colheita de pimenta doce. Seu sistema é semelhante aos outros descritos anteriormente, com plataforma de navegação, braço robótico e sensores para os frutos e obstáculos. O sensoriamento do fruto é realizado, primeiramente, obtendo uma imagem do ambiente através das duas câmeras acopladas na estrutura. Essas imagens são processadas em conjunto com dados 3D para se determinar as cores e evitar a colisão com a planta. Então, o braço é posicionado na posição correta para a coleta do fruto. Uma câmera comum e outra 3D, localizadas na extremidade do braço, auxiliam na precisão do movimento a fim de não danificar a pimenta. Em testes de laboratório, os sensores de obstáculos tiveram precisão de 1cm. Porém, nos testes realizados dentro da casa de vegetação, a precisão foi de 4,5 cm. Como resultado final, os autores propõem que mais estudos sejam realizados para aumentar a precisão do sistema.

O projeto de sistemas que realizem a navegação autônoma dos robôs móveis nas casas de vegetação é complexo e importante nesse contexto. Como a principal tarefa de um robô móvel é a locomoção, a escolha de métodos que irão realizar essa atividade é crucial (GOZALEZ et al., 2009). Não só em termos de hardware, definindo a formas de locomoção (rodas, esteiras) mas também de software, planejando caminhos ou trajetórias, o sistema deve ser projetado para possibilitar a melhor maneira de conferir a mobilidade requerida ao robô para que realize a sua tarefa de maneira eficiente (METHA; BURKS; DIXON, 2008; SALES; OSÓRIO; WOLF, 2011). Gaspareto (2013) expõe que pesquisas e trabalhos vêm visando desenvolver tecnologias para que robôs móveis, inseridos em casas de vegetação, possam detectar paredes, realizar curvas e manobras nos corredores e evitar colisões com os elementos da planta. Uma meta importante no desenvolvimento de sistemas de navegação em estufas agrícolas é a detecção de

obstáculos no seu interior (GASPARETO, 2013). Portanto, no contexto geral, espera-se que a robótica móvel contribua cada vez mais para o desenvolvimento e otimização das atividades agrícolas.

### 3.2 Planejamento de caminhos e trajetórias

Cada vez mais é necessário que os robôs móveis se desloquem rapidamente a fim de otimizar a produção e reduzir os custos dos processos em que estão envolvidos (GASPARETTO et al., 2015). Porém, esse aumento de velocidade ocasiona em dificuldades e problemas para a realização das manobras de locomoção, exigindo muito das engrenagens, atuadores e processamento do microcontrolador. Em situações de desgaste, o tempo de vida útil do robô diminui e as tarefas não são executadas com qualidade. Assim, planejar a movimentação por caminhos suaves que possibilitem uma movimentação rápida, sem exigir muito do robô, se torna um desafio importante para a robótica móvel (PEDROSA, 2003).

Gasparetto et al. (2015) mostram que planejar um caminho específico é diferente de planejar uma trajetória. Enquanto que no planejamento de caminho leva-se em conta apenas o ponto inicial, de onde o robô parte, e o ponto final, onde ele deve chegar, no planejamento de trajetória deve-se levar em conta variáveis temporais, por exemplo, aceleração e velocidade, no cálculo do deslocamento.

O planejamento da movimentação do robô pode ser realizado de duas maneiras: *on-line* ou *off-line* (RAJA, 2012). Quando realizado *off-line*, o desenvolvedor já possui todos os conhecimentos sobre a estrutura da planta por onde o robô irá trafegar, tais como posição de obstáculos, curvas e outras limitações. Caso existam outros objetos que se locomovam na planta, além do robô, suas trajetórias também já serão conhecidas previamente. No planejamento *on-line*, nem todas as informações da planta são conhecidas, sendo necessária a realimentação através de sensores e outras comunicações para se tomar decisões. Operações que envolvem planejamento de movimento *on-line* requerem mais recursos em relação ao planejamento *off-line*. Como exemplos de situações em que o planejamento *off-line* é utilizado pode-se citar usinas nucleares, dutos de transporte de material, fábricas, ambientes com poucas mudanças em sua configuração e pouca movimentação. Mas de acordo com Masehian et al. (2007), devido à alta dinâmica e complexidade dos ambientes de aplicação atuais, as abordagens *on-line* têm sido mais estudadas e desenvolvidas. Outro fator que contribui para isso é, em certas situações, o conhecimento incompleto dos parâmetros do ambiente, o que torna necessário o acompanhamento em tempo real do robô para a tomada de decisões (

HACHOUR, 2008). Atividades como exploração planetária, reconhecimento de áreas e mineração são exemplos de situações onde não se obtém todas as informações prévias da planta.

Uma das ferramentas mais usadas para simplificar a simulação da estrutura do robô é o “Espaço C” (LOZANO-PEREZ; WESLEY, 1979). Nessa abordagem, também conhecida por espaço de configuração, o robô é representado pontualmente, ou seja, suas dimensões e formatos não serão levados em conta, transformando o planejamento de movimento num problema bidimensional. A fim de se compensar essa representação pontual do robô, todos os obstáculos são ampliados pela dimensão do robô (RAJA; PUGAZHENTHE, 2012). Santana (2007) mostra que existem duas categorias de espaço dentro dessa abordagem que auxiliam no planejamento do movimento. Os *Cobs* se referem ao conjunto de possibilidades de interação entre o robô e os obstáculos. Já os *Cfree* se referem ao conjunto de configurações onde não existe essa interseção .

A representação do ambiente é outro fator muito importante para se planejar uma estratégia de movimentação, e ela pode ser realizada utilizando dois tipos de mapa – topológico e métrico (THRUN, 2002; WOLF et al., 2009). Os topológicos utilizam o mecanismo de grafos para representar um cenário, sendo os nós as possíveis localidades a serem alcançadas, e as linhas as vias que as conectam (MATARIC 1990; KUIPERS; BYUN, 1991). Wolf et al. (2009) expõem que, embora forneçam representações pouco detalhadas e simplificadas, os mapas topológicos resultam como opções satisfatórias quando aplicados na geração de trajetórias. A representação mais detalhada e complexa é feita utilizando mapas métricos, como por exemplo as Grades de Ocupação (*Occupancy Grids*; ELFES 1989). Ela consiste na representação bidimensional da planta, dividindo o espaço de operação em subáreas chamadas de células (MORAVEC; ELFES, 1985; MORAVEC, 1989; BORENSTEIN; KOREN, 1991; MARTIN; MORAVEC, 1996; WOLF et al., 2009). Thrun (2002) demonstra que, utilizando o sistema binário, o estado “ocupado” ou “livre” de cada célula pode ser representado. Em seguida, esses estados são dispostos numa matriz que irá representar de maneira robusta e geral o espaço de movimentação do robô (ALVES, 2015).

Os algoritmos usados para fazer o planejamento de trajetórias estão relacionados às funções de otimização que trabalham com tempo mínimo e/ou menor gasto energético (GASPARETTO et al., 2013). Já os que realizam planejamento de caminho estão divididos entre métodos que utilizam *roadmap* (rotas), campo potencial artificial e decomposição de células (CHOSSET et al., 2005; LAVALLE, 2006).

Os métodos de *roadmap* reduzem o espaço de configuração a apenas uma dimensão (GASPARETTO et al., 2015). Eles mapeiam os *Cfree's* formando um roteiro de curvas unidimensionais, o qual possui o conjunto de caminhos possíveis que será vinculado ao ponto de saída e chegada do robô, formando o caminho final a ser percorrido. Pode-se associar alguma função de otimização para obter o caminho mais curto (CANNY; DONALD, 1988; TAKAHASHI; SCHILLING, 1989).

Os métodos de decomposição de células subdividem o espaço livre em regiões menores (células), formado, entre elas, a partir da relação de adjacência umas com as outras, um gráfico de conectividade. Os nós representam os espaços livres e são interligados somente se forem adjacentes. Assim, realiza-se a pesquisa no gráfico pelo caminho desejado a partir de um número finito de opções (GASPARETTO et al., 2015).

Na teoria do campo potencial, o robô é considerado como um ponto móvel sob influência de um campo (KATHIB, 1985). Esse campo é devido à disposição geográfica dos obstáculos na planta. O destino final do robô é considerado como um potencial atraente, o qual será somado ao potencial repulsivo dos obstáculos gerando o resultado final, que será chamado de potencial total. A partir daí, o potencial total é interpretado como uma força que aproxima o robô do seu objetivo e evita os obstáculos. (GASPARETTO et al., 2015; KHATIB, 1985; VOLPE, 1990; VOLPE; KHOSLA, 1990; GASPARETTO et al., 2015).

### **3.3 Algoritmos usados para geração de caminhos e o MRP**

Segundo Junior (2018), os principais algoritmos utilizados para geração de caminhos são o algoritmo A\* (lê-se A-estrela), campo potencial, Árvores Aleatórias de Rápida Exploração (RRT), algoritmos genéticos e Mapas de Rotas Probabilístico (MRP). Baseado em critérios de alta robustez, praticidade, possibilidade de execução *off-line* em plantas estáticas, desenvolveu seu trabalho utilizando o MRP em vários ambientes simulados e obteve resultados altamente satisfatórios (JUNIOR, 2018).

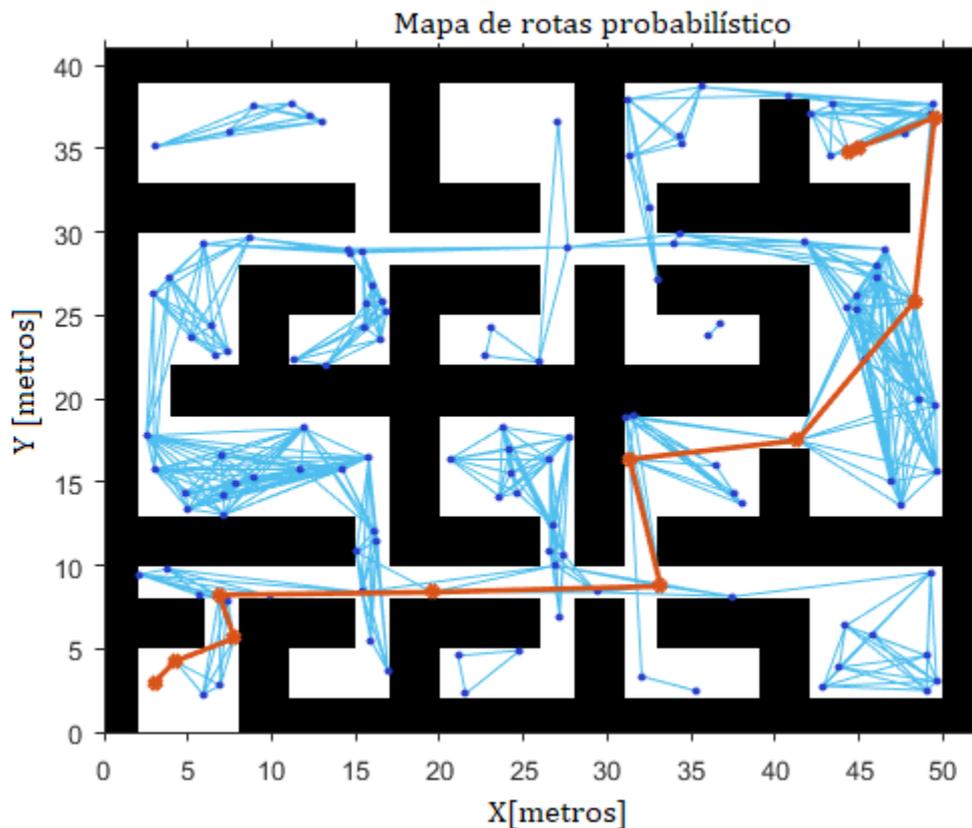
Paiva (2007) aponta o MRP como sendo um dos algoritmos para planejamento de caminho com melhor rendimento em cenários variados. Kvraki et al. (1998) mostram que, por ter sido um dos primeiros algoritmos viáveis em aplicações no mundo real, ele é largamente utilizado para planejar caminhos livres de colisões para quaisquer tipos de robôs móveis em plantas com obstáculos imóveis. Também é utilizado para o cálculo de trajetórias de robôs operando em ambientes estáticos.

O funcionamento do MRP começa com um grafo  $R(N,E)$ , onde  $(N)$  é o número de nós,  $(R)$  e  $(E)$  as arestas, situado no espaço livre de um conjunto de configurações  $m$ -dimensional, sendo  $(m)$  o número de graus de liberdade do robô (ADÔRNO, 2008).

A primeira fase é a de aprendizado, onde ocorre a construção da rota a partir de amostras aleatórias da configuração do robô (nós) geradas, as quais serão adicionadas aos conjuntos de nós do grafo  $(N)$ . Um planejador local conecta esses nós, através de segmentos de retas, com seus nós vizinhos desde que estejam a uma distância máxima já definida (KAVRAKI et al., 1996; KAVRAKI; KOLOUNTZAKIS; LATOMBE, 1998). As conexões que atenderem a esse critério são adicionadas ao conjunto de arestas  $(E)$  do grafo.

A próxima fase é chamada de questionamento e se resume em formar o caminho entre o ponto inicial e final através do grafo gerado (SOUSA, 2017). Assim, a fase de aprendizado é a mais demorada e que consome maiores recursos computacionais. A fase de questionamento demanda pouco tempo e não requer processamento elevado. Junior (2018) fez várias simulações aplicando o MRP como é exemplificado na Figura 8.

Figura 8 - Exemplo de caminho gerado pelo algoritmo MRP



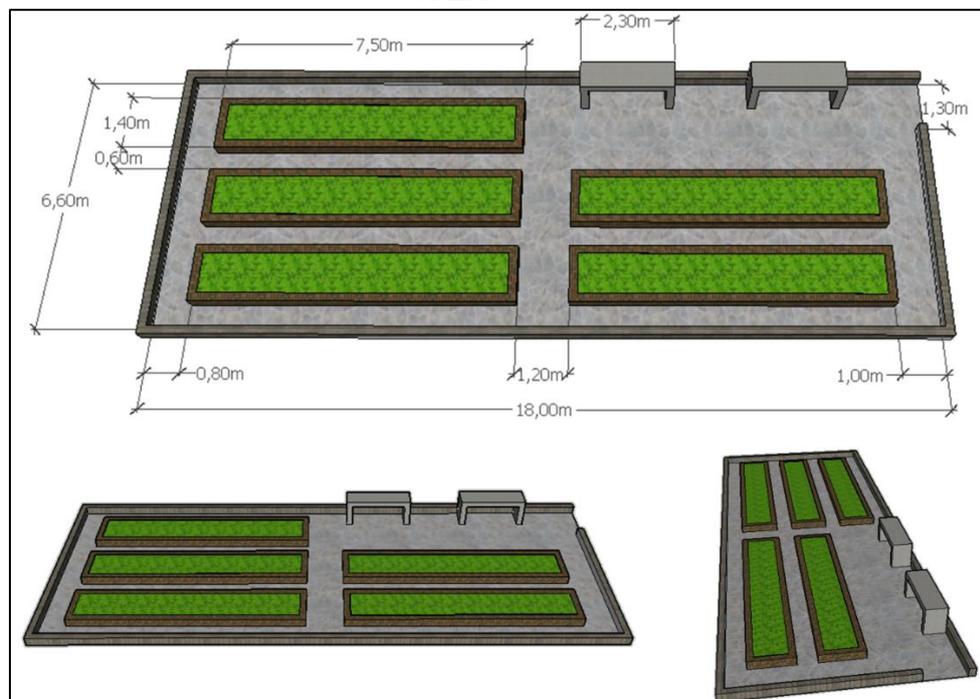
Fonte: Adaptado de <https://www.mathworks.com/help/robotics/examples/path-planning-in-environments-of-different-complexity.html>

Souza (2017) mostra que novos caminhos podem ser buscados a partir do grafo gerado na fase de aprendizado, levando em conta novos pontos inicial e final. Barraquand et al. (2000) expõem que a probabilidade de gerar um caminho válido se aproxima de 1 se a quantidade de amostras tender ao infinito. No entanto, quanto maior a quantidade de amostras, maior será a demanda de processamento, afetando negativamente o tempo de operação. Assim, são necessários critérios corretos que determinem o espaçamento entre os pontos de forma a resultar em uma representação adequada para a aplicação sem prejudicar o gasto computacional (KAVRAKI et al., 1996).

Uma das vantagens do MRP é a velocidade com que fornece soluções para aplicações que envolvam áreas extensas. Contudo, as soluções fornecidas não são completamente otimizadas na maioria das vezes. Trabalhos foram feitos para tentar sanar essa questão, utilizando planejadores locais mais eficientes e melhores amostragens. O MAPRM (*Medial Axis Probabilistic Roadmap*), é uma variante do MRP que é especializada em planejar caminhos em locais estreitos (WILMARTH; AMATO; STILLER 1999).

Junior (2018) também simulou o uso do MRP em uma planta de uma casa de vegetação do Departamento de Fitopatologia da UFPA. A Figura 9 mostra a sua simulação.

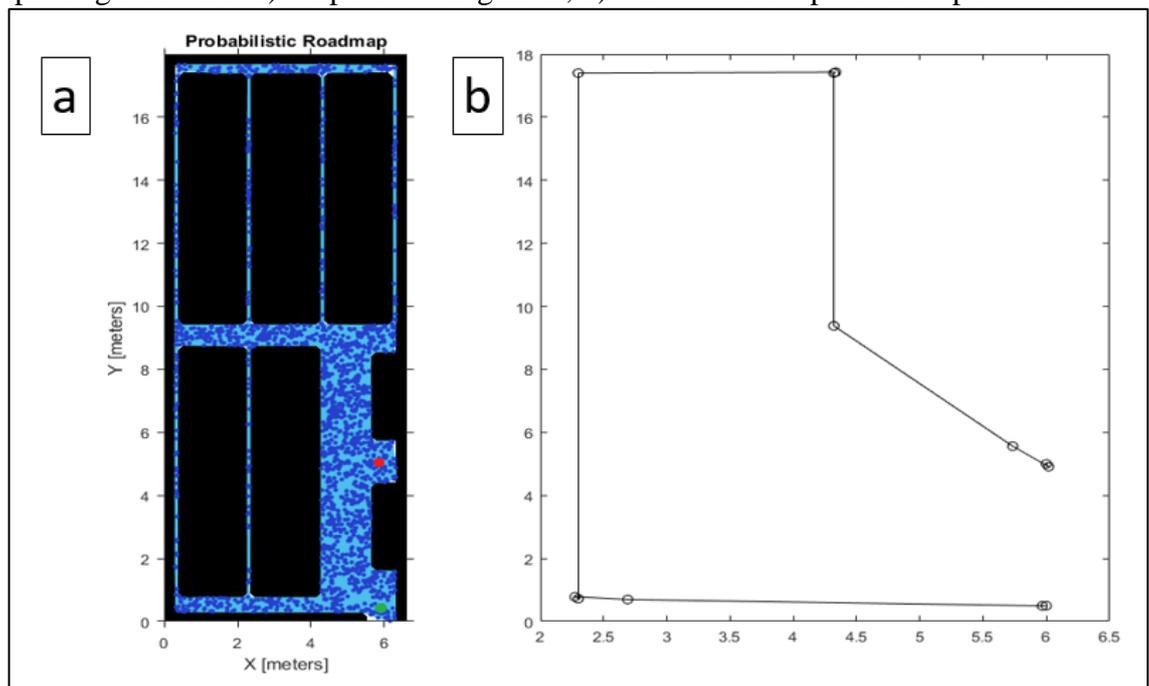
Figura 9 – Simulação 3D da casa de vegetação do Departamento de Fitopatologia da UFPA.



Fonte: Junior (2018).

O objetivo foi fazer o robô trafegar por pontos especificados por entre os corredores da estufa, simulando um aplicador de agentes químicos, partindo do ponto inicial especificado até o ponto desejado. Dessa forma, a fase de aprendizado foi executada apenas uma única vez e, a partir do grafo gerado, foram sendo determinados caminhos para que o robô circulasse por entre os corredores até chegar no seu destino. Na figura 10 é mostrado esse processo.

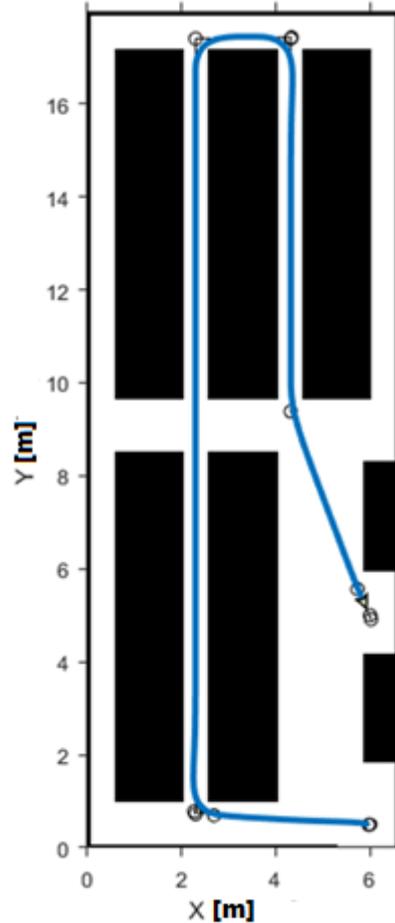
Figura 10 - Aplicação do MRP no mapa da casa de vegetação do Departamento de Fitopatologia da Ufla. a) Mapa de rotas gerado; b) Caminho a ser percorrido pelo robô



Fonte: Junior (2018).

O caminho final percorrido pelo robô simulado é representado na Figura 11:

Figura 12 - Caminho percorrido (em azul) pelo robô



Fonte: Júnior (2018)

Foram realizadas outras simulações em ambientes diferentes, inclusive com o mapa da UFLA, obtendo-se resultados satisfatórios em todos os casos. Por fim, o autor sugere a construção de modelos reais e aplicações do algoritmo com robôs móveis nessas plantas.

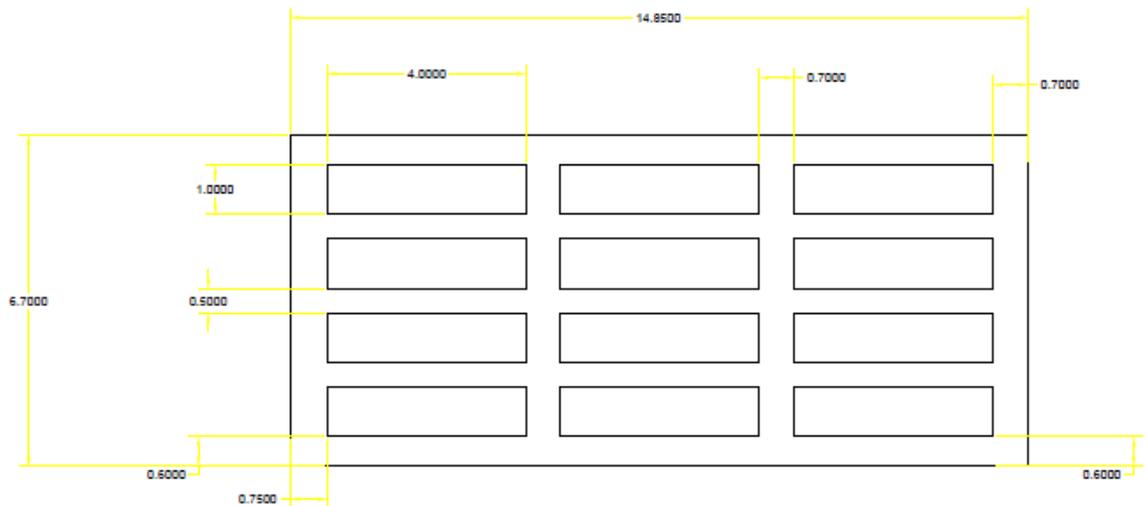
#### 4. MATERIAIS E MÉTODOS

O ambiente selecionado para a realização dos testes e validação do algoritmo foi uma casa de vegetação do Departamento de Recursos Hídricos da UFLA. A partir da modelagem e geração da imagem em bitmap, o algoritmo MRP será aplicado a fim de se determinar o melhor caminho dentro dos contextos requeridos e diferentes níveis de complexidade. Depois de validar o algoritmo no ambiente virtual, serão realizados os testes numa planta em escala reduzida da mesma casa de vegetação. É importante ressaltar que o todos os testes e validações já são realizados com total conhecimento prévio do ambiente e suas condições estáticas.

#### 4.1 Descrição dos ambientes

Após a visita nas instalações da casa de vegetação e realização de medidas, foram obtidos os seguintes valores para suas dimensões: 6,7m de largura, 14,85m de comprimento e área interna de 99,5m<sup>2</sup>. Em seu interior existem 12 canteiros de 1m de largura por 4m de comprimento, separados 0,7m longitudinalmente e 0,5m em latitude. Na Figura 12 é mostrado a representação da planta da casa de vegetação.

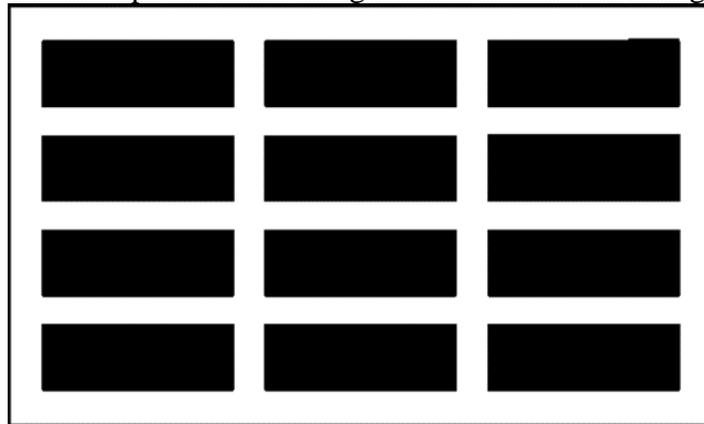
Figura 12 – Modelagem da casa de vegetação do Departamento de Recursos Hídricos da UFLA



Fonte: Do autor (2019).

O algoritmo MRP utiliza a representação do ambiente em *bitmap* (mapa de bits), onde cada bit descreve uma informação da imagem. A representação em *bitmap* da planta é mostrada na Figura 13.

Figura 13 – Mapa de bits da imagem referente à casa de vegetação



Fonte: Do autor (2019).

## 4.2 O algoritmo MRP

O objeto de trabalho do algoritmo MRP é a imagem discreta (*bitmap*) do ambiente. O software utilizado para rodar esse algoritmo foi o Matlab. No código, foi criada uma Grade de Ocupação (*Binary Occupancy Grid*) que delega o valor 1 para regiões (pixels) ocupadas da imagem, e o valor de 0 para as livres. Conseqüentemente, esse será o padrão para traçar o caminho, onde só serão utilizadas as regiões que receberam 0 (livres). A resolução do mapa foi determinada em células por metro, ou seja, cada célula representa uma fração da Grade de Ocupação Binária. Assim, a imagem é passada como entrada no início da função principal no campo *map*. Nessa etapa, é preciso garantir que a extensão do arquivo da imagem seja “.btm” e que ele esteja na mesma pasta que contém as funções do algoritmo. O próximo passo é informar, no código da mesma função, o local de partida do robô. Isto é realizado no campo *source*. O algoritmo trabalha com a estrutura bidimensional do ambiente em coordenadas cartesianas, sendo necessário passar, primeiramente, a coordenada referente ao eixo y e depois a do eixo x. Deve ser informado, no campo *goal*, o destino final do robô. Para isso, também é utilizado, da mesma maneira, o sistema de coordenadas do campo *source*. Por fim, é necessário informar a quantidade de nós desejados. Na Figura 14, é feita uma representação do código com suas linhas para declaração da imagem, parâmetros de posição inicial, final e número de nós.

Figura 14 – Exemplo de declaração dos parâmetros de posição inicial, final, número de nós e imagem.

```
map=im2bw(imread('ESTUFA.bmp'));
source=[19 16];
goal=[150 600];
k=200;
display=true;
```

Fonte: Do autor (2019).

O código é executado, sendo a primeira etapa (aprendizado) responsável por bombardear a figura com os nós e, respeitando os critérios de espaços livres, gera todos os caminhos possíveis. O sistema exibe a imagem com esses caminhos e pede ao usuário que pressione qualquer tecla para a realização da próxima etapa, a qual será a de questionamento. É gerado e exibido, na tela, o caminho final que levará o robô do seu ponto inicial até o destino, no menor percurso possível.

#### 4.1.2 Testes de validação do MRP

Para a realização dos testes com o robô móvel, foi necessário, primeiramente, validar o algoritmo juntamente à planta da casa de vegetação do Departamento de Recursos Hídricos da UFLA. Essa etapa serve para confirmar se existem erros lógicos ou de sintaxe em alguma parte do código (BARNES, 2009). Erros lógicos são erros que produzem resultados não desejados. Por exemplo, um caminho que extrapole o limite entre um espaço livre com outro ocupado. Já um erro de sintaxe está relacionado às falhas durante a escrita na linguagem do programa. Esses erros são mais fáceis de serem encontrados, pois são apontados pelo compilador ainda na etapa de compilação. Os erros lógicos são mais difíceis pois dependem da análise do operador do teste e só são visíveis depois da execução do programa.

Foram realizados 3 testes, variando os parâmetros de entrada de posição inicial, final, número de nós e alterando a complexidade da planta com obstáculos nos caminhos livres. Todos os resultados serão analisados posteriormente.

##### 4.1.2.a) Teste 1

Para o primeiro teste, não foram incluídos obstáculos no ambiente e foram realizadas 2 iterações variando o número de nós. A posição inicial foi 152, no eixo y, e 311, referente ao eixo x. O restante dos parâmetros e resultados são mostrados na Tabela 1. Foram realizados dois casos (1 e 2) onde o número de nós foi mudado.

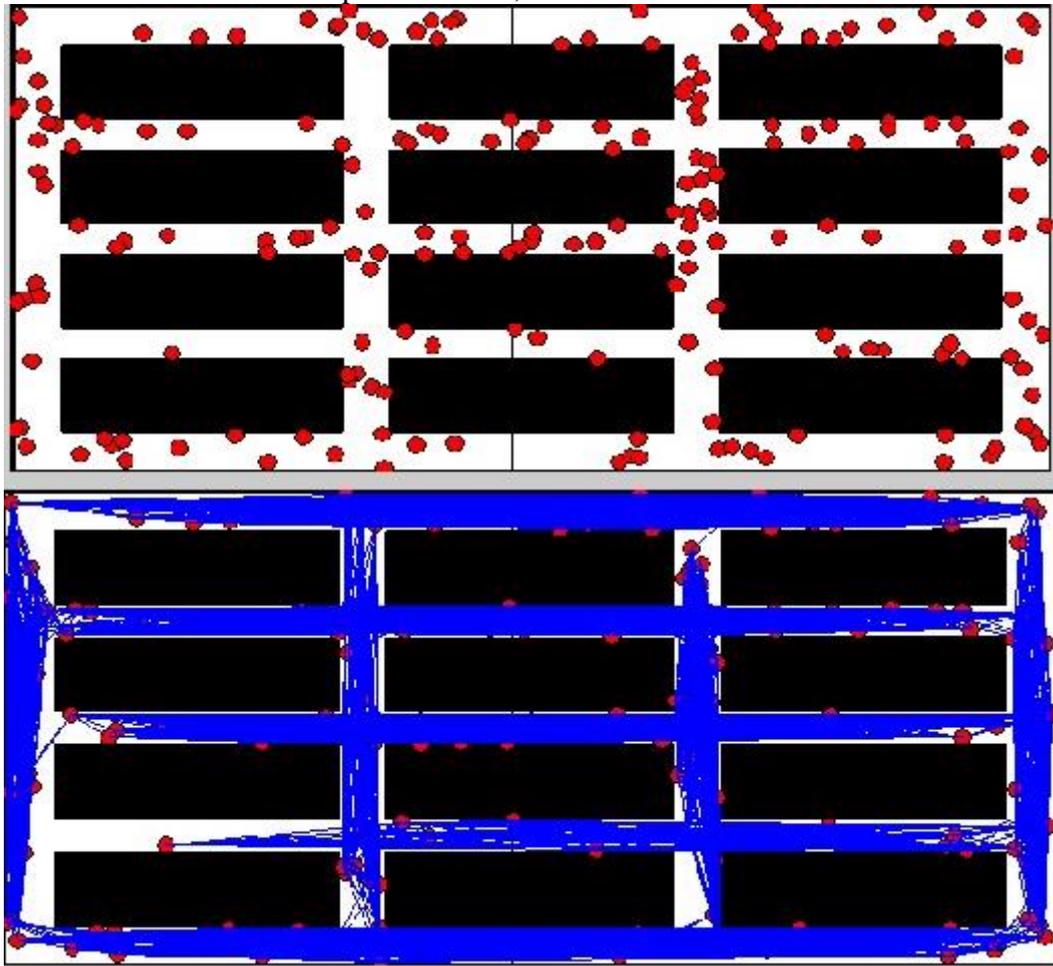
Tabela 1: Parâmetros de entrada e resultados do Teste 1

Caso	Número de nós	Tempo de execução da etapa de aprendizagem (s)	Tempo de execução da etapa de questionamento (s)	Distância euclidiana (m)
1	200	23,59	0,968	9,12
2	400	126,56	0,106	8,46

Fonte: do autor (2019)

A Figura 15 mostra a fase de “bombardeamento” e aprendizado do caso 1.

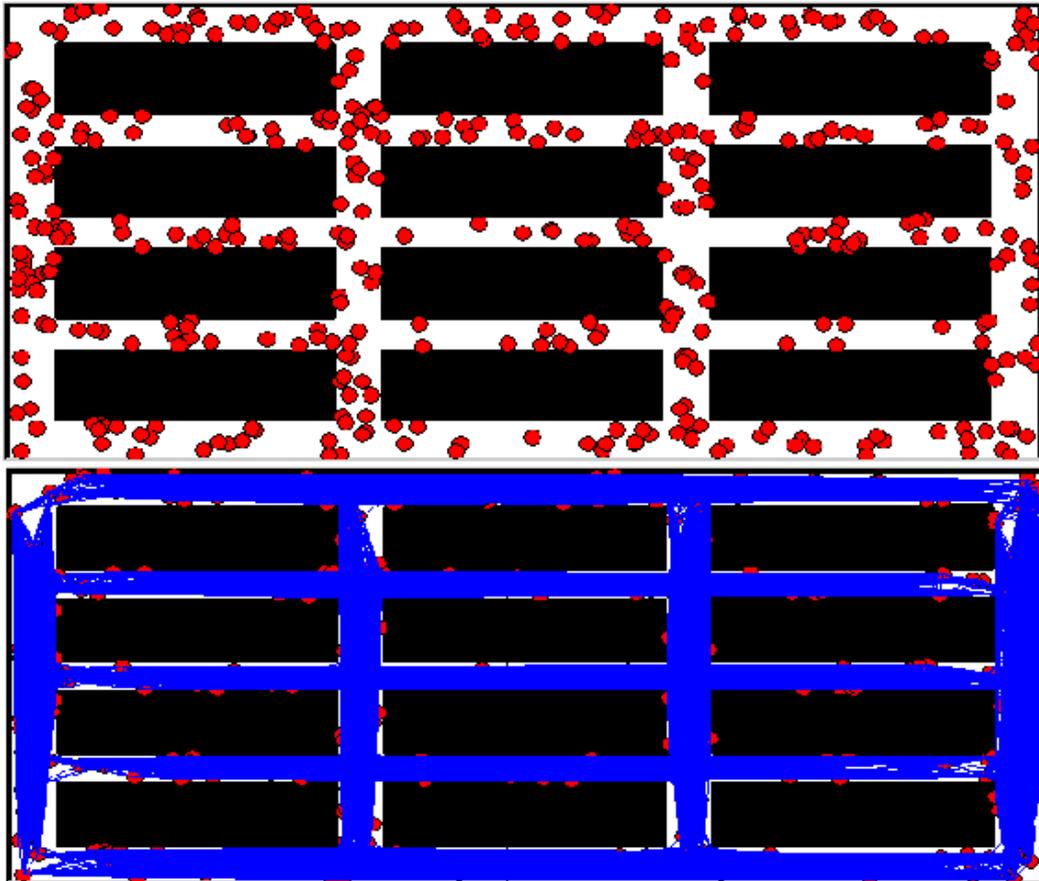
Figura 15 – Fases de “bombardeamento” e aprendizado, respectivamente, do caso 1.



Fonte: Do autor (2019).

A foto da Figura 16 mostra as fases de “bombardeamento” e aprendizado do caso 2.

Figura 16 - Fases de “bombardeamento” e aprendizado, respectivamente, do caso 2.



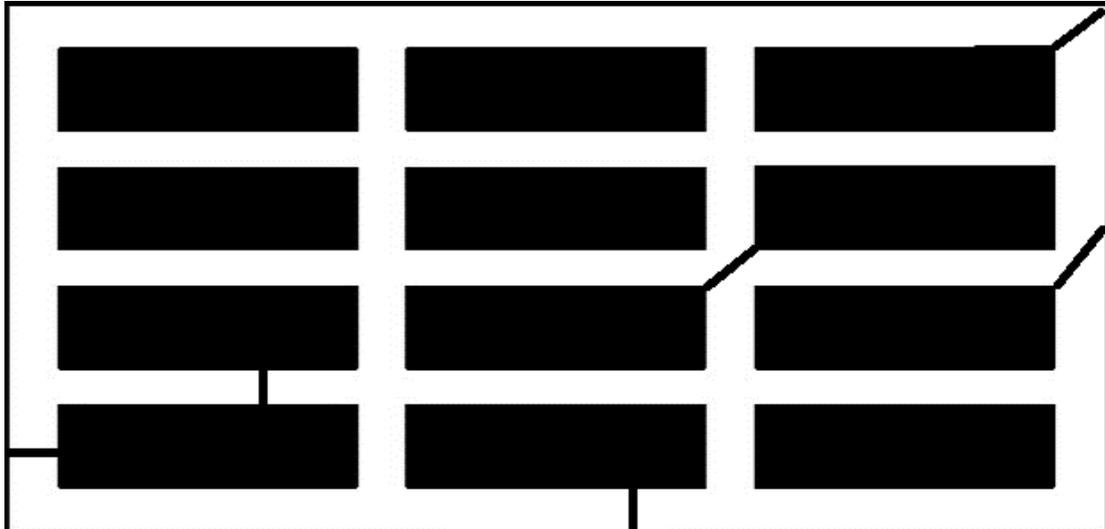
Fonte: Do autor (2019).

Os caminhos finais gerados dos casos 1 e 2, respectivamente, são representados na Figura 17.



É mostrado na Figura 18 o desenho da planta com obstáculos impedindo certos caminhos.

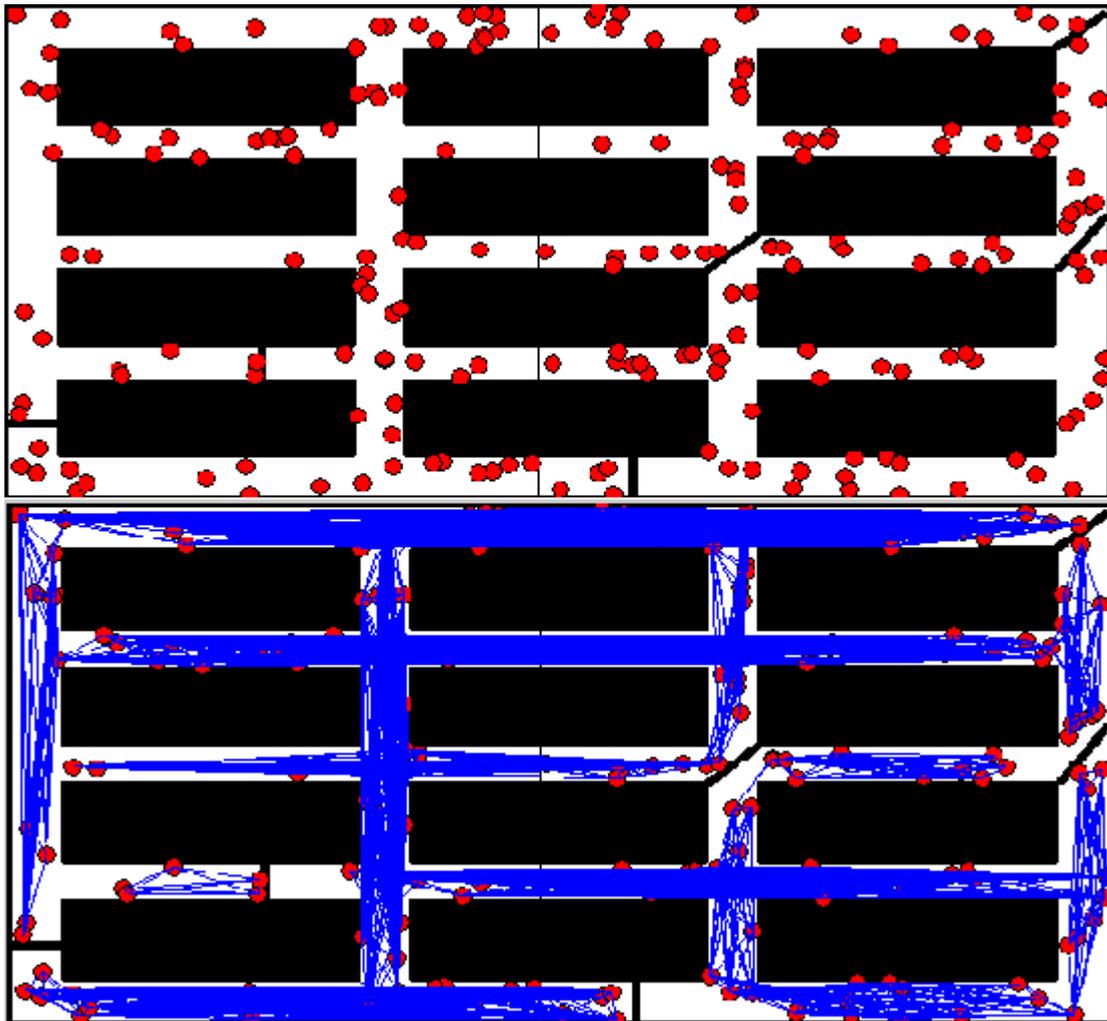
Figura 18 – Planta com obstáculos, representados por traços que impedem o trajeto



Fonte: Do autor (2019).

Na Figura 19, é possível ver as fases de “bombardeamento” e aprendizado do caso 1.

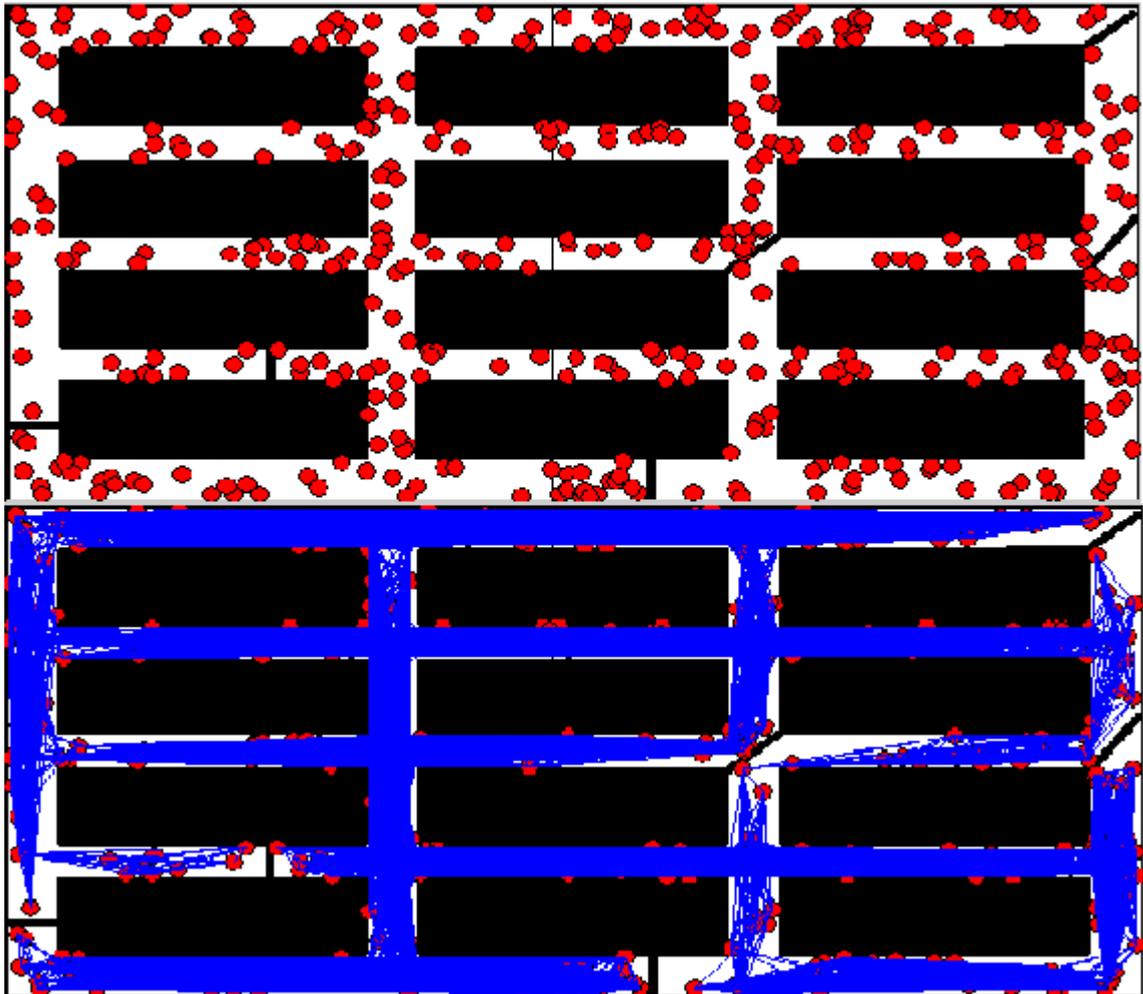
Figura 19 – Fases de “bombardeamento” e aprendizado do Caso 1, respectivamente.



Fonte: Do autor (2019).

A Figura 20 mostra as fases de “bombardeamento” e aprendizado do caso 2.

Figura 20 – Fases de “bombardeamento” e aprendizado do Caso 2, respectivamente.

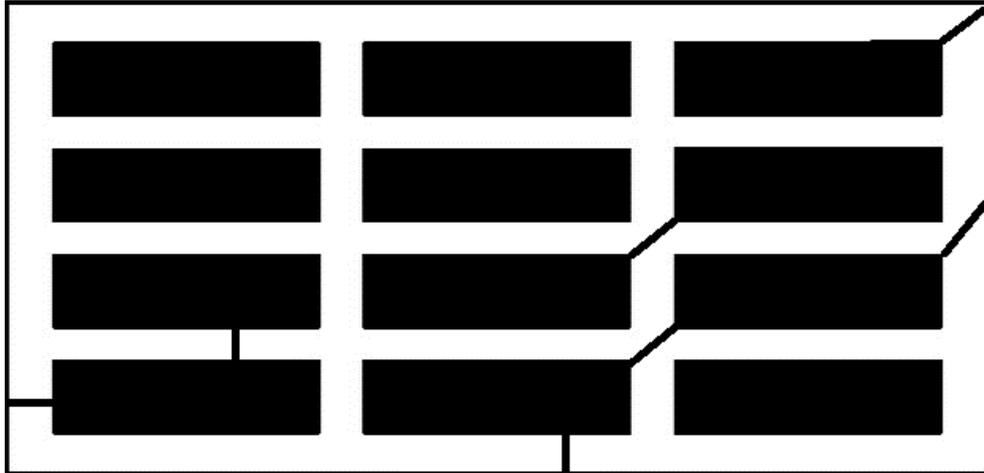


Fonte: do autor (2019).

Os caminhos finais dos casos 1 e 2, respectivamente, são mostrados na Figura 21.



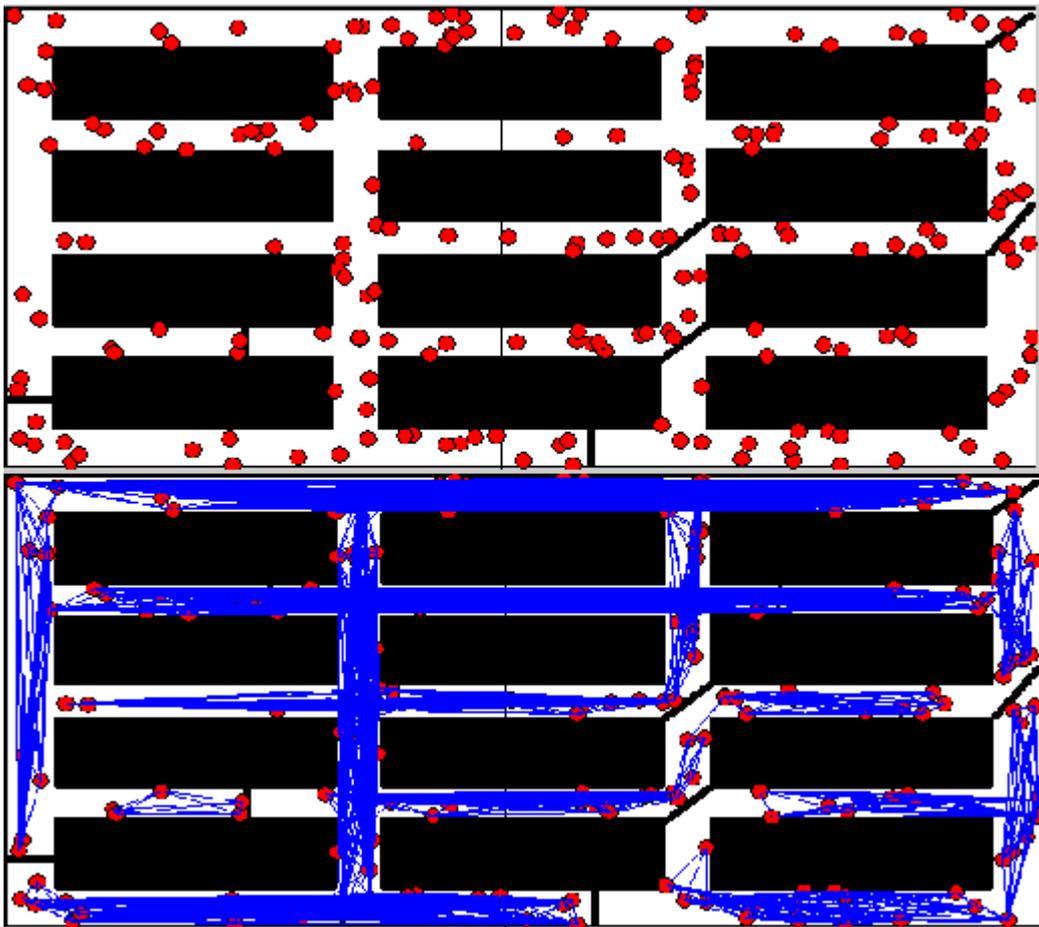
Figura 22 – Planta com todos os caminhos bloqueados por obstáculos.



Fonte: Do autor (2019).

Na Figura 23 é demonstrado as duas primeiras fases de aplicação do algoritmo na planta.

Figura 23 – Aplicação do algoritmo.



Fonte: Do autor (2019).

Porém, na última fase, o caminho não foi gerado e surgiu uma mensagem de erro na janela de comando, como mostra a Figura 24.

Figura 24 – Mensagem de erro “Caminho não encontrado”.

```
<
Command Window
click/press any key
click/press any key
Error using astart (line 79)
no path found
```

Fonte: do autor (2019).

### 4.3 Plataforma móvel

O robô móvel utilizado nesse trabalho foi montado a partir de peças adquiridas separadamente. Constitui, basicamente, de um sistema com dois motores e três rodas para locomoção, um chassi, micro controlador com Ponte H e alimentação à base de bateria. A Figura 25 mostra uma foto do robô utilizado.

Figura 25 – Robô utilizado no projeto.



Fonte: Do autor (2019).

#### 4.3.1 Descrição dos componentes

- **Microcontrolador Arduino Uno R3** - é uma placa baseada no microcontrolador Tmega328. Sua função é realizar o controle do dispositivo lendo dados de suas entradas analógicas e digitais e enviando sinais de comando a partir das saídas digitais. Possui quatorze pinos de entrada/saída digital, sendo que seis desses pinos podem ser usados como saída PWM (*Pulse Width Modulation*). Possui também outras seis entradas analógicas, cristal oscilador de 16MHz, entrada USB (*Universal Serial Bus*), conexão do tipo ICSP (*In-Circuit Serial Programming*) e um botão de reset para reiniciá-la. A programação é feita em linguagem C/C++. Devido à simplicidade e praticidade dos microcontroladores Arduino, na prototipagem de projetos, a sua popularidade é muito grande. Assim, existe grande disponibilidade de documentação e bibliotecas de programação, o que foi uma vantagem para o projeto. Na figura 26 é mostrado um Arduino Uno R3.

Figura 26 – Exemplo de uma placa Arduino Uno R3, como a usada no robô.

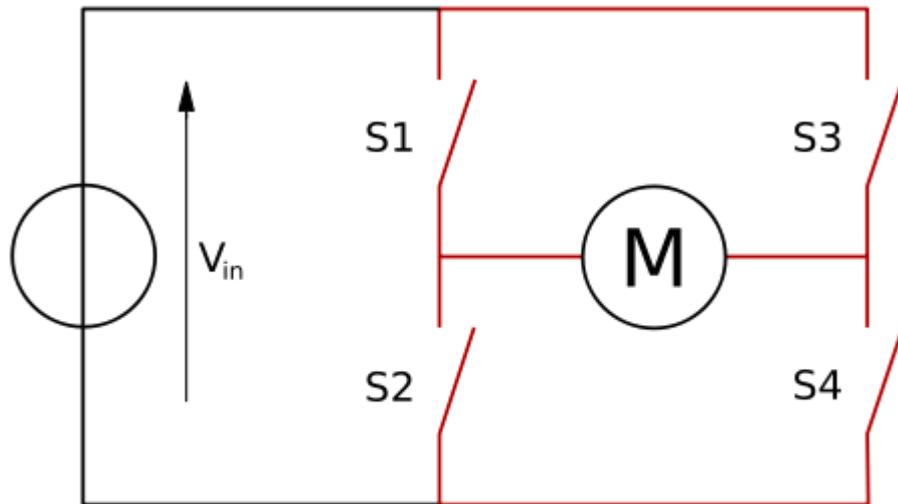


Fonte: Arduinio *homepage* ([www.arduino.cc](http://www.arduino.cc)).

- **Motor Shield L293D Driver (Ponte H)** – é um circuito elétrico que permite que uma tensão seja aplicada à uma carga, neste caso o motor, fazendo-o girar em qualquer direção e velocidade específica. Isso será necessário para controlar o sentido de giro para realizar manobras e definir velocidades de deslocamento. O funcionamento desse circuito pode ser

descrito analisando quatro chaves que fazem ligação entre o motor e fonte de alimentação, como mostra a Figura 27.

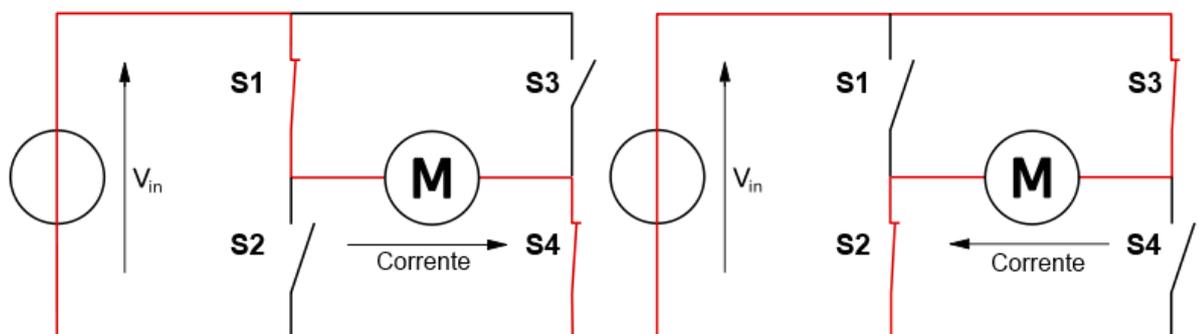
Figura 27 – Exemplificação do funcionamento de um circuito Ponte H.



Fonte: Wikipedia ([https://pt.wikipedia.org/wiki/Ponte\\_H](https://pt.wikipedia.org/wiki/Ponte_H))

Caso as chaves S1 e S4 forem fechadas, a corrente elétrica circulará no sentido da esquerda para a direita, fazendo o motor girar no mesmo sentido. Se forem desligadas essas duas chaves e forem ligadas S3 e S2, o sentido da corrente e giro do motor invertem. A Figura 28 mostra como os sentidos da corrente variam conforme o acionamento das chaves.

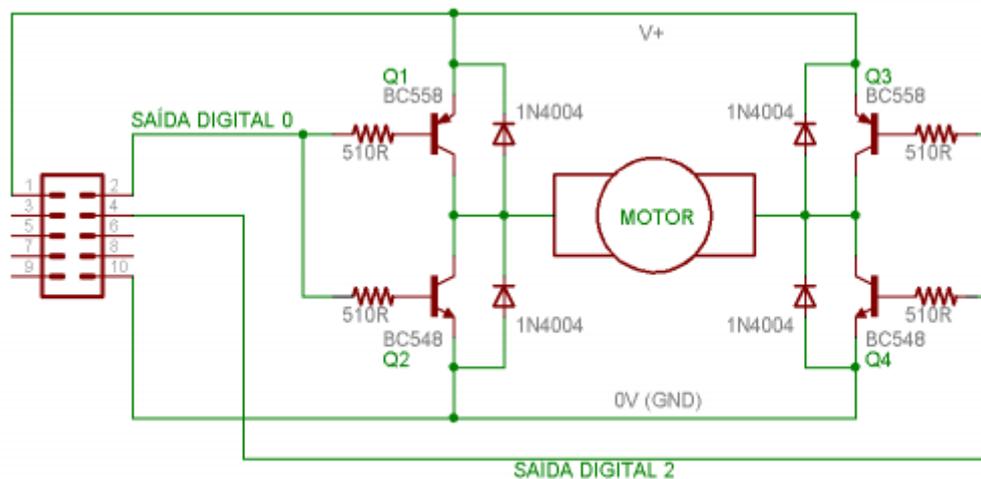
Figura 28 – Sentidos da corrente e do giro do motor variam conforme as chaves são acionadas.



Fonte: Wikipedia ([https://pt.wikipedia.org/wiki/Ponte\\_H](https://pt.wikipedia.org/wiki/Ponte_H))

No caso da placa L293D, em vez de chaves mecânicas, transistores e diodos realizam esse chaveamento. Quando a base de um transistor é polarizada, ele é capaz de conduzir corrente entre seus terminais Coletor e Emissor (BOYLESTAD, 1996). Se o transistor for do tipo NPN (Negativo – Positivo – Negativo), a condução da corrente será do Coletor para o Emissor. Caso ele seja do tipo PNP (Positivo – Negativo – Positivo), a corrente fluirá do Emissor para o Coletor. Os transistores NPN funcionam como condutores quando há um nível lógico alto em sua Base, e os do tipo PNP, quando o nível lógico é baixo. A Figura 29 mostra um exemplo de um circuito similar ao da placa.

Figura 29 – Circuito similar ao da placa, com transistores e diodos realizando o chaveamento.



Fonte: Maxwell – Bohr Instrumentação Eletrônica

A ponte está ligada às saídas digitais do microcontrolador. Q1 e Q3 são transistores do tipo PNP e Q2 e Q4 são do tipo NPN. Se o nível lógico na saída digital zero for 1 (alto), Q1 não conduzirá corrente, enquanto que Q2 poderá conduzir. Se o nível lógico na saída dois for baixo, Q3 conduzirá e Q4 não. Nessa configuração, haverá uma corrente fluindo entre Q3 e Q2 fazendo o motor girar num determinado sentido. Para invertê-lo, basta inverter o nível lógico das duas saídas. Se os sinais lógicos nelas forem iguais, o motor irá parar de girar pois serão acionados, simultaneamente, ou os dois transistores superiores ou os inferiores. Os diodos têm a função de drenar a corrente que ainda existirá quando um transistor é desligado, pois as propriedades indutivas do motor forçarão a passagem de corrente pelo circuito. Dessa forma, os diodos funcionarão como dispositivo de segurança. Apesar de todo esse mecanismo descrito ser trabalhoso e pouco prático, em termos de programação, o seu uso fica facilitado ao se incorporar, ao código do robô, a biblioteca *AFMotor.h*, a qual simplifica os comandos de alterar

sentido de giro e faz controle da velocidade através do PWM. Isso será discutido no tópico que abordará os comandos do robô.

- **Motor DC 3-6V** – é responsável por movimentar as rodas do robô (Figura 30). São utilizados dois motores acoplados ao chassi. Operam numa faixa de tensão entre 3V a 6V e corrente contínua. Possuem caixa de redução de 1:48. A 3V, consomem 3mA de corrente, produzindo torque de 0,35Kgf/cm, cada um. A 6V, consomem 120 mA com torque de 0,8Kgf/cm. Sem carga, vão de 90 a 200 rotações por minuto (RPM). Seu peso é de 30g, o que não traz dificuldades com peso para a locomoção da plataforma. É um tipo de motor facilmente encontrado no mercado de peças para projetos de robótica. Seu controle é feito a partir da ponte H L293D. Na Figura 30 é mostrado um Motor DC 3-6V.

Figura 30 – Motor DC 3-6V.



Fonte: Robocore ([https://www.robocore.net/loja/motores/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo?gclid=CjwKCAjw\\_YPnBRBREiwAIP6TJ84-ao8IiwMs0iIpQK1ZAbDJek6Izpx4-dbcFhLcBtmL9qGXUO--xoCn\\_AQAvD\\_BwE](https://www.robocore.net/loja/motores/motor-dc-3-6v-com-caixa-de-reducao-e-eixo-duplo?gclid=CjwKCAjw_YPnBRBREiwAIP6TJ84-ao8IiwMs0iIpQK1ZAbDJek6Izpx4-dbcFhLcBtmL9qGXUO--xoCn_AQAvD_BwE)).

- **Chassi 2WD e rodas** – constitui a estrutura mecânica do robô, onde estão acoplados todos os outros dispositivos. O chassi possui 21,2cm de comprimento por 15,2cm de largura e é confeccionado em acrílico. Em sua extremidade traseira, possui uma pequena roda móvel, chamada de “roda boba”, por não possuir mecanismo de controle, servindo para dar sustentação à estrutura e auxiliar os movimentos. As rodas dianteiras são de borracha, com 68mm de diâmetro e 25mm de largura.

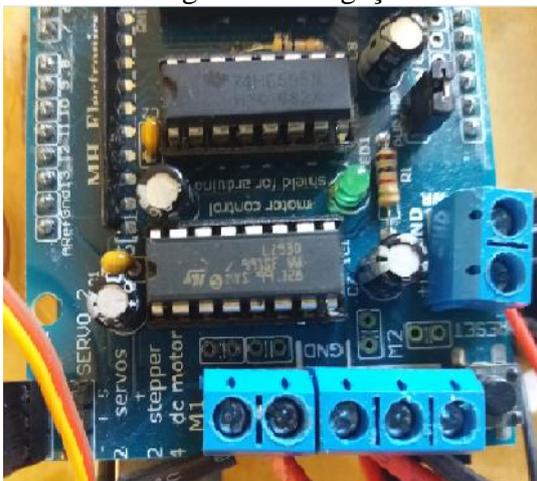
A alimentação do robô foi realizada com uma bateria de 9V. As ligações entre os componentes e o microcontrolador foram feitas com cabos de conexão (*jumpers*). Foi adicionado também, a fim de se proteger e conservar o sistema, um interruptor para ligar ou desligar a bateria no sistema.

#### 4.4 Código de programação do robô

A programação foi realizada na IDE (*Integrated Development Environment*) do Arduino utilizando a linguagem C++. Para realizar as operações sobre os motores, em conjunto com a ponte H, foi necessário o uso da biblioteca *AFMotor.h*, a qual media a comunicação entre o usuário e a placa. Através do comando “*import*” ela é adicionada ao código e pode-se utilizar os seus comandos:

- **AF\_DCMotor** – utilizado na primeira parte do código para definir em qual pino o motor está ligado. O motor esquerdo foi conectado aos pinos referentes a M1 (*jumper* vermelho polo positivo, *jumper* preto negativo). O motor direito foi acoplado nos pinos de M2. Eles foram nomeados como motor1 e motor2, respectivamente. Nessa sintaxe, utiliza-se o argumento 1 para o caso de estar ligado em M1, e 2 para M2. A Figura 31 exibe a ligação dos motores na placa e o código de programação.

Figura 31 – Ligação dos motores na placa e código na IDE do Arduino



```
#include <AFMotor.h> // inclui biblioteca de manipulação de motores DCs.

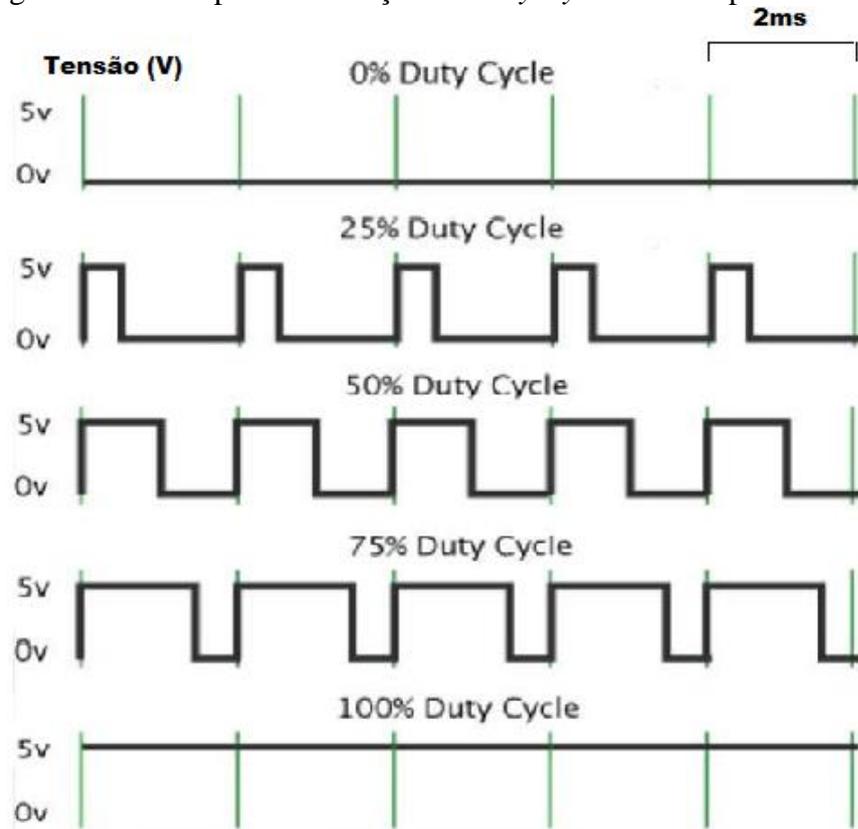
//Definindo os pinos
AF_DCMotor motor1(1); // Define o motor1 ligado ao M1
AF_DCMotor motor2(2); // Define o motor2 ligado ao M2
```

Fonte: Do autor (2019).

- **Método “motor.setSpeed( )”** – método utilizado para definir a velocidade do robô, a partir do PWM. Como o Arduino é um controlador digital, a técnica de PWM é empregada para se obter resultados analógicos a partir de entradas digitais. Consiste em produzir uma onda quadrada, de alta frequência, cujo tempo em que ela permanece em nível lógico 1 pode ser controlado. Com a variação da porcentagem desse tempo, conhecido como *Duty Cycle*, é possível conseguir valores entre o mínimo e máximo da tensão total disponível, por meio da alteração do valor médio da onda obtida. No caso do Arduino, esses valores (mínimo e máximo) vão de 0V (0% do *Duty Cycle*) a 5V (100% do *Duty Cycle*). A frequência no

microcontrolador é de 500Hz, sendo o período de 2ms (linha verde) como mostrado na Figura 32.

Figura 32 – Exemplos da variação do *Duty Cycle* e ondas produzidas.



Fonte: Do autor (2019).

O valor da onda média pode ser calculado multiplicando a tensão máxima ( $V_{max}$ ) pelo *Duty Cycle*. Por exemplo, um *Duty Cycle* de 80%, nessa aplicação, produziria uma tensão de 4V. Na sintaxe do código, o valor desse ciclo de trabalho é fornecido como sendo um valor entre 0 (mínimo) e 255 (máximo). Isso deve-se ao fato dele ser armazenado em 8 bits pelo microcontrolador. No código, foi definida uma variável inteira “*velocidadeMotores*” para receber esse valor. Em seguida, ela é passada como parâmetro para o método “*motor1.setSpeed*” e “*motor2.setSpeed*”, como mostrado na figura 33.

Figura 33 – Velocidade sendo passada como parâmetro, em destaque, no código.

```
// inclusão de bibliotecas.
#include <AFMotor.h> // inclui biblioteca de manipulação de motores DCs.

//Definindo os pinos
AF_DCMotor motor1(1); // Define o motor1 ligado ao M1
AF_DCMotor motor2(2); // Define o motor2 ligado ao M2
int velocidadeMotores = 80;
void setup() {
  Serial.begin(9600); // inicializa a comunicação serial para mostrar dados
  motor1.setSpeed(velocidadeMotores); // Define a velocidade para os motores.
  motor2.setSpeed(velocidadeMotores);
}
}
```

Fonte: Do autor (2019).

- **Método “motor.run() –** método utilizado para fazer o robô se locomover. É utilizado tanto para andar para frente, para trás ou parar. Se o argumento for “*FORWARD*”, o sentido de giro das rodas será para frente. Caso seja “*BACKWARD*”, será para trás. “*RELEASE*” faz os motores pararem. Esse método é utilizado em conjunto com um método básico chamado “*delay()*”. Ele é que determina por quanto tempo a ação irá acontecer (andar 1 segundo para frente, por exemplo). O parâmetro desse método deve ser um número inteiro que será interpretado como o tempo em milissegundos. Os movimentos de virar são realizados utilizando os parâmetros “*FORWARD*” e “*BACKWARD*”, ao mesmo tempo, em rodas diferentes. Se a intenção é virar à esquerda, o motor do lado direito deve girar para frente (*FORWARD*) enquanto que o da esquerda deve girar para trás (*BACKWARD*). O ângulo de giro será determinado pelo método “*delay()*”, através do parâmetro de tempo, que manterá o robô fazendo o giro. A rotina mostrada na figura 34, mostra um movimento de andar para frente durante 1 segundo, realizar uma curva para a esquerda de aproximadamente 45°, andar para frente outro segundo e parar. Importante manter qualquer rotina desse tipo dentro do “*void setup*”, pois será realizada apenas uma vez. Caso a coloque dentro do “*void loop*”, ela ficará se repetindo. Na Figura 34 é demonstrado um exemplo de código de movimentação do robô.

Figura 34 – Exemplo de movimentação para o robô.

```

// inclusão de bibliotecas.
#include <AFMotor.h> // inclui biblioteca de manipulação de motores DCs.

//Definindo os pinos
AF_DCMotor motor1(1); // Define o motor1 ligado ao M1
AF_DCMotor motor2(2); // Define o motor2 ligado ao M2
int velocidadeMotores = 80;
void setup(){
  Serial.begin(9600); // inicializa a comunicação serial para mostrar dados
  motor1.setSpeed(velocidadeMotores); // Define a velocidade para os motores.
  motor2.setSpeed(velocidadeMotores);
  motor1.run(FORWARD); // Gira motor esquerdo para frente.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(3000); // Prolonga a ação por 1 segundo.
  motor1.run(BACKWARD); // Gira motor esquerdo para trás.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(150); // Tempo aproximado para girar o robô 30°, nessa velocidade
  motor1.run(FORWARD); // Gira motor esquerdo para frente.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(3000); // Prolonga a ação por 1 segundo.
  motor1.run(RELEASE); // Pára motor esquerdo.
  motor2.run(RELEASE); // Pára motor direito.
}
void loop(){
  ,

```

Fonte: Do autor (2019).

#### 4.5 Testes com o robô

De posse de todas as informações descritas acima, alguns testes com o robô foram realizados a fim de se determinar uma velocidade padrão com tempos de deslocamento e de giro. Foi escolhido parâmetro de 80 para o método “*motor.setSpeed ( )*”. Nesse padrão, o robô anda em linha reta, com poucos erros, apresentando uma velocidade de 15 cm/s. Para realizar um giro de 45°, o parâmetro de tempo em “*delay ( )*” é de 600ms. Um giro de 90°, requer 1200 ms.

##### 4.5.1 – Pista de testes

Baseado na planta da casa de vegetação do Departamento de Recursos Hídricos da UFLA, foi construída outra, em escala menor, com 4,2m de comprimento e 2,7m de largura. As estufas têm dimensão de 60x30 cm, separadas por 60cm ao longo do comprimento e 30cm verticalmente. Na Figura 35, é possível ver como ficou a planta final.

Figura 35 – Planta em escala menor da estufa.

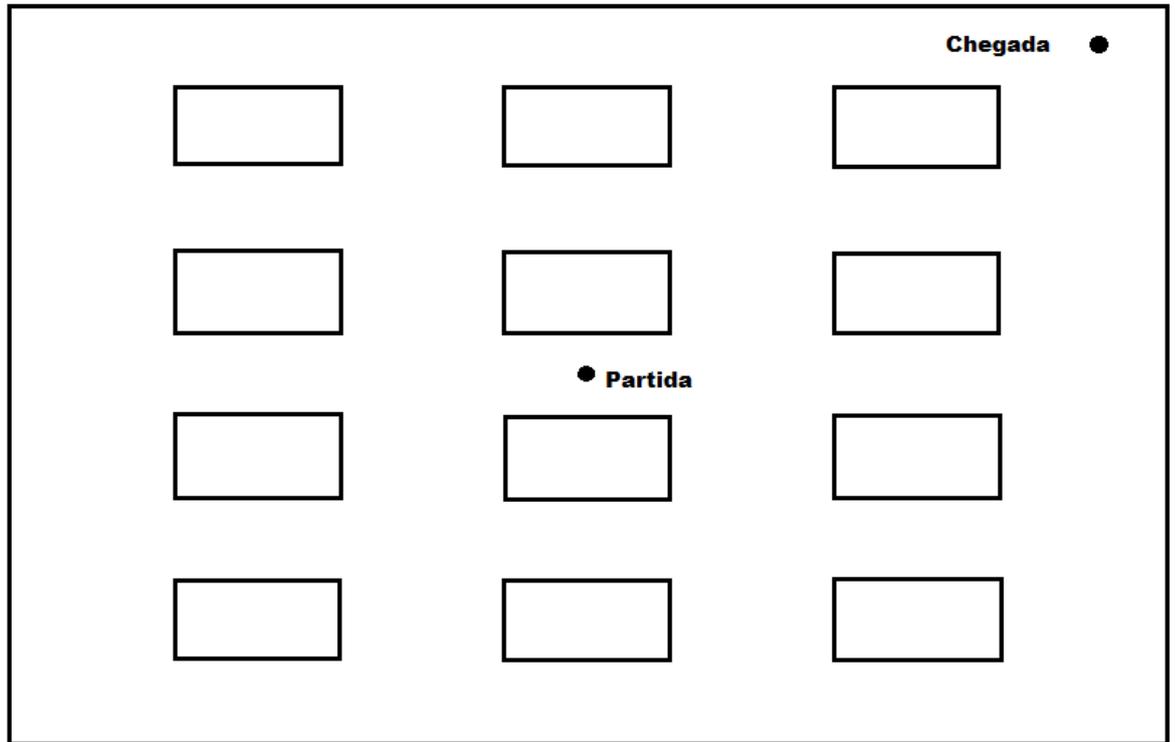


Fonte: Do autor (2019).

Foram realizados dois testes com o robô para avaliar aplicação do sistema:

- **Teste 1** – ponto inicial (288;538) e ponto final (43;931 ). Número de nós igual a 100. A Figura 36 mostra os pontos de partida e chegada do caminho.

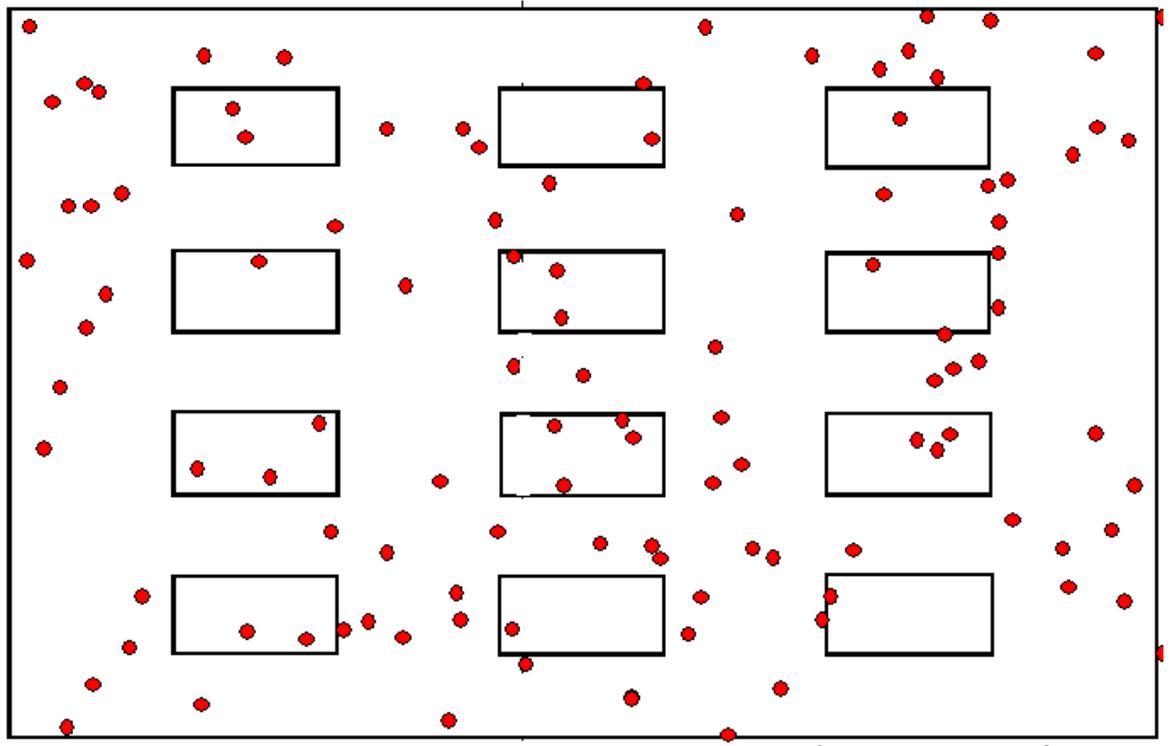
Figura 36 – Pontos de partida e chegada do robô na planta construída.



Fonte: Do autor (2019).

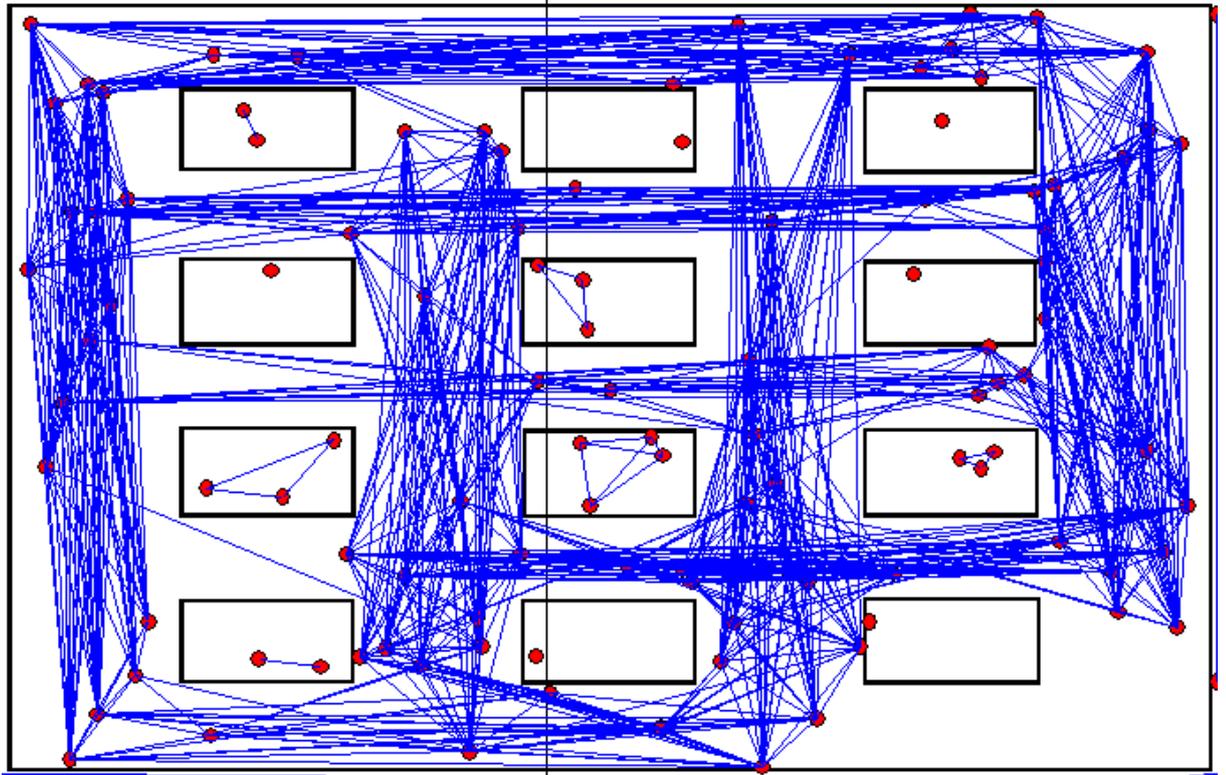
São mostradas, nas Figuras 37 e 38, respectivamente, as fases de bombardeamento, aprendido.

Figura 37 – Fase de bombardeamento do Teste 1.



Fonte: Do autor (2019).

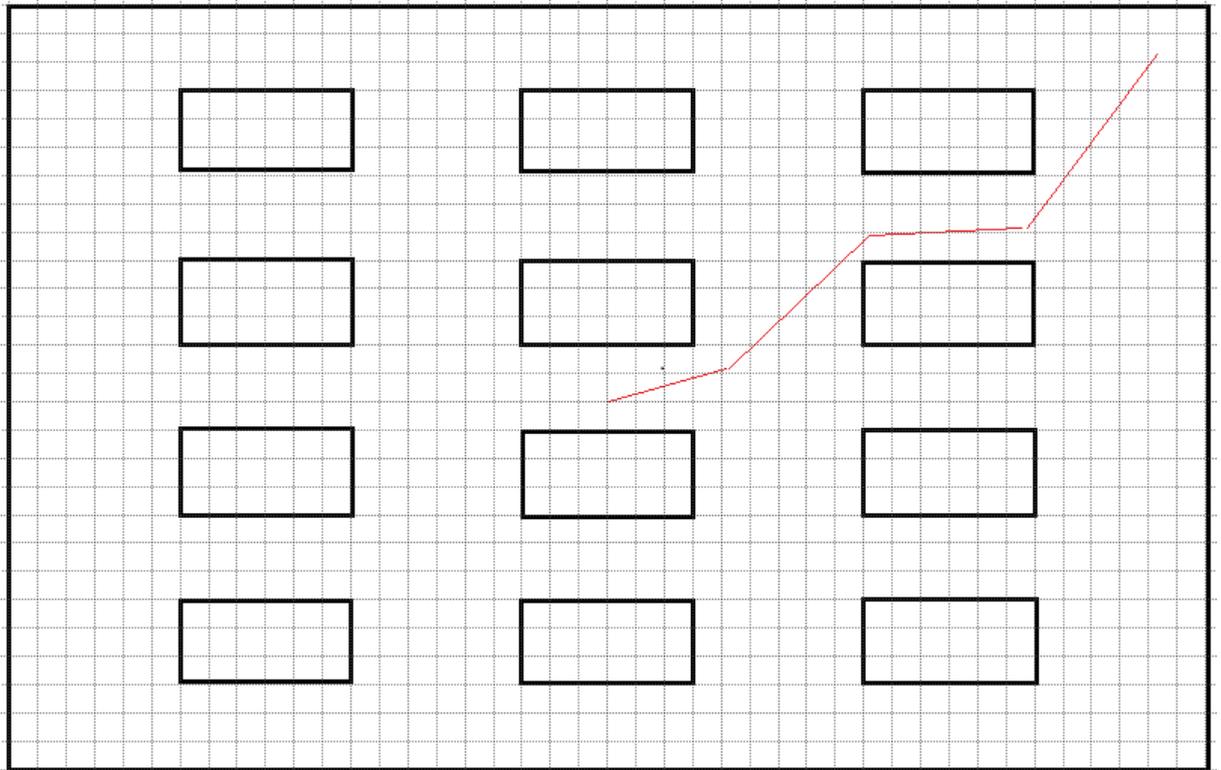
Figura 38 – Fase de aprendizado do teste 1



Fonte: Do autor (2019).

Na Figura 39 é mostrado o caminho gerado. O fundo quadriculado é um artifício empregado para se conhecer a extensão correta a ser percorrida no caminho. Cada quadrado representa 10cm da planta construída. O caminho foi dividido em 4 segmentos de reta (a,b,c e d). Cada segmento foi considerado um vetor e, a partir dele, foram derivadas suas duas componentes em x e y (por exemplo,  $a_x$  e  $a_y$ ). Aplicou-se o teorema de Pitágoras onde o segmento de reta foi considerado como sendo a hipotenusa e as componentes os catetos do triângulo retângulo formado.

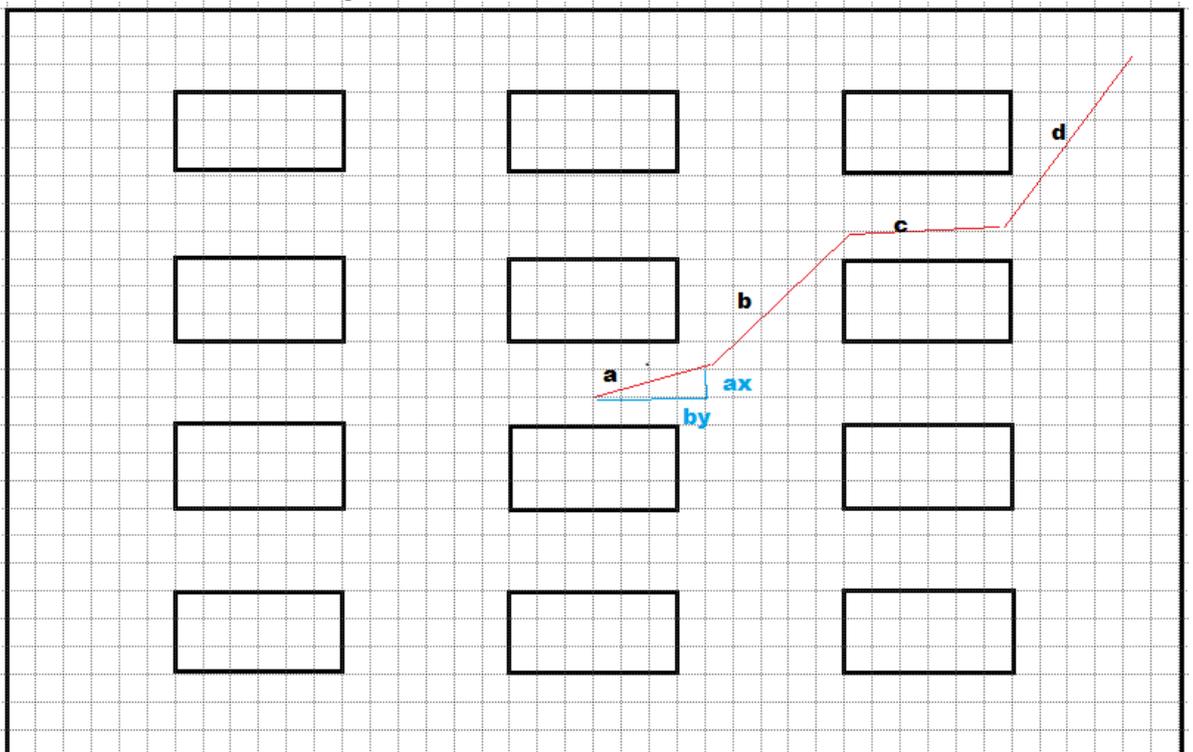
Figura 39 – Caminho gerado.



Fonte: Do autor (2019).

A Figura 40 mostra como foram feitas as divisões dos segmentos de reta e um exemplo das relações do triângulo retângulo para o segmento a.

Figura 40 – Relações de triângulo retângulo para encontrar a extensão dos segmentos de reta do caminho.



Fonte: Do autor (2019).

Depois do cálculo, utilizando o Teorema de Pitágoras, em uma função no MatLab, que recebe a medida dos catetos em centímetros, foi obtido que  $a = 41$  cm,  $b = 70,7$ cm e  $d = 72,1$  cm. O segmento  $c$  não possui componente em  $y$  e seu cálculo foi feito apenas contando os quadrados, onde  $c = 60$ cm. Os ângulos de rotação foram apenas considerados como  $90^\circ$ ,  $45^\circ$  ou  $15^\circ$ .

A partir dessas informações, foi implementado o seguinte código, como mostra a Figura 41, para o deslocamento do robô:

Figura 41 – Código implementado no Arduino para o deslocamento do robô.

Teste1 §

```

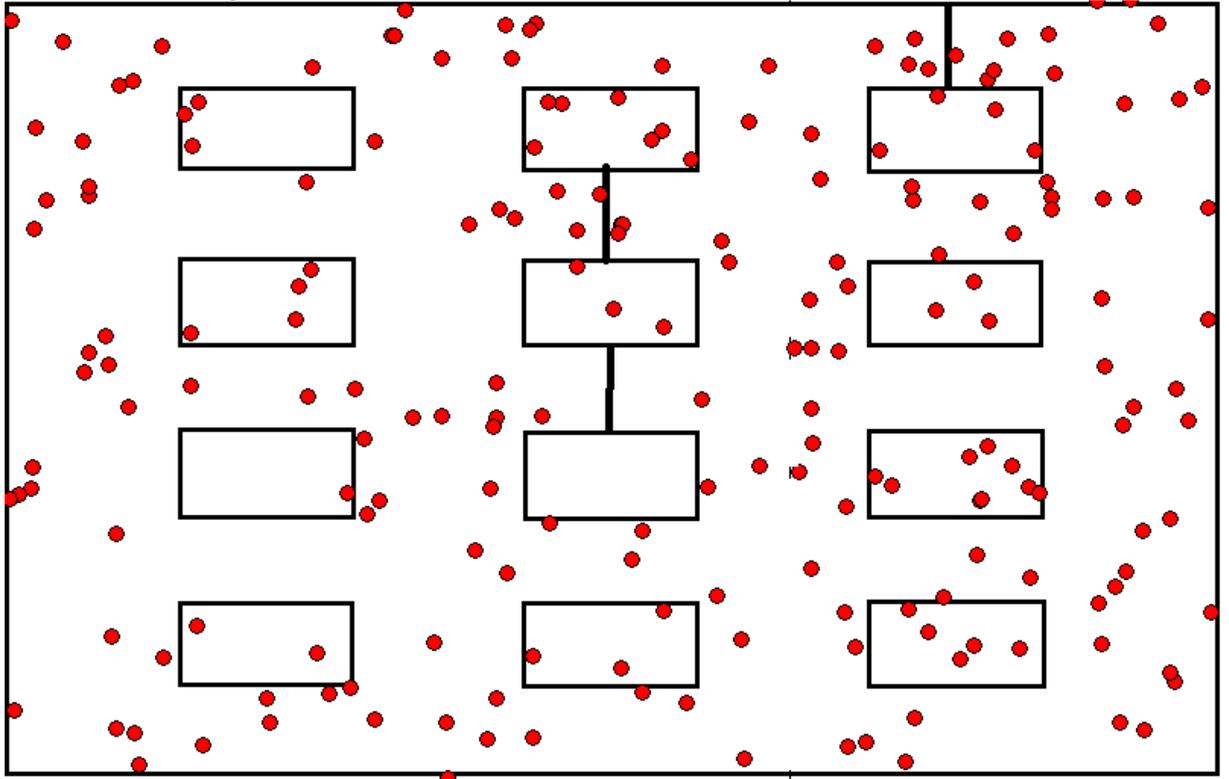
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(400); // Gira o robô 15°, para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(2985); // robô percorre os 41cm do segmento a.
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(1000); // Gira o robô 45° para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(4850); // Robô percorre os 71,7cm.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(BACKWARD); // Gira motor direito para trás.
delay(1000); // Gira o robô 45° para a direita.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(4000); // Robô percorre os 60 cm.
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(1000); // Gira o robô 45° para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(4900); // Robô percorre os 72,1cm.
motor1.run(RELEASE); // Para motor esquerdo.
motor2.run(RELEASE); // Para motor direito.
// Robô está no seu destino final

```

Fonte: Do autor (2019).

- **Teste 2** – foi usado como ponto inicial as coordenadas (49;275) e final (336;965). Um obstáculo também foi adicionado. O número de nós foi de 200. A Figura 42 mostra o “bombardeamento”.

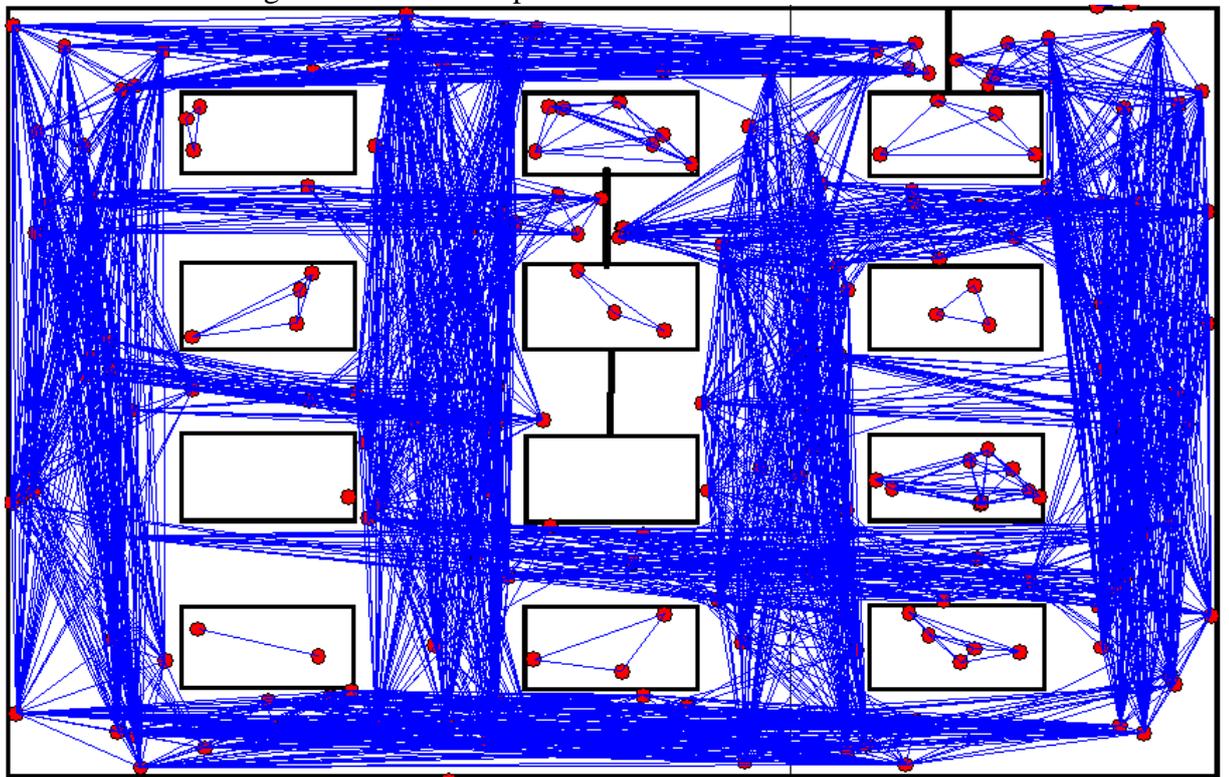
Figura 42 – Fase de bombardeamento do Teste 2.



Fonte: Do autor (2019).

Na Figura 43 é mostrada a uma foto da fase de aprendizado do Teste 2.

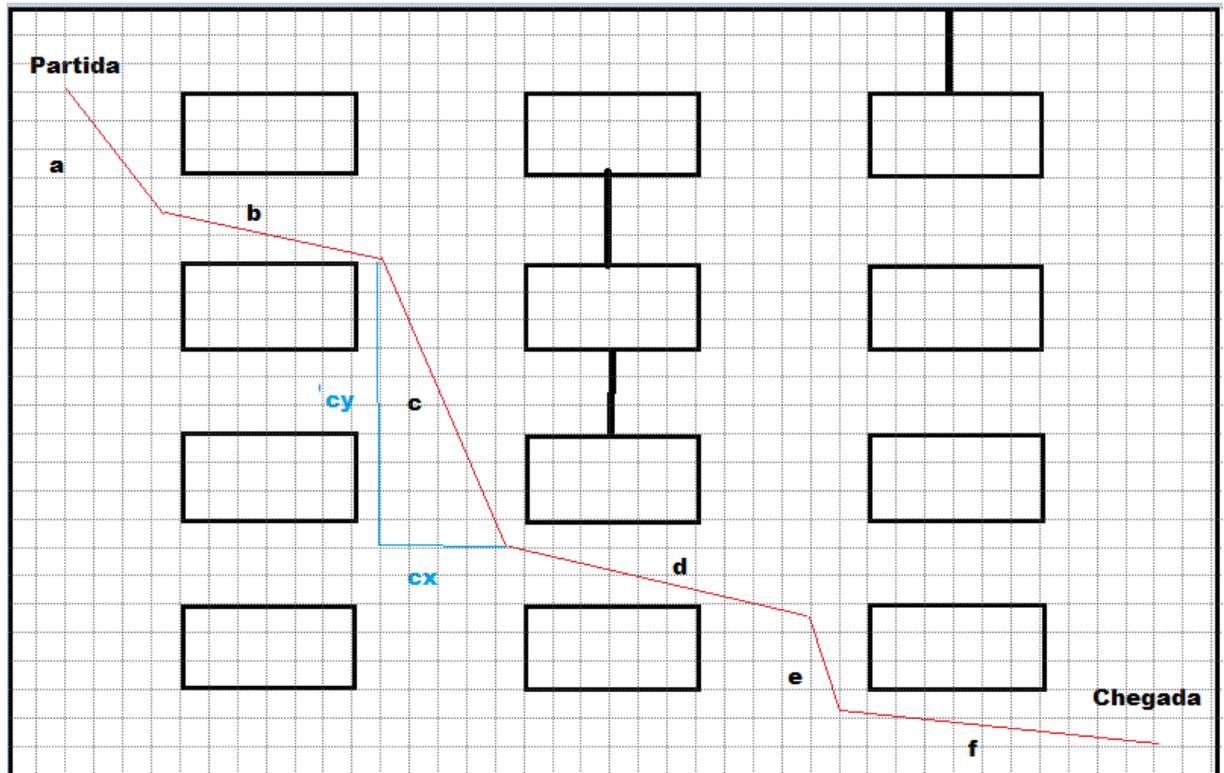
Figura 43 – Fase de aprendizado do Teste 2



Fonte: Do autor (2019).

Com o caminho gerado, é aplicada a mesma técnica do fundo quadriculado e as componentes do segmento formando triângulos retângulos. A Figura 44 mostra o caminho gerado e um exemplo de relações do triângulo retângulo para o vértice c.

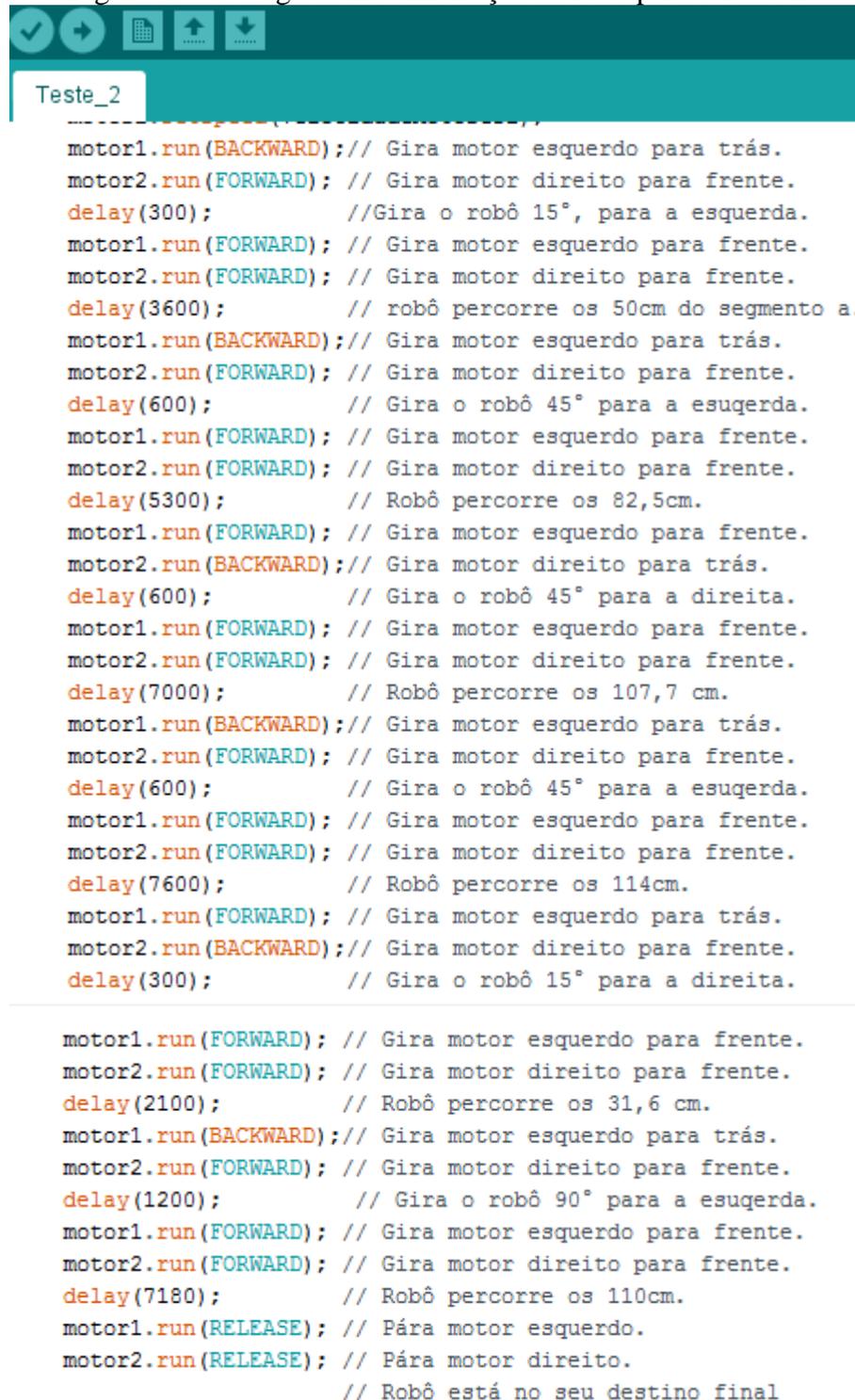
Figura 44 – Caminho gerado (vermelho), subdivisão dos segmentos (a até f) e exemplo da aplicação do Teorema de Pitágoras no segmento c.



Fonte: Do autor (2019).

O tamanho dos segmentos obtidos foi:  $a = 50\text{cm}$ ,  $b = 82,5\text{cm}$ ,  $c = 107,7\text{cm}$ ,  $d = 114$ ,  $e = 31,6\text{cm}$  e  $f = 110\text{cm}$ . Com essas informações, o código para locomover o robô foi implementado, como mostra a Figura 45.

Figura 45 – Código de movimentação do robô para o Teste 2



```

motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(300); //Gira o robô 15°, para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(3600); // robô percorre os 50cm do segmento a.
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(600); // Gira o robô 45° para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(5300); // Robô percorre os 82,5cm.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(BACKWARD); // Gira motor direito para trás.
delay(600); // Gira o robô 45° para a direita.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(7000); // Robô percorre os 107,7 cm.
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(600); // Gira o robô 45° para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(7600); // Robô percorre os 114cm.
motor1.run(FORWARD); // Gira motor esquerdo para trás.
motor2.run(BACKWARD); // Gira motor direito para frente.
delay(300); // Gira o robô 15° para a direita.

motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(2100); // Robô percorre os 31,6 cm.
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(1200); // Gira o robô 90° para a esquerda.
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(7180); // Robô percorre os 110cm.
motor1.run(RELEASE); // Para motor esquerdo.
motor2.run(RELEASE); // Para motor direito.
// Robô está no seu destino final

```

Fonte: do autor (2019).

## 5. RESULTADOS E DISCUSSÕES

Os testes de validação do algoritmo apresentaram resultados satisfatórios. Em todos os três casos de aplicação, juntamente com as variações de cada um, o algoritmo realizou a fase de bombardeamento e aprendizado corretamente, sem extrapolar os limites com obstáculos ou

de elementos da planta. A geração de caminho foi correta para os dois primeiros casos e o erro ocorrido no último já era esperado, uma vez que não havia possibilidade de geração de caminho por causa dos impedimentos posicionados na pista. A diferença entre os tempos de processamento para os casos com 200 nós e os com 400 também está conforme o funcionamento normal do algoritmo, visto que com 400 nós, as possibilidades de caminhos são maiores, o que resulta em mais combinações e processamento maior. A diferença entre os tempos da fase de aprendizado e questionamento também aconteceu em todos os casos, sendo o tempo de aprendizado maior, o que também está conforme o descrito na teoria sobre o funcionamento do MRP.

Os testes envolvendo o robô e a planta foram satisfatórios, evidenciando ser essa metodologia um bom início para o desenvolvimento de aplicações com robótica móvel em casas de vegetação. No entanto, muitas correções precisaram ser realizadas para que o robô tivesse um desempenho bom ao seguir o caminho traçado. A primeira foi a necessidade de calibração constante das velocidades das rodas com valores diferentes para cada uma, para que o robô fizesse uma linha reta. A potência entregue ao motor direito foi configurada para ser menor porque o robô estava se desviando para a direção esquerda quando acionado para andar em linha reta. Outra correção importante foi em relação à padronização das velocidades mediante os valores informados no código. Os testes mostraram que o robô andava uma faixa de 15 cm para o valor de 80% do *Duty Cycle*. No entanto, a medida que a bateria se descarregava, esse valor mudava, o que comprometia o deslocamento correto do robô na planta. A potência para o giro do robô precisou ser aumentada porque o terreno da planta tinha muito atrito. Em vista da necessidade de potências diferentes nos dois motores para o deslocamento em linha reta, sempre que era preciso realizar uma manobra de giro, os motores eram configurados novamente com a mesma potência para que o giro não ficasse desequilibrado, perdendo precisão. A Figura 46 mostra como foram implementadas todas essas mudanças no código do teste 1:

Figura 46 – Código do Teste 1 com correções.

```

Teste1_corre_es$
// inclusão de bibliotecas.
#include <AFMotor.h> // inclui biblioteca de manipulação de motores DCs.

//Definindo os pinos
AF_DCMotor motor1(1); // Define o motor1 ligado ao M1
AF_DCMotor motor2(2); // Define o motor2 ligado ao M2
int velocidadeMotores1 = 80;
int velocidadeMotores2 = 55;
int velocidadegiro = 100;
void setup(){
  Serial.begin(9600); // inicializa a comunicação serial para mostrar dados
  motor1.setSpeed(velocidadegiro); //
  motor2.setSpeed(velocidadegiro);
  motor1.run(BACKWARD); // Gira motor esquerdo para trás.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(200); //Gira o robô 15°, para a esquerda.
  motor2.setSpeed(velocidadeMotores2); //Correção de potência
  motor1.setSpeed(velocidadeMotores1); //correção de potência
  motor1.run(FORWARD); // Gira motor esquerdo para frente.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(2985); // robô percorre os 41cm do segmento a.
  motor1.setSpeed(velocidadegiro); //Correção de potência para girar
  motor2.setSpeed(velocidadegiro); //Correção de potência para girar
  motor1.run(BACKWARD); // Gira motor esquerdo para trás.
  motor2.run(FORWARD); // Gira motor direito para frente.
  delay(500); // Gira o robô 45° para a esquerda.
}

```

```

motor2.setSpeed(velocidadeMotores2);
motor1.setSpeed(velocidadeMotores1);
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(4000); // Robô percorre os 71,7cm.
motor1.setSpeed(velocidadegiro); // Correção de potência
motor2.setSpeed(velocidadegiro); // Correção de Potência
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(BACKWARD); // Gira motor direito para trás.
delay(1200); // Gira o robô 45° para a direita.
motor2.setSpeed(velocidadeMotores2);
motor1.setSpeed(velocidadeMotores1);
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(5200); // Robô percorre os 60 cm.
motor1.setSpeed(velocidadegiro);
motor2.setSpeed(velocidadegiro);
motor1.run(BACKWARD); // Gira motor esquerdo para trás.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(500); // Gira o robô 45° para a esquerda.
motor2.setSpeed(velocidadeMotores2);
motor1.setSpeed(velocidadeMotores1);
motor1.run(FORWARD); // Gira motor esquerdo para frente.
motor2.run(FORWARD); // Gira motor direito para frente.
delay(4000); // Robô percorre os 72,1cm.
motor1.run(RELEASE); // Para motor esquerdo.

motor2.run(RELEASE); // Para motor direito.
// Robô está no seu destino final

```

Fonte: Do autor (2019).

Em relação ao cálculo dos comprimentos baseado no Teorema de Pitágoras juntamente com o fundo quadriculado, os resultados foram satisfatórios. Um fator negativo do algoritmo que influenciou o trabalho, foi o fato dos caminhos serem gerados sem levar em conta as dimensões do robô, o que resultou em correções quando a rota se aproximava demais de uma estufa ou dos limites da casa de vegetação. Esse sistema não possui uma ferramenta para medir os erros em relação ao caminho que o robô percorre, com o qual ele deveria percorrer, sendo a precisão dos comandos informados crucial para um bom desempenho. Uma sugestão seria a implementação de sensores que avaliassem a trajetória, sendo com isso possível implementar uma malha fechada de controle baseada nos erros que o robô produzisse ao seguir a trajetória.

A abordagem do MRP foi útil ao planejar um caminho otimizado a partir das condições iniciais informadas. No Teste 1, com a implementação feita inteiramente nos cálculos das condições iniciais, o robô ultrapassou em dois momentos os elementos da planta, como o canto superior esquerdo do segundo canteiro da terceira coluna e o canto inferior direito do primeiro canteiro da mesma coluna. Para corrigir isso, os ângulos do segundo e terceiro giro foram

acentuados. Tal problema foi devido às dimensões do robô não consideradas pelo algoritmo no momento da geração do caminho, como mostrado pela Figura 47.

Figura 47 – No Teste 1 o robô invadiu alguns limites dos obstáculos.



Fonte: Do autor (2019).

No Teste 2 foram encontradas as mesmas dificuldades de extrapolação de barreiras, durante o trajeto dos trechos b e d, mas que foram sanados depois de algumas repetições. Todos os testes envolveram várias tentativas e erros até conseguir se chegar em um bom desempenho do robô no caminho traçado.

## 6 CONCLUSÃO

O presente trabalho procurou aplicar o algoritmo Mapas de Rotas Probabilístico juntamente com um robô móvel em um cenário representando uma casa de vegetação. O resultado foi satisfatório, visto que foi possível calcular os caminhos otimizados a partir da imagem do ambiente e programar o robô para que se deslocasse na planta sem colidir com os seus elementos. O maior desafio foi encontrar a precisão correta dos movimentos a serem desenvolvidos, uma vez que não se fez uso de qualquer sensor ou ferramentas que corrigissem eventuais erros de rota, e essas foram geradas sem levar em conta as dimensões do robô. Como sendo uma primeira abordagem para o desenvolvimento de aplicações práticas de robôs móveis em casas de vegetação em conjunto com o uso do MRP, esse projeto cumpriu o seu papel ao mostrar e padronizar uma metodologia prática para a continuidade desses estudos podendo se chegar a sistemas bem avançados de robótica para o setor agrícola na Universidade Federal de Lavras. Como discutido nos referenciais teóricos, esse tipo de tecnologia pode economizar recursos e otimizar atividades de cultivo, resultando em aumento da produtividade e maior lucro. Seus estudos estão em fase praticamente inicial, visto que aplicações de robótica móvel no setor agrícola só têm aproximadamente 30 anos. Torna-se, portanto, um campo muito promissor com demanda crescente de novas ideias e soluções para os problemas que são vastos.

Um fator muito importante a ser notado, é que este tipo de aplicação da robótica vem, principalmente, para suprir a necessidade exponencial de aumento da produção de alimentos. Esse é um problema presente no mundo atual devido, entre muitas coisas, ao crescimento demográfico exorbitante e as perdas e dificuldades nas lavouras do mundo todo causadas pelas mudanças climáticas e surgimento de novas pragas. Esses fatos vem, mais uma vez, ressaltar a importância de estudos como esse

### **6.1 Perspectivas futuras**

Para os futuros estudos que possam surgir em função desse, como sugestão, ficam essas ideias que seriam boas ferramentas para o desenvolvimento do sistema que esse trabalho propôs:

- Sensores: o uso de sensores, sejam baseados em ultrassom, imagem, velocidade ou luz, melhoraria em muito o grau de precisão e segurança dos movimentos do robô. Por exemplo, implementar rotinas para um sensor ultrassônico evitar colisões com obstáculos.
- Monitoramento da trajetória: é muito necessário algum mecanismo para monitorar os erros entre o caminho percorrido pelo robô e o que, de fato, ele deveria percorrer. A informação matemática desses erros pode resultar em estratégias de controle e avaliações melhores do desempenho. Por exemplo, uma câmera que monitore a trajetória do robô e a que ele deveria seguir, gerar um erro e executar os comandos no atuador para manobras corretivas.
- Microcontroladores com maior poder de processamento: o Arduino é uma ferramenta muito boa para prototipar projetos. No entanto, existem outros microcontroladores (ex: *Raspberry*) com poder de processamento maior, ocasionando a possibilidade de se realizar tarefas mais variadas e complexas. Como foi feito nesse projeto, todo o processamento das imagens foi realizado fora do microcontrolador do robô. A partir de um microcontrolador mais potente, pode-se desenvolver estratégias para seja feito nele mesmo.
- Precisão dos movimentos de giro do robô: aumentar a faixa de valores de giro do robô, para se realizar manobras com maior grau de concordância com o caminho gerado.

### **6.2 Perspectivas éticas**

Não se pode negar o caráter empreendedor e comercial que norteiam os estudos em robótica e em tecnologias de automação e controle. Como evidenciado no referencial teórico, o surgimento dos robôs atende à uma demanda da sociedade da época em aumentar a sua produção industrial economizando recursos de mão de obra, tempo e energias. Quanto mais um

sistema sofre mudanças, indo do trabalho baseado na mão de obra humana para os sistemas automatizados e/ou controlados, mais se economiza na produção, portanto, mais lucro é gerado. No entanto, os efeitos adversos desse “lucro aumentado” são severos para a outra ponta da história, a qual depende do trabalho que será desempenhado por máquinas e sistemas automáticos.

O presente trabalho propõe uma ideia que, devidamente estudada e desenvolvida, em um futuro próximo, pode resultar numa ferramenta com excelentes resultados para a produção agrícola. Sua implementação, muito provavelmente, iria substituir muitas pessoas em seus locais de trabalho. Isto posto, torna-se igualmente necessário e um dever ético propor uma reflexão sobre os impactos que essas tecnologias podem ter na sociedade atual, principalmente em relação à classe trabalhadora. Quando se fala em uma “tecnologia que substitui a mão de obra humana e traz lucro”, é preciso ficar atento a pergunta “lucro para quem?”. Com certeza os donos da empresa e quem realizou o projeto saem ganhando. Mas e o trabalhador que será substituído? Alguns paradoxos surgem:

- O engenheiro de Controle e Automação e outros profissionais que lidam com o desenvolvimento dessas tecnologias precisam trabalhar. Mas seu trabalho, aos moldes do que é proposto hoje, gera, na maioria das vezes, desemprego para outras profissões.

Como encontrar um ponto de equilíbrio para isso? Existe tal ponto?

- Ao se automatizar e controlar um processo gera-se muito lucro. Mas o lucro gerado é, em muitos casos, em função da substituição da mão de obra humana, que tem gastos como salário, alimentação, saúde e outros, por robôs, que não comem, não cansam e não reclamam. Portanto gera-se desemprego. Em um cenário mundial onde a tendência é que todos os mercados sigam essas ideias, será possível produzir muito, com um baixíssimo custo e elevado potencial de lucro. Mas nesse mesmo cenário, o desemprego, dentre outros problemas, estará em níveis altíssimos. Quem irá consumir esses produtos? Para quem o mercado, se ele ainda existir, oferecerá seus serviços?
- Existe a máxima hoje no mundo da automação e controle para o desenvolvimento de sistemas cada vez mais inteligentes, rápidos, com poucos erros e capacidade de correção dos mesmos (pensar). Quando realmente esses níveis de excelência começarem a ser alcançados, não estariam então perdendo o seu valor a maioria dos profissionais dessas áreas, indo rumo também à obsolescência? Mas, concomitante a isso, o desenvolvimento e progresso das tecnologias é necessário e imparável. O que podemos fazer, como engenheiros?

Embora sem tempo para tentar começar a responder esses questionamentos e muitos outros, deixá-los de lado, ou fingir que não existem, é um erro grave. Dessa forma, é posta aqui uma tentativa de estimular futuros estudantes e outros que leiam esse projeto a refletirem e, principalmente, proporem alternativas que tenham em vista também o lado humano do progresso científico e tecnológico, e que tentem trazer benefícios reais para todos os agentes envolvidos nos processos em que se dedicarem a intervir.

## 7 REFERÊNCIAS

ABRAHÃO, G. Q. S.; MEGDA, P. T.; BECKER, M. O Uso de Algoritmos A\* e Field A\* em Robôs Agrícolas. In: **Congresso Nacional de Engenharia Mecânica**. Campina Grande, 2010.

ACACCIA, G. M. et al. Mobile robots in greenhouse cultivation inspection and treatment of plants. In: **International Workshop on Advances in Service Robotics**, Itália, 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: Congresso Brasileiro de Automática, 2006, Brasília **Anais do Congresso Brasileiro de Automática. Brasília, 2006.**

ADÔRNO, B.V. **Planejamento probabilístico de rotas no espaço de configuração e sua aplicação em robótica móvel**. 2008. 124 p. Dissertação (Mestrado em Engenharia Elétrica) - Universidade de Brasília, Brasília, DF, 2008.

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**. 2015. 78 p. Dissertação (Mestrado em Computação Aplicada) – Universidade Estadual de Ponta Grossa, Ponta Grossa, PR, 2015.

BAC, C. W.; HEMMING, J.; VAN HENTEN, E. J. Stem localization of sweet-pepper plants using the support wire as a visual cue. **Computers and electronics in agriculture**, v. 105, p. 111-120, 2014.

BARTIK, Jean Jennings. **Pioneer programmer: Jean Jennings Bartik and the computer that changed the world**. Truman State University Press, 2013.

BORENSTEIN, J.; KOREN, Y. The vector field histogram-fast obstacle avoidance for mobile robots. **Robotics and Automation**, v. 7, n. 3, p. 278– 288, 1991.

BEKEY, George A. On autonomous robots. **The Knowledge Engineering Review**, v. 13, n. 2, p. 143-146, 1998.

BOYLESTAD, L. NASHELSKY. **Dispositivos Eletrônicos e Teoria de Circuitos**. 1996.

BRITO, Glauca da Silva; DA PURIFICAÇÃO, Ivonélia. **Educação e novas tecnologias**. Editora Ibpex, 2008.

CANNY J., DONALD B. Simplified Voronoi diagrams. **Discrete & Computational Geometry**, v. 3, p. 219-236, 1988.

CAPEK, Karel. **RUR (Rossum's universal robots)**. Penguin, 2004

CHOSSET, H. et al. **Principles of Robot Motion: Theory, Algorithms and Implementations**, MIT Press, 2005.

DE SOUSA PISSARDINI, Rodrigo; WEI, Daniel Chin Min; DA FONSECA JÚNIOR, Edvaldo Simões. Veículos Autônomos: conceitos, histórico e estado-da-arte. In: **Anais do XXVII Congresso de Pesquisa e Ensino em Transportes–ANPET**. 2013.

DEVOL, George C. **Magnetic process control**. U.S. Patent n. 2,590,091, 25 mar. 1952.

ELFES, A. Using occupancy grids for mobile robot perception and navigation. **Computer**, v. 22, p. .46-57, 1989.

GASPARETO, M. J. **Seleção de atributos em imagens coletadas sob condições de iluminação não controlada e sua influência no desempenho de classificadores *NAIVE BAYES* para identificação de objetos em estufas agrícolas.** 2013. Dissertação (Mestrado em Computação Aplicada) – Universidade Federal de Ponta Grossa, 2013.

GASPARETTO, A. et al. Path Planning and Trajectory Planning Algorithms: A General Overview. **Research Gate**, v. 29, p. 3-27, 2015.

GOLÇALVES, M. A. **Algoritmo A-Estrela de estado híbrido aplicado à navegação autônoma de veículos.** 2013. 64 p. Dissertação (Mestre em Informática) – Universidade Federal do Espírito Santo, Vitória, ES, 2013.

GONZALEZ, R. et al. Navigation techniques for mobile robots in greenhouses. **Applied Engineering in Agriculture**, v. 25, p. 153-165, 2009.

HACHOUR, O. The proposed genetic FPGA implementation for path planning of autonomous mobile robot. **Circuits, Systems, and Signal Processing**, v. 2, p. 151-167, 2008.

HARTENBERG, Richard S.; DENAVIT, Jacques. A kinematic notation for lower pair mechanisms based on matrices. **Journal of applied mechanics**, v. 77, n. 2, p. 215-221, 1955.

HOCKSTEIN, Neil G. et al. A history of robots: from science fiction to surgical robotics. **Journal of robotic surgery**, v. 1, n. 2, p. 113-118, 2007.

JUNIOR, A. M. **O uso do algoritmo Mapas de Rotas Probabilístico na Robótica Móvel: planejamento de caminhos para ambientes estáticos com foco em casas de vegetação.** 2018. 53 p. Monografia (Graduação em Engenharia de Controle e Automação) – Universidade Federal de Lavras, Lavras, MG, 2018.

KAVRAKI, L. E. et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. **IEEE Transactions on Robotics and Automation**, v. 12, n. 4, p. 566-580, 1996.

KAVRAKI, L. E.; KOLOU NTZAKIS, M. N.; LATOMBE, J. C. Analysis of probabilistic roadmaps for path planning. **IEEE Transactions on Robotics and Automation**, v. 14, p. 166–171, 1998.

KAVRAKI, L.; LATOMBE, J. C. Randomized preprocessing of configuration space for fast path planning. In: **IEEE International Conference on Robotics and Automation**, p. 2138-2145, San Diego, 1994.

KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. **The International Journal of Robotics Research**, v. 5, p. 396-404, 1985.

KO, Min Hyuc et al. Autonomous greenhouse mobile robot driving strategies from system integration perspective: Review and application. **IEEE/ASME Transactions On Mechatronics**, v. 20, n. 4, p. 1705-1716, 2015.

KUIPERS, B; BYUN, Y. T. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. **Journal of Robotics and Autonomous Systems**, v.8, p. 47–63, 1991.

LATOMBE, J. C. **Robot Motion Planning.** Boston: Kluwer Academic Publishers, 1990.

LAVALLE, S. M. **Planning Algorithms.** Cambridge University Press., 2006.

LENTEREN, J. C.; WOETS, J. Biological and integrated pest control in greenhouses. **Annual Reviews**, v. 33, p. 239-269, 1988.

LOZANO-PÉREZ, T.; WESLEY, M. A. Na algorithm for planning collision-free paths among polyhedral obstacle. **Communications of the ACM**, v. 22, p. 560-570, 1979.

MAHADEVAN, B.; NARENDRAN, T. T. Design of an automated guided vehicle based material handling system for a flexible manufacturing system. *International Journal of Production Research*, v. 28, n.º 9, p. 1611-1622, 1990.

MANDOW, A. et al. The autonomous mobile robot AURORA for greenhouse operation. **IEEE Robotics & Automation Magazine**, v. 3, p.18-28, 1996.

MARTIN, M. C.; MORAVEC, H. P. **Robot evidence grids**, 1996.

MARTINS, A. D M. et al. Comparação entre tempos de localização de objetos em imagens. In: **VI Simpósio Brasileiro de Automação Inteligente, 2003**.

MASEHIAN, E.; KATEBI, Y. Robot motion planning in dynamic environments with moving obstacles and target. **International Journal of Mechanical Systems Engineering**, v. 1, p. 20-25, 2007.

MATARIC, M. J. **A distributed model for mobile robot environment-learning and navigation**. 1990. 129p. Dissertação (Mestre em Engenharia Elétrica e Ciências da Computação) - Cambridge, MA, 1990.

MEHTA, S. S.; BURKS, T.F.; DIXON, W. E. Vision-based localization of a wheeled mobile robot for greenhouse applications: A daisy-chaining approach. **Computers and electronics in agriculture**, v. 63, p. 28-37, 2008.

MORAVEC, H.; ELFES, A. High resolution maps from wide angle sonar. **In: Robotics and Automation. Proceedings**. IEEE International Conference, v. 2, p. 116–121, 1985.

MORAVEC, H. Sensor fusion in certainty grids for mobile robots. **In: Sensor Devices and Systems for Robotics. Springer Berlin Heidelberg**, v. 52, p. 253–276, 1989.

MUNSON, George E. THE RISE AND FALL OF UNIMATION INC.-A story of robotics innovation & triumph that changed the world. **Robot-Congers**, n. 24, p. 36, 2010.

NEZHAD, Mohammad Ali Kieh Badroudi; MASSH, Jafar; KOMLEH, Hossein Ebrahimpour. Tomato Picking Machine Vision Using with the Open CV's Library. In: **2011 7th Iranian Conference on Machine Vision and Image Processing**. IEEE, 2011. p. 1-5.

NIKU, Saeed Benjamin. **Introdução a robótica**. Grupo Gen-LTC, 2013.

NILSSON, Nils J. **Shakey the robot**. SRI INTERNATIONAL MENLO PARK CA, 1984.

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

PEDROSA, Diogo; MEDEIROS, Adelardo AD; ALSINA, Pablo J. Um método de geração de trajetória para robôs não-holonômicos com acionamento diferencial. **Simpósio Brasileiro de Automação Inteligente**, p. 840-845, 2003.

PIO, J. L. S.; CASTRO, T. H. C.; CASTRO JÚNIOR, A. N. A Robótica Móvel como Instrumento de Apoio à Aprendizagem de Computação. In: **XVII Simpósio Brasileiro de Informática na Educação – SBIE**. Departamento de Ciência da Educação – Universidade Federal do Amazonas (Manaus), 2006, p. 497-506.

QINGCHUN, Feng et al. **Study on strawberry robotic harvesting system**. In: 2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE). IEEE, 2012. p. 320-324.

RAJA, P.; PUGAZHENTHI, S. Optimal path planning of mobile robots: A review. *International Journal of Physical Sciences*, v. 7, p. 1314-1320, 2012.

ROBOTIC INDUSTRIES ASSOCIATION. A tribute to Joseph Engelberger: Unimate, the first industrial robot. Disponível em: <<http://www.robotics.org/joseph-engelberger/unimate.cfm>>. Acesso em: 23 jan. 2016.

ROCHA, Rui Paulo. Estado da Arte da Robótica móvel em Portugal. **Instituto de Sistemas e Robótica, Coimbra, Portugal**. URL: <http://www.dee.uc.pt/~rprocha>, 2000.

RUSSEL, Stuart; NORVIG, Peter. Inteligência Artificial. 3a Edição. **Editora Campus**, 2013.

SALES, D; OSÓRIO, F.; WOLF, D. **Topological autonomous navigation for mobile robots in indoor environments using ann and fsm**. In: Proceedings of the I Brazilian Conference on Critical Embedded Systems (CBSEC), São Carlos, Brazil, 2011.

SAMMONS, P. J.; FURUKAWA, T.; BULGIN, A. Autonomous pesticide spraying robot for use in a greenhouse. In: **Australian Conference on Robotics and Automation**, Austrália, 2005.

SANCHEZ-HERMOSILLA, J. A mechatronic description of autonomous mobile robot for agricultural tasks in greenhouses. **Mobile Robots Navigation**, p. 583-608, 2010.

SANTANA, A. M. **Localização e planejamento de caminhos para um robô humanoide e um robô escravo com rodas**. 2007. 63 p. Dissertação (Mestrado em Ciências) – Universidade Federal do Rio Grande do Norte, Natal, RN, 2007.

SECCHI, H. A. **Una Introducción a los Robots Móviles**. 2008. 81 p. Monografía - Instituto de Automática – INAUT Universidade Nacional de San Juan – UNSJ – Argentina, 2008.

SIEGWART, R; NOURBAKHSI, I., R. **Introduction to Autonomous Mobile Robots**. Inglaterra, 2004.

SILVEIRA, Y. S. **Um novo método de planejamento de caminho para robôs baseado em espuma probabilística**, 2016. 51 p. Dissertação (Mestrado em Engenharia Mecatrônica) – Universidade Federal do Rio Grande do Norte, Natal, RN, 2016.

SUPREM, Abhijit; MAHALIK, Nitaigour; KIM, Kiseon. A review on application of technology systems, standards and interfaces for agriculture and food sector. **Computer Standards & Interfaces**, v. 35, n. 4, p. 355-364, 2013.

TAKAHASHI, O.; SCHILLING, R. J. Motion planning in a plane using generalized Voronoi diagrams. **IEEE Transactions on Robotics and Automation**, v. 5, p. 143-150, 1989.

THRUN, S. **Robotic Mapping: A Survey**. Pittsburgh: M. Kaufmann, 28 p., 2002.

UNIVERSIDADE FEDERAL DE LAVRAS. Biblioteca Universitária. Manual de normalização e estrutura de trabalhos acadêmicos: TCCs, monografias, dissertações e teses. 2. ed. rev., atual. e ampl.

Lavras, 2016. Disponível em: <<http://repositorio.ufla.br/jspui/handle/1/11017>>. Acesso em: data de acesso.

VICTORINO, A. C.; RIVES, P. SLAM with consistent mapping in a hybrid model. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, Beijing. *Proceedings...* (CD ROM).

VOLPE, R. A., KHOSLA, P. K. Manipulator control with superquadric artificial potential functions: theory and experiments. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 20, p. 1423-1436, 1990.

VOLPE, R. A. **Real and artificial forces in the control of manipulators: theory and experiments**. 1990. 162 p. Tese (Doutorado em Filosofia) - Carnegie Mellon University, Pittsburgh, Pennsylvania, 1990.

VOTH, Danna. A new generation of military robots. **IEEE Intelligent Systems**, v. 19, n. 4, p. 2-3, 2004.

WILLMARTH, S. A.; AMATO, N. M.; STILLER, P. F. MAPRM: A probabilistic roadmap planner with sampling on the medial axis. In: **IEEE International Conference on Robotics & Automation**. Detroit, Michigan, 1999.

WOLF, D. F. et al. **Robótica inteligente: Da Simulação às Aplicações no Mundo Real.[Tutorial]** In: André Ponce de Leon F. de Carvalho; Tomasz Kowaltowski (Org). JAI: Jornada de Atualização em Informática da SBC. Rio de Janeiro, 2009.

YANG, Ming-Ta; CHEN, Cheng-Chuan; KUO, Yen-Long. Implementation of intelligent air conditioner for fine agriculture. **Energy and Buildings**, v. 60, p. 364-371, 2013.