



MATRICKS DE OLIVEIRA ROCHA

**DESENVOLVIMENTO DE UMA PLATAFORMA MÓVEL
PARA GERAÇÃO DE TRAJETÓRIAS EM UM AMBIENTE
AGRÍCOLA CONTROLADO**

LAVRAS - MG

2020

MATRICKS DE OLIVEIRA ROCHA

**DESENVOLVIMENTO DE UMA PLATAFORMA MÓVEL PARA GERAÇÃO DE
TRAJETÓRIAS EM UM AMBIENTE AGRÍCOLA CONTROLADO**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras como parte das
exigências do Curso de Engenharia de Controle
e Automação, para a obtenção do título de
Bacharel.

Prof. Dr. Leonardo Silveira Paiva
Orientador

LAVRAS - MG
2020

MATRICKS DE OLIVEIRA ROCHA

**DESENVOLVIMENTO DE UMA PLATAFORMA MÓVEL PARA GERAÇÃO DE
TRAJETÓRIAS EM UM AMBIENTE AGRÍCOLA CONTROLADO**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal de Lavras como parte das
exigências do Curso de Engenharia de Controle
e Automação, para a obtenção do título de
Bacharel.

APROVADO em 30 de Junho de 2020.

Prof. Dr. Leonardo Silveira Paiva, UFLA

Prof. Dr. Adriano Valentim Diotto, UFLA

Prof. Dr. Carlos Renato Borges dos Santos, IFMG

Prof. Dr. Leonardo Silveira Paiva

Orientador

LAVRAS - MG

2020

AGRADECIMENTOS

A Deus por sempre me proteger.

A meus pais por me apoiarem, incondicionalmente, desde o início, apesar da saudade diária.

Aos amigos de longa data, especialmente Giovane Couto de Souza e Bruna de Freitas Martins por todo apoio, carinho e cumplicidade de sempre.

À Universidade Federal de Lavras, em especial ao Departamento de Engenharia e ao Departamento de Automática e a todos os professores e técnicos que me ensinaram, instruíram e me guiaram, contribuindo para a formação do profissional que me torno hoje.

Ao meu orientador Dr. Leonardo Silveira Paiva, pela orientação, paciência, confiança e disponibilidade em me ajudar em mais essa etapa.

A Beatriz Bonucci pela amizade, paciência e disponibilidade em me ajudar na realização deste trabalho.

Aos meus amigos que conheci em Lavras, que cultivei ao longo dos últimos cinco anos, dentro e fora da universidade, com os quais compartilhei incontáveis experiências e que me acolheram da melhor maneira possível.

A todos que direta ou indiretamente, contribuíram em minha jornada durante a graduação.

Muito obrigado!

RESUMO

Com o aumento demográfico mundial nas últimas décadas, a demanda de alimentos cresce cada vez mais. A automação na agricultura tornou-se uma ferramenta extremamente importante para atender tal demanda, otimizando os processos agrícolas e aumentando a produtividade de alimentos. Os robôs móveis podem ser utilizados na implementação de sistemas automatizados em casas de vegetação, se locomovendo no ambiente de acordo com a necessidade, ditada por sensores instalados no mesmo. Este trabalho foi realizado com as finalidades de: desenvolver uma plataforma móvel autônoma e; gerar trajetórias em um ambiente de cultura genérico de acordo com dados obtidos de sensores de umidade instalados no mesmo. Inicialmente, construiu-se uma plataforma móvel e foram realizados alguns testes para se obter dados relevantes de velocidade da mesma. Posteriormente, foi projetado e construído um ambiente de cultura genérico e, foram instalados nove sensores de umidade no mesmo. Estes sensores são responsáveis por realizar leituras de umidade em pontos estratégicos dentro do ambiente, possibilitando a determinação de alguns parâmetros de umidade que definem se é necessário que o robô se desloque até um determinado local ou não. Quando necessário que o robô se desloque até o sensor, a trajetória é gerada pelo algoritmo Mapa de Rotas Probabilístico (MRP) e as informações da trajetória são enviadas ao robô por meio de comunicação Bluetooth. Após o recebimento, o robô interpreta essas informações e se desloca até o local desejado.

Palavras-chave: Robótica Móvel, Mapa de Rotas Probabilístico, Comunicação V2I, Planejamento de Trajetórias, MATLAB.

ABSTRACT

The significant worldwide demographic increase in the last decades, resulted in a constant growing of the demand for food. The agricultural automation has become an extreme important tool to attend this demand, optimizing agricultural processes and increasing food productivity. Mobile robots are fundamental pieces for the implementation of an automated system in greenhouses, being able to displace in the environment according to the need, dictated by sensors installed in it. This work was carried out in order to develop an autonomous mobile platform and generating trajectories in a generic culture environment according to data obtained from humidity sensors. Initially, a mobile platform was built and some tests were made to obtain relevant speed data about it. Subsequently, a generic culture environment has been designed and built, and nine humidity sensors were installed in it. These sensors are responsible for collecting humidity data at strategic points within the environment, making possible the determination of some humidity parameters that define if it is necessary for the robot to displace to a certain location or not. When the robot needs to go to the sensor, the trajectory is generated by the Probabilistic Road Map (PRM) algorithm and the trajectory information are sent to the robot through Bluetooth communication. After receiving the data, the robot interprets this information and displaces to the desired location.

Keywords: Mobile Robotic, Probabilistic Road Map, V2I Communication, Trajectory Planning, MATLAB.

SUMÁRIO

PRIMEIRA PARTE	8
INTRODUÇÃO GERAL	8
1 INTRODUÇÃO.....	8
2 REFENCIAL TEÓRICO.....	9
2.1 Robótica.....	9
2.2 Robótica Móvel	10
2.3 Mapa de Rotas Probabilístico (MRP).....	11
2.4 Comunicação V2I (Veículo para Infraestrutura)	12
3 CONSIDERAÇÕES FINAIS	13
4 REFERÊNCIAS BIBLIOGRÁFICAS	13
SEGUNDA PARTE.....	15
ARTIGO.....	15

PRIMEIRA PARTE

INTRODUÇÃO GERAL

1 INTRODUÇÃO

O objetivo geral deste Trabalho de Conclusão de Curso foi o de construir uma plataforma móvel capaz de locomover-se de forma autônoma em um cenário que representa uma casa de vegetação e; gerar trajetórias de acordo com dados obtidos de sensores de umidade, além de iniciar uma série de estudos futuros que, juntos, serão capazes de automatizar processos agrícolas de pequeno porte.

A finalidade do trabalho foi a de utilizar técnicas de robótica móvel, onde o algoritmo Mapa de Rotas Probabilístico (MRP) foi o responsável por gerar as trajetórias do robô, de acordo com os dados obtidos através dos sensores de umidade, presentes no cenário. Espera-se que este projeto esclareça a respeito dos benefícios de se utilizar uma plataforma móvel autônoma em um processo agrícola e sirva de material de apoio para continuidade do projeto.

Este trabalho foi desenvolvido no formato de artigo científico, sendo dividido em duas partes:

A primeira parte é composta por:

- 1) Uma Introdução, contendo os objetivos do trabalho e sua organização;
- 2) Um Referencial Teórico, contendo uma contextualização dos assuntos abordados no artigo. Neste momento serão apenas citadas as partes mais importantes referentes ao trabalho, pois o mesmo se concentra na produção do artigo, onde está contida uma explicação mais detalhada a respeito dos conceitos apresentados nesta introdução geral;
- 3) E Considerações Finais, contendo uma breve análise dos resultados obtidos com a realização deste trabalho.

A segunda parte é composta por um artigo científico descrevendo e analisando todo o processo do trabalho. O artigo será publicado em um periódico a ser definido pelo autor e pelo orientador.

2 REFERENCIAL TEÓRICO

2.1 Robótica

Define-se robótica, como a ligação inteligente entre a percepção e a ação. Trabalhar com robótica significa estudar, projetar e implementar sistemas ou dispositivos que, sejam úteis na realização de uma determinada tarefa que envolva interação física entre o sistema (ou dispositivo) e o meio onde a tarefa está sendo realizada (PIO; CASTRO; JÚNIOR, 2006). Seu estudo engloba áreas de mecânica, eletrônica e computação, teorias de controle e inteligência artificial (PIMENTA, 2009).

Atualmente, todos que vivem no mundo desenvolvido se beneficiam dos avanços da robótica em seu cotidiano. Os setores de manufatura, serviços e assistência médica incorporaram a robótica para aumentar a eficiência e precisão de seus processos. Os robôs ajudam a construir nossas máquinas, embalar nossos alimentos, e lavar nossos carros (HOCKSTEIN et al., 2007).

Na Figura 1, é apresentado o Sistema Cirúrgico daVinci, que permite a realização de uma cirurgia de forma remota.

Figura 1 – Sistema Cirúrgico daVinci



Fonte: Hockstein et al. (2007)

Os robôs devem interagir com o mundo físico respeitando as Três Leis de Asimov¹:

1° Lei: Um robô não pode ferir um ser humano ou, por omissão, permitir que um ser humano sofra algum mal;

2° Lei: Um robô deve obedecer às ordens que lhe sejam dadas por seres humanos, exceto nos casos em que tais ordens contrariem a Primeira Lei;

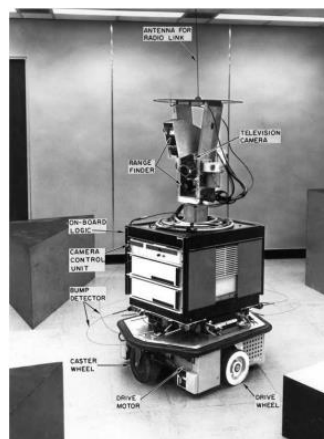
3° Lei: Um robô deve proteger sua própria existência desde que tal proteção não entre em conflito com as Primeira e Segunda Leis.

2.2 Robótica Móvel

Um robô móvel é um agente capaz de extrair informações do ambiente e utilizar esse conhecimento do mundo para deslocar-se com segurança no ambiente, atuando e executando tarefas. Os robôs móveis são muito mais versáteis, pois não precisam estar fixados em seu local de trabalho como os robôs industriais convencionais, movimentando-se por meio de pernas, rodas, esteiras, por empuxo ou flutuação (PIO; CASTRO; JÚNIOR, 2006).

Atualmente, existem robôs móveis com as capacidades de locomoção e de operação de modo semi ou completamente autônomo. Estes são chamados de Robôs Móveis Autônomos (WOLF et al., 2009). Na Figura 2, é apresentado o primeiro robô móvel construído.

Figura 2 – *Shakey*, o primeiro robô móvel



Fonte: Moravec (1999)

¹ Escritor de Ficção Científica Issac Asimov (1920-1992), elaborada em seu livro de ficção *I, Robot* (“*Eu, Robô*”) o comportamento dos robôs (Leis da Robótica).

Na Figura 3, é apresentado um robô móvel militar desenvolvido pela *iRobot*.

Figura 3 – *Warrior X700*, robô militar da *iRobot*



Fonte: Secchi (2008)

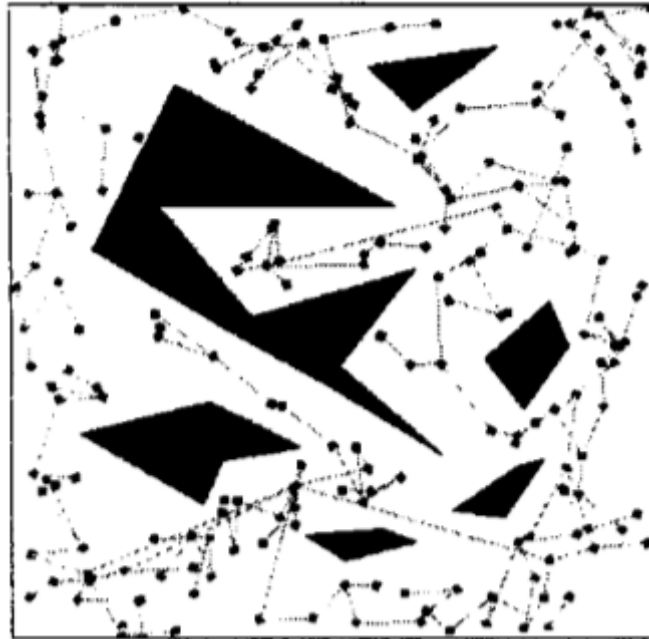
2.3 Mapa de Rotas Probabilístico (MRP)

Durante as últimas duas décadas o problema do planejamento de movimento tem sido estudado incansavelmente. Diferentes métodos foram propostos, incluindo técnicas de campo potencial, mapa de rotas, decomposição celular, redes neurais e algoritmos genéticos (BOOR; OVERMARS; VAN DER STAPPEN, 1999).

Os Mapas de Rotas Probabilísticos (MRPs) constroem uma rede de caminhos simples, conectando configurações sem colisão, escolhidas aleatoriamente. Eles têm sido bem sucedidos na resolução de problemas difíceis de planejamento de trajetórias, mas sua eficiência é baixa quando o espaço livre contém passagens estreitas (HSU; KRAVAKI, 2013).

A ideia geral do MRP é pegar uma coleção de configurações aleatórias no espaço livre de obstáculos. Essas configurações livres formam os nós de um grafo. Alguns pares de nós são escolhidos e um planejador de movimentos é usado para tentar conectar esses nós e formar um caminho livre de obstáculos. Quando o planejador tem sucesso, uma aresta é adicionada ao grafo (BOOR; OVERMARS; VAN DER STAPPEN, 1999). A Figura 4 contém um grafo gerado pelo algoritmo.

Figura 4 – Um grafo gerado pelo algoritmo MRP



Fonte: Boor et al. (1999)

2.4 Comunicação V2I (Veículo para Infraestrutura)

As tecnologias mais promissoras de comunicação sem fio para sistemas de transporte conectados são Veículo para Veículo (V2V) e Veículo para Infraestrutura (V2I, ou combinadas como V2X) (GHEORGHIU et al., 2017).

Segundo Gheorghiu et al. (2017), as tecnologias mais testadas e apropriadas para comunicação V2X são Wi-Fi, Bluetooth e ZigBee. Devido ao seu baixo consumo de energia e baixo alcance, a tecnologia Bluetooth (BT) é conveniente no desenvolvimento de redes de dispositivos de pequena extensão e que precisam trocar informações.

Neste trabalho, apenas utilizou-se a comunicação entre o ambiente e o robô móvel, ou seja, comunicação V2I. Além disso, o cenário criado foi pequeno o bastante para se justificar o uso da tecnologia Bluetooth.

3 CONSIDERAÇÕES FINAIS

A realização deste trabalho comprovou a veracidade dos conceitos de robótica para a construção e programação de robôs móveis, e a eficácia do algoritmo de geração de trajetórias MRP em gerar rotas livres de obstáculos, para que o robô se desloque com segurança no ambiente. A tecnologia Bluetooth se mostrou muito eficiente e robusta para implementação da comunicação V2I em um ambiente de baixa extensão. Por fim, foi possível demonstrar o quanto a eficiência de um processo agrícola pode aumentar, automatizando-o através do uso de robôs móveis e sensores instalados no ambiente.

4 REFERÊNCIAS BIBLIOGRÁFICAS

ASIMOV, I., Guest commentary: **The Three Laws**, *Compute! Magazine*, Vol 18 No 3, pp 18, November 1981

ASIMOV, I., Runaround, *Astounding Science Fiction*, 1942. Reprinted in *I, Robot*, Gnome Press, 1950.

BOOR V.; OVERMARS, M. H.; VAN DER STAPPEN, A. F. "**The Gaussian sampling strategy for probabilistic roadmap planners**" Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C), Detroit, MI, USA, 1999, pp. 1018-1023 vol.2, doi: 10.1109/ROBOT.1999.772447.

GHEORGHIU, R. A. et al. (2017). **Bluetooth Latency Analysis for Vehicular Communications in a Wi-Fi Noisy Environment**. 10.1109/TSP.2017.8075956.

HOCKSTEIN, N. G. *et al.* **A history of robots: from science fiction to surgical robotics**. *J Robotic Surg* **1**, 113–118 (2007).

HSU, D.; KRAVAKI, L. (2002). **On Finding Narrow Passages with Probabilistic Roadmap Planners**.

MORAVEC, H. **Robot: Mere machine to transcendent mind**. [S.l.]: Oxford University Press, (1999).

PIMENTA, T. T. **CONTROLE DE MANIPULADORES ROBÓTICOS**. 2009. Dissertação (Trabalho de Conclusão de Curso) – PUC. Rio de Janeiro.

PIO, J. L. S.; CASTRO, T. H. C.; JÚNIOR, A. N. C. **A Robótica Móvel como Instrumento de Apoio à Aprendizagem de Computação**. Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE), [S.l.], p. 497-506, nov. 2006. ISSN 2316-6533. Disponível em: <<https://br-ie.org/pub/index.php/sbie/article/view/510/496>>. Acesso em: 04 jun. 2020.

SECCHI, H. **Uma Introdução aos Robôs Móveis**. 2008. 1 v. Tese (Doutorado), Ingeniería de Sistemas de Control, Universidade Nacional de San Juan, San Juan (Argentina), 2008.

WOLF D. F. et al. “**Robótica Móvel Inteligente: Da Simulação às Aplicações no Mundo Real**,” Atualizações em informática 2009 JAI SBC, pp. 1-51, 2009.

SEGUNDA PARTE
ARTIGO

**ARTIGO – DESENVOLVIMENTO DE UMA PLATAFORMA MÓVEL PARA
GERAÇÃO DE TRAJETÓRIAS EM UM AMBIENTE AGRÍCOLA CONTROLADO**

Desenvolvimento de uma Plataforma Móvel para Geração de Trajetórias em um Ambiente Agrícola Controlado

Matricks de Oliveira Rocha*

*matricks.rocha@estudante.ufla.br

Universidade Federal de Lavras - Campus Universitário, CEP 37200-000 • Lavras/MG

Resumo: Nas últimas décadas, a demanda de alimentos vem crescendo cada vez mais, devido ao grande aumento demográfico mundial. Com isso, fica clara a necessidade do aumento da produtividade de alimentos e da otimização dos processos agrícolas. Os robôs móveis são uma ótima ferramenta para otimizar esses processos e automatizá-los e, por isso, estão cada vez mais presentes na agricultura. Neste trabalho, foi construído um robô móvel equipado com um microcontrolador Arduino UNO, que se desloca em um ambiente de cultura genérico de maneira autônoma, com o objetivo de gerar trajetórias de acordo com dados obtidos de sensores de umidade, além de outras possíveis finalidades que serão definidas posteriormente, em função do ambiente de cultura onde a plataforma móvel será utilizada. Inicialmente, a metodologia do trabalho teve como base, a construção de uma plataforma móvel e a realização de alguns testes a fim de obter dados de velocidade da mesma. Posteriormente, foi projetado e construído um ambiente de cultura genérico, onde foram instalados nove sensores de umidade. No software Matlab, foi realizada toda a programação do sistema, de maneira que, quando necessário que o robô se desloque até um sensor que identificou umidade baixa, o algoritmo MRP (Mapa de Rotas Probabilístico) gera o melhor caminho até este sensor e, com os parâmetros fornecidos pelo algoritmo, são realizados os cálculos necessários para identificar todas as distâncias e ângulos que devem ser percorridos. Com isso, essas informações são passadas ao robô por meio de comunicação Bluetooth e o mesmo interpreta as informações recebidas e percorre o caminho desejado. Esse sistema pode ser facilmente adaptado pelo desenvolvedor e utilizado em qualquer ambiente, desde que ele seja conhecido e mapeado previamente.

Palavras chaves: Robótica móvel, Mapa de rotas probabilístico, Planejamento de trajetórias.

1. INTRODUÇÃO

Devido ao crescente estudo a respeito de robótica e inteligência artificial com o passar dos anos, o ser humano vem conseguindo projetar e construir robôs cada vez mais robustos e confiáveis, no que diz respeito a capacidade de realizar ações que geralmente são realizadas por seres humanos.

A finalidade de um robô móvel, é que ele seja capaz de se deslocar de um ponto a outro dentro de um determinado ambiente, sem colidir com nenhum obstáculo ao longo do percurso. Para que isso seja possível, é necessário o planejamento do caminho que será percorrido pelo mesmo, considerando-se os obstáculos presentes no ambiente em questão. Este planejamento geralmente é realizado por meio de algoritmos de otimização e é extremamente importante, pois o seu objetivo é fornecer o menor caminho possível entre os dois pontos, diminuindo assim, a energia gasta pelo robô, o tempo gasto para locomoção e, conseqüentemente, aumentar consideravelmente a eficiência de um processo que dependa deste.

Atualmente, existem robôs móveis que são capazes de se deslocar no meio terrestre, aéreo ou aquático. Essa diversidade de ambientes faz com que estes sejam muito estudados dentro das universidades, e cada vez mais utilizados em grandes indústrias, bases militares, agricultura e ambientes domésticos. Além da diversidade de ambientes, também existe grande diversidade nas possíveis tarefas que estes podem realizar, como por exemplo o monitoramento por câmeras, transporte de carga, colheita de frutos, irrigação, limpeza, reconhecimento de áreas desconhecidas, entre outros.

Neste artigo, será construído um robô móvel autônomo, com a finalidade de validar as trajetórias geradas pelo algoritmo MRP (Mapa de Rotas Probabilístico), de acordo com dados obtidos de sensores de umidade, em um ambiente mapeado e construído para testes, que simula uma casa de vegetação. Este tema foi escolhido com base no crescente aumento populacional, que resulta em uma maior demanda de alimentos no mercado e, conseqüentemente, é necessário um aumento na eficiência da produção de alimentos no meio agrícola a fim de atender essa demanda.

2. REFERENCIAL TEÓRICO

2.1. ROBÓTICA

Segundo Russel e Norvig (2013), os robôs são máquinas projetadas para realizar tarefas no mundo físico. Os atuadores que são equipados no robô (rodas, garras e articulações), permitem que tais tarefas sejam realizadas nos mais diversos ambientes de trabalho. Além disso, eles podem ser equipados com sensores e, assim, entender o ambiente a sua volta.

Russel e Norvig (2013), classificam os robôs em três categorias: manipuladores, robôs móveis e manipuladores móveis. Os manipuladores são fixos em seu local de trabalho e em sua grande maioria, estão situados nas grandes indústrias. Os robôs móveis não ficam em um local fixo e podem se movimentar no ambiente, isso faz com que eles possuam um espaço de trabalho maior do que os manipuladores. E por fim, os manipuladores móveis são uma junção das outras duas categorias.

Atualmente, existem robôs que atuam em diversos setores, indo desde setores domésticos, até setores como agricultura, indústrias e áreas militares. O setor militar explora bastante a robótica móvel em uma diversidade grande de ambientes e tarefas, por isso, este setor foi um dos responsáveis pelo desenvolvimento da robótica móvel. No ambiente militar existem robôs que são responsáveis pelo transporte de materiais, desarme de bombas, monitoramento de instalações, reconhecimento de locais hostis, ataques com armas de fogo e espionagem (VOTH, 2004). Na Figura 1, é apresentada a *TALON SWORDS*, uma plataforma móvel desenvolvida por Foster-Miller, capaz de carregar armas letais de alto calibre. Três plataformas dessas já serviram no Iraque por mais de um ano (MARCHANT et al., 2011).

Figura 1 – Robô *TALON SWORDS* desenvolvido por Foster-Miller



Fonte: Marchant et al. (2011)

Como citado anteriormente, os robôs móveis possuem um espaço de trabalho maior do que os manipuladores e, conseqüentemente, isso faz com que possam ser utilizados em uma variedade maior de aplicações do que os manipuladores. Porém, a maior problemática dos robôs móveis é o sistema de alimentação, que geralmente é constituído por baterias. A curta durabilidade dessas baterias, pode ocasionar na parada de determinada operação, quando a mesma descarregar. Portanto, os manipuladores possuem vantagem nesse quesito, uma vez que eles são fixos ao seu local de trabalho e não dependem de baterias para alimentar seu sistema (PAIVA, 2017). Na Figura 2, são apresentados alguns exemplos famosos de robôs móveis.

Figura 2 – Exemplos de Robôs Móveis: a) Roomba [iRobot 2009], b) Automower [Husqvarna 2009], c) Aibo [Sony 2009] e d) Asimo [Honda 2009]



Fonte: Wolf et al. (2009)

2.2. APLICAÇÃO DE ROBÔS MÓVEIS EM CASAS DE VEGETAÇÃO

A demanda global de alimentos vem aumentando nas últimas décadas, devido ao grande aumento demográfico mundial. Com isso, em 2004 se iniciou um período de escassez de alimentos e, conseqüentemente, seus preços aumentaram drasticamente, ocasionando uma crise internacional (ONU, 2008).

Neste contexto, fica clara a necessidade do aumento da produtividade de alimentos e melhorias nos processos agrícolas. Segundo Gaspareto (2013), as estufas são uma ótima alternativa para o aumento da produtividade dos alimentos, por isolarem a cultura do ambiente externo e, conseqüentemente, reduzir as chances de pragas e doenças. Por serem isolados, esses ambientes também são mais fáceis de serem monitorados e controlados.

O uso de robôs autônomos é muito importante no processo agrícola, no que diz respeito a fatores econômicos e ambientais, uma vez que robôs projetados para aplicação de pesticidas podem executar essa tarefa de forma precisa, em diversos pontos dentro da estufa. Ou seja, é possível ajustar a quantidade de pesticida utilizado pelo robô e, assim, haverá uma otimização no processo, além da diminuição da agressão ao meio ambiente e dos custos de produção de alimentos (WOLF et al., 2009).

Paiva (2017), ressalta que as estufas são ambientes que proporcionam maior facilidade para locomoção dos robôs, uma vez que esses ambientes são cuidadosamente projetados para que a plataforma móvel se desloque até o local desejado sem que haja nenhuma colisão.

No trabalho de Hayashi et al. (2005), foi desenvolvido um robô para colheita de morangos. Foi realizado um teste com o robô parado e 20 morangos no local, dos quais 15 foram colhidos com sucesso. Cada fruta levou de 9,3 até 17,9 segundos para ser colhida. O projeto pode ser visto na Figura 3.

Figura 3 – Robô móvel para colheita de morangos



Fonte: Hayashi et al. (2005)

Nezhad, Massh e Komleh (2011), desenvolveram um robô focado na colheita de tomates equipado com uma câmera capaz de identificar o tomate e a distância entre a plataforma e o fruto. O projeto é mostrado na Figura 4.

Figura 4 – Robô móvel para colheita de tomates



Fonte: Nezhad, Massh e Komleh. (2011)

Outra aplicação da robótica móvel na agricultura é o robô Aurora, descrito por Mandow et al. (1996). O projeto foi desenvolvido por pesquisadores na Universidade de Málaga na Espanha, com o objetivo de se locomover de maneira autônoma dentro de estufas e realizar tarefas como a pulverização de agentes químicos na plantação, diminuindo assim, a necessidade de humanos transitarem dentro da estufa para realizarem tais tarefas.

Figura 5 – Robô Aurora



Fonte: Universidade de Málaga

2.3. PLANEJAMENTO DE TRAJETÓRIAS

O planejamento da trajetória é uma etapa extremamente importante em um processo que dependa de um robô móvel. Este planejamento é aplicado em diversos campos científicos como robótica, teoria de controle e inteligência artificial. No que diz respeito a robótica, o planejamento de uma trajetória tem como principal objetivo fornecer o menor caminho possível, livre de obstáculos e respeitando as limitações mecânicas do robô. Além disso, com os avanços nos estudos a respeito do planejamento de trajetórias, também pode-se considerar algumas outras variáveis como incertezas, erro do modelo e dinâmica da plataforma móvel (LAVALLE, 2016).

Raja e Pugazhenth (2012), dividem o planejamento de trajetórias em duas categorias: *off-line* e *on-line*. No planejamento de trajetórias *off-line*, todos os obstáculos, curvas e informações a respeito do ambiente no qual o robô irá trafegar são conhecidas. Caso existam outros objetos que se movimentem no ambiente, além do robô, suas trajetórias também devem ser conhecidas. Já no planejamento de trajetórias *on-line*, não se tem conhecimento de todos os elementos presentes no local, portanto, o robô faz o reconhecimento do ambiente através de sensores, à medida que o mesmo se desloca pelo local.

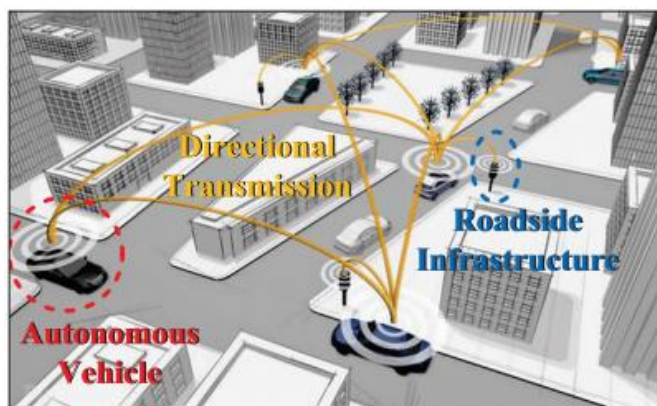
No planejamento de trajetórias, é muito importante a simplificação da representação da estrutura do robô. Uma das maneiras mais utilizadas para realizar essa simplificação é o “Espaço C” (LOZANO-PEREZ; WESLEY, 1979). Essa ferramenta consiste em desconsiderar as dimensões do robô e considerá-lo como um ponto e, expandir os obstáculos de acordo com as dimensões do robô, a fim de compensar a representação pontual do mesmo (RAJA; PUGAZHENTHI, 2012).

2.4. COMUNICAÇÃO V2I (VEÍCULO PARA INFRAESTRUTURA)

Uma rede de comunicação sem fio robusta é a base para sistemas de transporte conectados (DEY et al., 2019). Através de dois modos sem fio, Veículo para Infraestrutura (V2I) e

Veículo para Veículo (V2V), veículos autônomos podem adquirir mais dados de tráfego a fim de otimizar sua estratégia de direção. Trabalhos existentes nessa área tentam integrar o WiFi comercial, Bluetooth, ZigBee, WiMax e 4G nos veículos (KONG et al., 2017).

Figura 6 – Modos de comunicação V2I e V2V



Fonte: Kong et al. (2017)

Segundo Gheorghiu et al. (2017), a tecnologia Bluetooth é conveniente no desenvolvimento de redes de dispositivos de pequena extensão e que precisam trocar informações, devido ao seu baixo consumo de energia e baixo alcance.

Operando na frequência 2,4GHz, a tecnologia Bluetooth conecta dispositivos em um raio que pode variar de 10 a 100 metros de distância, na velocidade de até 3Mbps, dependendo da classe do dispositivo Bluetooth (PIYARE; TAZIL, 2011).

Segundo Bekkelien (2012), os dispositivos Bluetooth são divididos em três classes diferentes, onde cada classe possui um alcance diferente. O alcance máximo de cada uma das classes de dispositivos Bluetooth pode ser visto na Tabela 1.

Tabela 1 – Alcance máximo de cada classe de dispositivos Bluetooth

Class	Range
Class 1	100 m
Class 2	10 m
Class 3	5 m

Fonte: Bekkelien (2012).

3. METODOLOGIA

3.1. CONSTRUÇÃO DE UM ROBÔ MÓVEL

Para a construção da estrutura do robô móvel, utilizou-se um kit pronto que contém o chassi em acrílico, dois motores DC (3~6 V), duas rodas de borracha e uma roda boba para dar sustentação ao chassi. A estrutura do robô pode ser vista na Figura 7.

Figura 7 – Chassi do robô móvel



Fonte: Eletrogate (<https://www.eletrogate.com/kit-chassi-2wd-robo-para-arduino>)

Após montada a estrutura, foram acoplados os seguintes componentes eletrônicos ao robô:

Arduino UNO: programado para controlar os movimentos do robô;

Módulo Ponte H Dupla L298N: utilizado para realizar o controle por modulação de largura de pulso (PWM) da velocidade dos dois motores presentes no robô. A velocidade de cada motor é controlada de maneira independente;

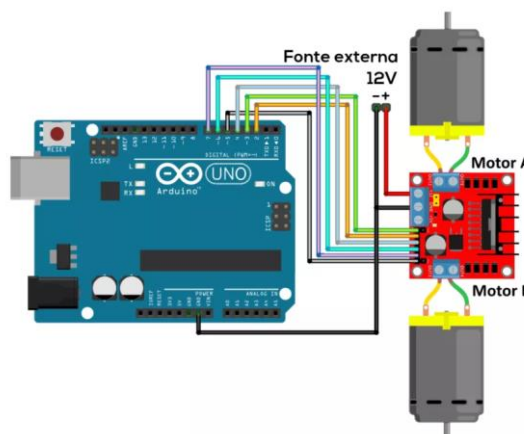
Protoboard: utilizada para realizar as conexões elétricas de maneira facilitada, sem necessitar de solda;

8 pilhas de 1,5 Volts cada (12 Volts no total): utilizadas para alimentar todo o sistema eletrônico do robô;

Acelerômetro e Giroscópio MPU-6050: utilizado apenas para coletar alguns dados de variação de ângulo do robô.

A ligação dos motores é apresentada na Figura 8.

Figura 8 – Esquemático de ligação dos motores do robô móvel

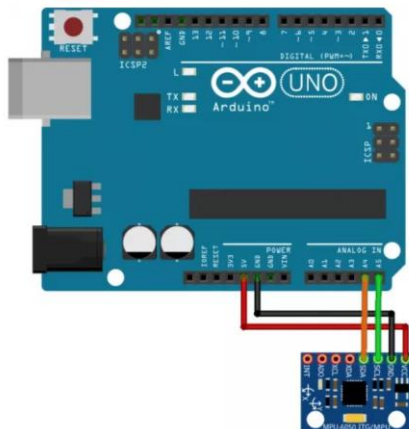


Fonte: Vida de Silício (<https://portal.vidadesilicio.com.br/ponte-h-l298n-controle-velocidade-motor/>)

A ponte H é alimentada com 12 Volts, provenientes das 8 pilhas mencionadas anteriormente. Para cada motor são utilizados 3 pinos digitais do Arduino, um para cada terminal do motor e o terceiro pino é utilizado para enviar o sinal PWM para a ponte H e, assim, controlar a velocidade do motor.

Na Figura 9, é apresentado o esquemático de ligação do acelerômetro MPU-6050.

Figura 9 – Esquemático de ligação do MPU-6050

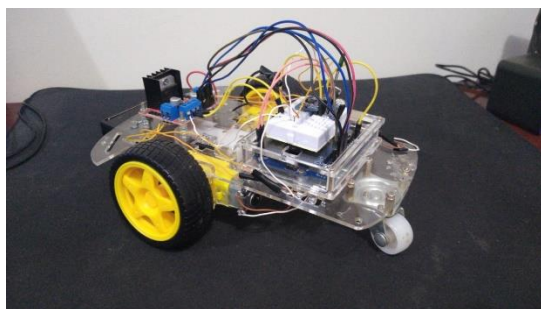


Fonte: Auto Core Robótica

(<https://autocorerobotica.blog.br/aprenda-a-utilizar-o-acelerometro-mpu6050/>)

O MPU-6050 é alimentado com uma tensão de 5 Volts provenientes do Arduino e o recebimento dos dados de variação de ângulo nos eixos X, Y e Z são recebidos nas portas analógicas A4 e A5 por meio do protocolo de comunicação I2C. O robô com todos os componentes acoplados no chassi pode ser observado na Figura 10.

Figura 10 – Robô móvel utilizado no projeto



Fonte: Do Autor (2020).

3.2. VALIDAÇÃO DA PLATAFORMA MÓVEL

Com o robô construído, foram realizados testes a fim de obter sua velocidade em 3 situações que são essenciais para o robô se locomover. São elas:

Robô se locomovendo para frente: para calcular a velocidade com que o robô se desloca para frente, mediu-se o tempo que o mesmo leva para percorrer a distância de 1 metro e assim, com o tempo e a distância, foi possível calcular, de maneira aproximada, a velocidade linear do veículo ($V = \frac{\Delta S}{\Delta t}$).

Para que o robô se locomova em linha reta, o motor direito deve girar a 40% de sua capacidade e o motor esquerdo a 30%. Essa diferença, se deve a uma distribuição ligeiramente assimétrica, dos componentes acoplados ao chassi do robô.

Posteriormente, observou-se que a velocidade a ser considerada deve ser diferente de acordo com a distância solicitada. Por isso, foram feitas marcações no chão de 10 em 10 centímetros e, empiricamente, foram encontradas as velocidades necessárias para o robô percorrer cada distância com exatidão. A velocidade para a distância de 1 metro calculada inicialmente, serviu apenas como parâmetro inicial para os ajustes realizados.

O Arduino presente no robô foi programado para realizar os testes com o auxílio da função “anda”, que recebe a distância a ser percorrida como parâmetro e, com a velocidade calculada para a distância de 1 metro (que foi sendo ajustada empiricamente para cada distância), é calculado o tempo que os motores devem ficar ligados para percorrer determinada distância e os motores ficam ligados durante esse tempo. As velocidades obtidas estão contidas na Tabela 2.

Tabela 2 – Velocidades lineares quando o robô se locomove para frente

Distancias (cm)	Velocidades (cm/s)
10	25
20	30
30	33.1
40	34
50	34.5
60	35
70	35.5
80	35.5
90	35.5
100	35.5
>100	35.5

Fonte: Do Autor (2020).

Com todas as velocidades calculadas, foram feitos os ajustes necessários na função “anda” para que a mesma funcione para qualquer uma das distâncias contidas na tabela.

Robô virando para a esquerda: para calcular a velocidade com que o robô se vira para a esquerda, fez-se com que o mesmo girasse 180° no sentido anti-horário e mediu-se o tempo que levou para o robô atingir determinado ângulo. Assim, foi possível calcular a velocidade angular do robô. Os dados de variação de ângulo ao longo do tempo foram obtidos através do módulo MPU-6050.

Para o robô virar para a esquerda, o motor direito gira a 40% de sua capacidade e o motor esquerdo fica parado.

Posteriormente, observou-se que a velocidade a ser considerada deve ser diferente de acordo com o ângulo solicitado. Por isso, foram feitas marcações no chão de 10 em 10 graus e, por tentativa e erro, encontrou-se as velocidades necessárias para o robô girar cada ângulo. A velocidade para o ângulo de 180 graus calculada inicialmente, serviu apenas como parâmetro inicial para os ajustes realizados.

O Arduino presente no robô foi programado, para realizar os testes, com o auxílio da função “vira”, que recebe o ângulo a ser percorrido como parâmetro e confere se o mesmo é positivo, se sim, significa que o robô deve virar para a esquerda. Com o ângulo passado por parâmetro e a velocidade calculada para o ângulo de 180 graus (que foi sendo ajustada empiricamente para cada ângulo), é calculado o tempo que o motor direito deve ficar ligado para percorrer determinado ângulo e o motor fica ligado durante esse tempo.

As velocidades obtidas estão contidas na Tabela 3.

Tabela 3 – Velocidades angulares quando o robô vira para a esquerda

Ângulos (graus)	Velocidades Reais (graus/s)
10	80
20	110
30	115
40	120
50	120
60	120
70	120
80	130
90	130
100	135
110	135
120	135
130	140
140	140
150	145
160	145
170	135
180	135

Fonte: Do Autor (2020).

Com todas as velocidades calculadas, foram feitos os ajustes necessários na função “vira” para que a mesma funcione para qualquer um dos ângulos contidos na tabela.

Robô virando para a direita: análogo ao robô virando para a esquerda. As velocidades obtidas estão contidas na Tabela 4.

Tabela 4 – Velocidades angulares quando o robô vira para a direita

Ângulos (graus)	Velocidades Reais (graus/s)
-10	55
-20	55
-30	65
-40	80
-50	80
-60	100
-70	110
-80	110
-90	110
-100	120
-110	120
-120	125
-130	125
-140	125
-150	125
-160	125
-170	125
-180	135

Fonte: Do Autor (2020).

Com todas as velocidades calculadas, foram feitos os ajustes necessários na função “vira” para que a mesma funcione para qualquer um dos ângulos contidos na tabela.

3.3. MONTAGEM DO AMBIENTE DE SIMULAÇÃO

Após calculados todos os parâmetros do robô móvel, foi realizado o mapeamento de um ambiente de cultura genérico e construiu-se uma pista de simulação para o mesmo, com dimensões de 245x245 cm. A pista construída pode ser observada na Figura 11.

Figura 11 – Pista construída para realização do projeto



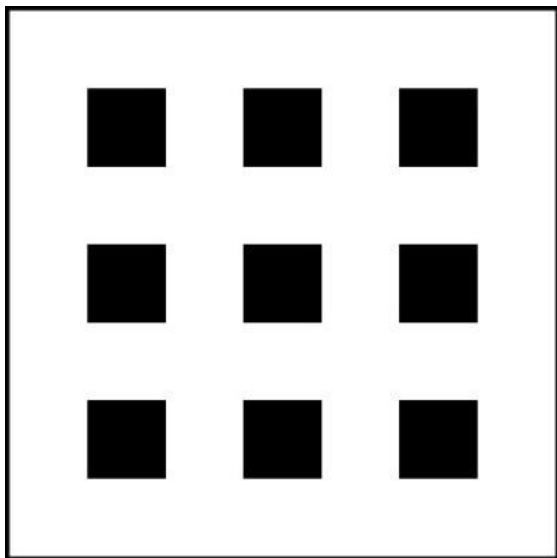
Fonte: Do Autor (2020).

Cada um dos 9 quadrados pequenos, contidos na pista, simulam alguma possível cultura que possa existir no local. Cada um desses quadrados possui dimensões de 35x35 cm.

Com o auxílio do Photoshop foi criada uma imagem com a extensão .bmp (*bitmap* – mapa de bits) da pista, pois o algoritmo que será utilizado para gerar os caminhos (MRP) reconhece apenas imagens com essa extensão. A imagem

criada possui uma resolução de 1225x1225 pixels. Portanto, 5 pixels na imagem correspondem a 1 centímetro na pista real.

Figura 12 – Representação da pista real



Fonte: Do Autor (2020).

Posteriormente, observou-se que o algoritmo utilizado gera caminhos que passam muito próximos das bordas dos quadrados internos e, para evitar que o robô passe por dentro desses quadrados, realizou-se um preenchimento neles. Os quatro lados de cada um dos quadrados foram aumentados em 8 centímetros (40 pixels), essa medida corresponde à metade da largura total do robô móvel.

3.4. O ALGORITMO MRP (MAPA DE ROTAS PROBABILÍSTICO)

O algoritmo MRP é responsável por gerar o caminho que deve ser percorrido pelo robô e deve-se utilizar o *software* Matlab para rodá-lo.

O algoritmo utiliza uma Grade de Ocupação Binária (*Binary Occupancy Grid*) que atribui o valor 1 para pixels que já estão ocupados na imagem (obstáculos), e 0 para pixels livres. Portanto, o MRP irá utilizar pixels com o valor 0 para traçar o caminho do robô.

Para utilizar este algoritmo, deve-se primeiro mapear o local em que deseja executar o MRP, criando uma imagem em *bitmap* deste local. O mapeamento do local foi realizado no item 3.3. Após a criação da imagem, deve-se colocá-la na mesma pasta em que está contido o algoritmo. No arquivo “Astart.m”, deve-se colocar o nome da imagem no campo “map”, escolher a posição inicial do robô no campo “source” e a posição final no campo “goal”. O algoritmo utiliza coordenadas bidimensionais do plano cartesiano, portanto, as posições devem ser inseridas no formato “(y, x)”, onde ‘y’ e ‘x’ indicam os pixels nos eixos ‘y’ e ‘x’ da imagem, respectivamente. E, por fim, deve-se informar o número de nós

desejados. Esses ajustes nos parâmetros iniciais do algoritmo estão ilustrados na Figura 13.

Figura 13 – Parâmetros iniciais do algoritmo MRP

```
map=im2bw(imread('Pista.bmp')); % input map read from a bmp file.
source=[1158 273]; % source position in Y, X format
goal=[448 623]; % goal position in Y, X format
k=100; % number of points in the PRM
display=true; % display processing of nodes
```

Fonte: Do Autor (2020).

Após a definição dos parâmetros iniciais, o código é executado e inicialmente ocorre a etapa de “bombardear” a imagem com os nós de maneira aleatória, apenas nos pixels livres. Após o “bombardeamento” dos nós, é pedido ao usuário que clique na tela para dar início a próxima etapa que é a de aprendizado, onde o algoritmo gera todos os possíveis caminhos através dos nós. E por fim, é pedido novamente que o usuário clique na tela e a trajetória final que o robô deverá percorrer é exibida na tela. No final é mostrado o tempo de execução do programa e também é obtida uma matriz contendo as coordenadas de todos os nós que fazem parte do caminho gerado.

Na Figura 14, é ilustrado o funcionamento do algoritmo para 50 nós em um ambiente genérico.

Figura 14 – Funcionamento do MRP para 50 nós

S. No.	k	Roadmap	Path	Path Length	Execution Time (sec)*
1.	50			736	3.19

Fonte: Kala (2014)

3.5. VALIDAÇÃO DO MRP

É necessário realizar a validação do algoritmo MRP no ambiente mapeado anteriormente. Esta validação é importante para garantir que não haja nenhum erro de sintaxe ou na lógica do algoritmo.

Para isso, foram realizados 3 testes diferentes variando-se as posições inicial e final da trajetória, o número de nós e, além disso, também foram acrescentados alguns obstáculos no ambiente. Os testes e seus resultados são apresentados a seguir:

Teste 1: O primeiro teste foi realizado sem acrescentar nenhum obstáculo no ambiente, a posição inicial foi 1158, no eixo y, e 273, no eixo x. A posição final foi 448, no eixo y, e 623, no eixo x. Este teste foi realizado duas vezes variando-se o número de nós.

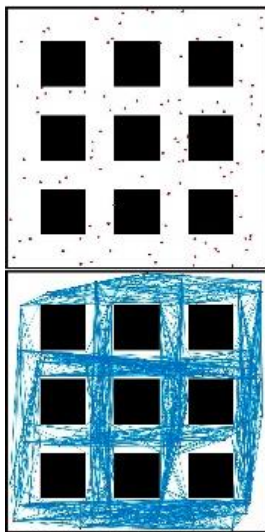
Tabela 5 – Parâmetros e resultados referentes ao Teste 1

Caso	Número de nós	Tempo de execução (s)	Distância (cm)
1	100	12,08	183,182
2	200	40,89	177,942

Fonte: Do Autor (2020).

As etapas de “bombardeamento” e aprendizado do caso 1 são apresentadas na Figura 15.

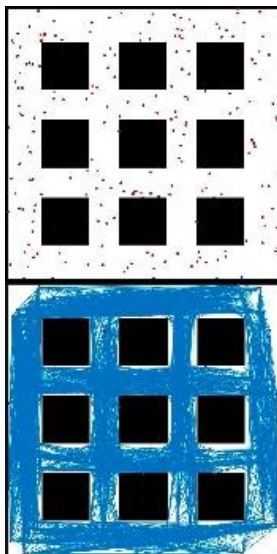
Figura 15 – Etapas de “bombardeamento” e aprendizado, respectivamente, do Caso 1



Fonte: Do Autor (2020).

As etapas de “bombardeamento” e aprendizado do caso 2 são apresentadas na Figura 16.

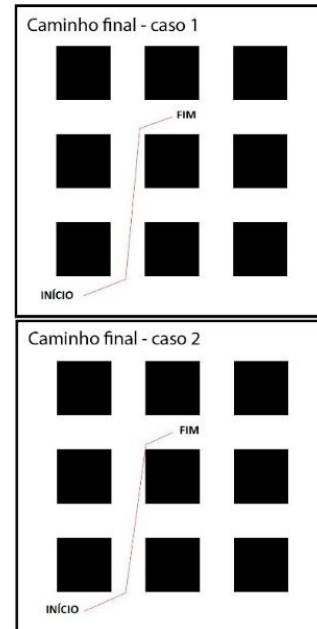
Figura 16 – Etapas de “bombardeamento” e aprendizado, respectivamente, do Caso 2



Fonte: Do Autor (2020).

Os caminhos finais gerados nos casos 1 e 2 são mostrados na Figura 17.

Figura 17 – Caminho final gerado no Caso 1 e no Caso 2, respectivamente.



Fonte: Do Autor (2020).

Teste 2: Para o segundo teste foram acrescentados alguns obstáculos no ambiente, as posições inicial e final (y, x) foram alteradas para (1158, 973) e (273, 88), respectivamente. Este teste foi realizado duas vezes variando-se o número de nós.

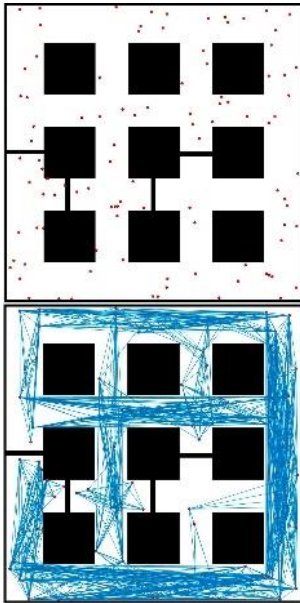
Tabela 6 – Parâmetros e resultados referentes ao Teste 2

Caso	Número de nós	Tempo de execução (s)	Distância (cm)
1	100	18,49	318,08
2	200	63,44	298,52

Fonte: Do Autor (2020).

As etapas de “bombardeamento” e aprendizado do caso 1 são apresentadas na Figura 18.

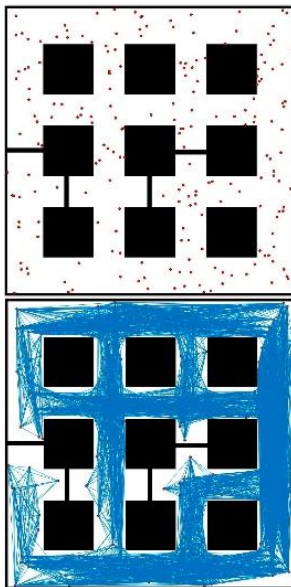
Figura 18 – Etapas de “bombardeamento” e aprendizado, respectivamente, do Caso 1



Fonte: Do Autor (2020).

As etapas de “bombardeamento” e aprendizado do caso 2 são apresentadas na Figura 19.

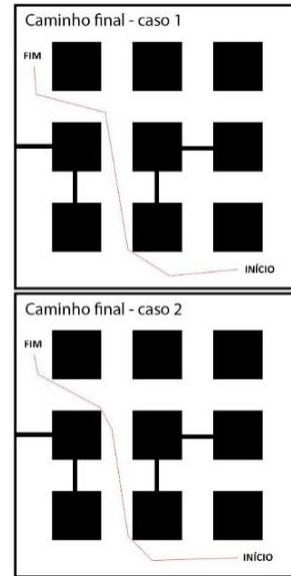
Figura 19 – Etapas de “bombardeamento” e aprendizado, respectivamente, do Caso 2



Fonte: Do Autor (2020).

Os caminhos finais gerados nos casos 1 e 2 são mostrados na Figura 20.

Figura 20 – Caminho final gerado no Caso 1 e no Caso 2, respectivamente.

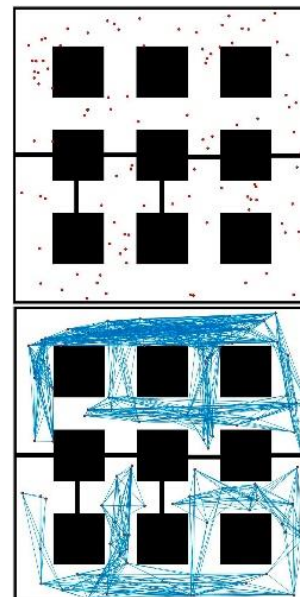


Fonte: Do Autor (2020).

Teste 3: Para o terceiro teste as posições inicial e final (y, x) foram mantidas em (1158, 973) e (273, 88), respectivamente. Foram adicionados obstáculos de maneira que fosse impossível o algoritmo traçar um caminho entre o ponto inicial e o final.

As etapas de “bombardeamento” e aprendizado do Teste 3 são apresentadas na Figura 21.

Figura 21 – Etapas de “bombardeamento” e aprendizado, respectivamente, do Teste 3



Fonte: Do Autor (2020).

Como esperado, na última etapa o caminho não foi gerado e uma mensagem de erro foi emitida na janela de comando do Matlab.

Figura 22 – Mensagem de erro: “caminho não encontrado”

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

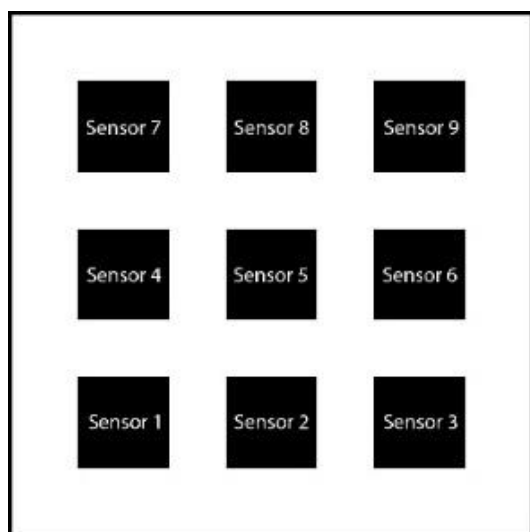
Fonte: Do Autor (2020).

3.6. INSTALAÇÃO DOS SENSORES NO AMBIENTE CONSTRUÍDO

A fim de automatizar o projeto, foram instalados nove sensores de umidade de solo no ambiente e, para realizar a leitura da umidade através desses sensores, utilizou-se um Arduino Mega. Este modelo de Arduino foi escolhido devido ao maior número de pinos analógicos em relação a outros modelos.

A disposição dos sensores no ambiente é apresentada na Figura 23.

Figura 23 – Posição dos sensores no ambiente



Fonte: Do Autor (2020).

O ambiente real após a instalação dos sensores é mostrado na Figura 24.

Figura 24 – Ambiente real com sensores instalados



Fonte: Do Autor (2020).

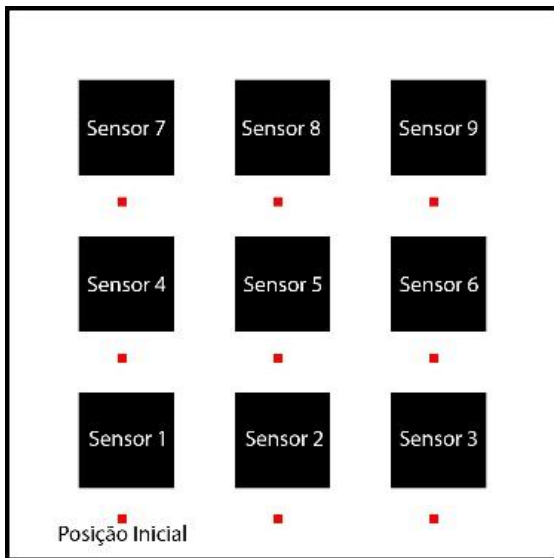
Os sensores foram colocados em recipientes com terra úmida para simular uma cultura com solo úmido. Quando o sensor é retirado do recipiente, é simulado um solo seco e o robô móvel deve se locomover até este sensor que identificou o solo seco. O robô apenas será liberado para se locomover novamente quando o solo voltar ao seu estado normal, ou seja, é como se o robô estivesse realizando uma irrigação no local e apenas fosse liberado ao terminar a irrigação, porém neste projeto o fim da irrigação é simulado colocando-se o sensor de volta no recipiente. Feito isso, o robô aguarda até que outro sensor obtenha a leitura de solo seco para se locomover até o mesmo. Esta é a ideia geral deste projeto.

3.7. PROGRAMAÇÃO PARA MONITORAR O AMBIENTE

Devido ao MRP ser utilizado no Matlab, o Arduino presente no ambiente também foi programado no *software* Matlab. Para isso, foi necessária a instalação da *toolbox* “ArduinoIO”. Antes de explicar como foi feita a programação, algumas considerações são necessárias:

A fim de simplificar a programação, em cada sensor existe apenas 1 posição possível para o robô se locomover, que é na parte de baixo do sensor e, sempre que o robô estiver parado, é considerado que ele está de frente para a direita. Inicialmente, é considerado que o robô está posicionado na parte de baixo do sensor 1. Na Figura 25, são ilustradas a posição inicial e todas as possíveis posições para o robô se deslocar. Os quadrados vermelhos indicam as possíveis posições do robô.

Figura 25 – Possíveis posições para o robô se deslocar



Fonte: Do Autor (2020).

Após todas as considerações iniciais, os códigos que foram implementados no *software* Matlab são explicados a seguir.

Função “ESCLIN.m”: Realiza o escalonamento linear de uma escala para outra. Esta função recebe os valores inicial e final da escala atual, os valores inicial e final da nova escala desejada e o valor atual da variável a ser escalonada, e retorna o valor da variável escalonada.

Função “posicaoSensor.m”: Recebe um número equivalente a um sensor e retorna as coordenadas no formato (y, x) deste sensor de acordo com as posições mostradas na Figura 25.

Função “PRM.m”: Recebe as coordenadas iniciais e finais do caminho a ser gerado como parâmetros e, além de gerar o caminho e plotar a imagem na tela, a função retorna a matriz “path” com 2 colunas, onde a primeira coluna indica as coordenadas y de cada nó do caminho gerado e a segunda coluna indica as coordenadas x de cada nó do caminho gerado. Seu funcionamento já foi explicado no item 3.4. Porém foram realizadas algumas mudanças no código para transformá-lo em uma função.

Função “caminho.m”: Recebe a matriz “path” (gerada pela função PRM) como parâmetro e retorna os vetores “distancias”, “ângulos” e “sinal”. O significado de cada vetor será explicado posteriormente.

Inicialmente, deve-se colocar as dimensões da pista real e da imagem em bitmap utilizada pelo MRP. E então, são declarados os vetores “distancias”, “ângulos” e “sinal” e o número de elementos de cada vetor é o mesmo número de nós contidos na matriz “path”.

Uma variável chamada “anguloAtual” é declarada como 0 (zero). Esta variável indica qual a angulação do robô em determinado momento.

O vetor “distancias” é preenchido com as distâncias em linha reta que o robô deverá percorrer. Com a matriz “path” é possível saber o deslocamento em x e y entre os nós do caminho. Assim, por Pitágoras, encontra-se cada uma das distâncias do caminho a ser percorrido. A função “ESCLIN.m” é usada para converter a distância em pixels, para centímetros.

O vetor “ângulos” é preenchido com os ângulos que o robô deve virar antes de percorrer cada distância. Esses ângulos são calculados com a cotangente da relação entre o deslocamento no eixo y e o deslocamento no eixo x, porém sempre levando em consideração a angulação atual do robô. Além disso, é importante conferir se o ângulo calculado é menor do que -180° ou maior do que 180° . Se sim, faz-se com que o robô vire pelo lado contrário a fim de virar o menor ângulo possível.

Após calcular todos os ângulos, verifica-se o ângulo atual do robô e, se necessário, é calculado mais um ângulo para que o robô termine o percurso virado para a direita (0°).

O vetor “sinal” é preenchido de acordo com o vetor “ângulos”. Se a posição x do vetor “ângulos” foi negativa, ela será multiplicada por -1 e a posição equivalente no vetor “sinal” será preenchida com o valor 1. Caso contrário, o vetor “ângulos” não é alterado e a posição equivalente no vetor “sinal” é preenchida com o valor 0. Portanto, os valores no vetor “ângulos” ficarão todos positivos e o vetor “sinal” indicará se os valores do vetor “ângulos” são positivos ou negativos. Isso terá extrema importância posteriormente.

Função “vaiParaSensor.m”: Recebe o sensor para qual o robô deverá se locomover e as coordenadas atuais do robô. A função retorna os vetores: “distancias”, “ângulos” e “sinal”. Além de retornar as coordenadas da nova posição do robô. Esta função é chamada quando se deseja que o robô se mova para um determinado sensor.

Inicialmente, obtém-se as coordenadas do sensor de destino através da função “posicaoSensor.m”. Posteriormente, a função “PRM.m” é chamada a fim de gerar o caminho que deverá ser percorrido pelo robô.

A posição atual do robô é atualizada como a posição do sensor de destino. E, por fim, a função “caminho.m” é chamada para se obter as distâncias e os ângulos que o robô deverá percorrer para completar o trajeto.

“Amain.m”: É aqui onde ocorre a programação do Arduino através do Matlab para leitura dos sensores de umidade. Este é o arquivo principal do projeto. Para que o sistema todo se inicie, basta rodar este arquivo.

Inicialmente, são definidas as portas analógicas do Arduino onde são ligados os sensores de umidade. Definem-se também os parâmetros de umidade ideal como 50%, e umidade baixa

como 20%, ou seja, a “irrigação” deve terminar quando o solo atingir a umidade de 50% e quando a umidade estiver abaixo de 20%, o robô deve ir até o local.

A posição inicial do robô é definida no sensor 1, como explicado anteriormente. Uma variável “sensor” é criada com o valor inicial 0 (zero). Esta variável indica se algum sensor está com a umidade baixa, se nenhum sensor estiver com umidade baixa, o valor 0 é atribuído a variável.

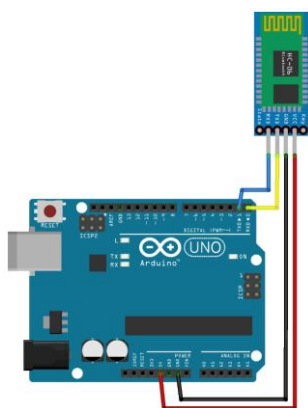
Após a definição dos parâmetros iniciais, é iniciado um loop infinito. Onde primeiramente o valor da variável “sensor” é mostrado na tela e depois é realizada a leitura da umidade em cada sensor através da função “analogRead”. Esta função obtém o valor analógico que a porta analógica correspondente ao sensor recebe. Este valor varia de 0 até 1024, onde 0 indica que o solo está 100% úmido e, 1024, indica que o solo está totalmente seco. Portanto, após a leitura do valor recebido na porta analógica, é realizado um escalonamento para se obter a umidade em porcentagem através da função “ESCLIN.m”. Isso é feito para todos os sensores.

Após a leitura dos sensores, é verificado se a umidade do sensor 1 está baixa. Se sim, é atribuído o valor 1 a variável “sensor” e é chamada a função “vaiParaSensor.m” com a finalidade de que o robô se locomova até o sensor. Enquanto a umidade do sensor 1 não for maior do que a umidade ideal, definida como 50%, apenas a ação de leitura da umidade do sensor 1 é realizada. Quando a umidade do sensor 1 voltar ao normal, é atribuído o valor 0 a variável “sensor”. Se a umidade do sensor 1 não estiver baixa, é conferida a umidade do sensor 2 e assim por diante. Até que a umidade do último sensor seja conferida e o ciclo se inicie novamente na etapa de leitura da umidade de cada um dos sensores.

3.8. COMUNICAÇÃO V2I (VEÍCULO PARA INFRAESTRUTURA)

A comunicação entre o veículo e o ambiente foi feita por Bluetooth. Para isso, foi instalado um módulo Bluetooth HC-06 no robô como mostrado na Figura 26.

Figura 26 – Esquemático de ligação do módulo HC-06



Fonte: Do Autor (2020).

O módulo HC-06 é um dispositivo Bluetooth de Classe 2, isso significa que o módulo possui um alcance máximo de 10 metros, o que é mais do que o suficiente para cobrir todo o ambiente que foi projetado e construído para testes. O HC-06 é alimentado com 5 Volts provenientes do Arduino. A comunicação do HC-06 com o Arduino é uma comunicação Serial e é realizada por meio dos pinos RXD (pino 0 digital) e TXD (pino 1 digital) presentes no Arduino.

O HC-06 instalado no robô móvel é mostrado na Figura 27.

Figura 27 – Módulo Bluetooth HC-06 instalado no robô



Fonte: Do Autor (2020).

Para estabelecer a comunicação, foi acrescentado um trecho de código no arquivo “Amain.m”, que é responsável por parear o notebook com o módulo HC-06 presente no robô. Este código pode ser observado na Figura 28.

Figura 28 – Trecho de código responsável por estabelecer a comunicação Bluetooth entre o Notebook e o robô

```
25 % Estabelece a comunicação bluetooth com o carro
26 - instrhwinfo('Bluetooth', 'HC-06');
27 - b = Bluetooth('HC-06', 1);
28 - fopen(b);
```

Fonte: Do Autor (2020).

Ainda no *software* Matlab, criou-se uma função que tem a finalidade de enviar os dados do caminho gerado para o robô móvel por meio da comunicação Bluetooth estabelecida.

Função “carroBluetooth”: Recebe o objeto Bluetooth criado anteriormente (HC-06) e também recebe os vetores “distancias”, “ângulos” e “sinal” como parâmetros. A função não retorna nada.

Essa função calcula o tamanho dos vetores recebidos como parâmetros e, através da função “fwrite”, ela manda o tamanho dos vetores e os 3 vetores para o robô através da comunicação Bluetooth.

A função “fwrite” envia um inteiro de 8 bits por vez. Isso significa que ela apenas consegue enviar valores inteiros dentro do intervalo de 0 a 255. Por isso, o vetor “sinal” é tão importante, pois sem ele, não seria possível enviar valores negativos ao robô.

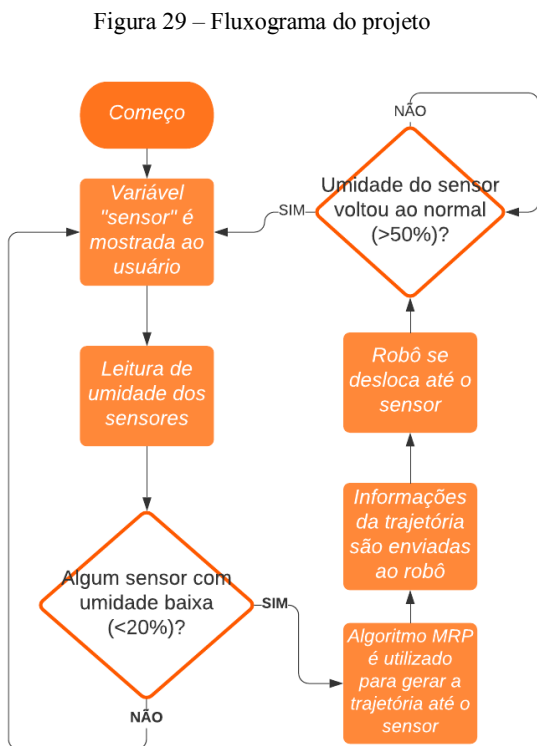
O Arduino presente no robô foi programado para receber os dados através do módulo Bluetooth HC-06 e interpretá-los para percorrer o caminho solicitado. Sua implementação é dividida em 3 partes.

Parâmetros: Nesta parte são declarados todos os pinos referentes ao Arduino e variáveis que serão necessárias para o pleno funcionamento do robô.

Leitura dos dados: Etapa responsável por receber todos os dados que chegam pela comunicação Bluetooth e armazená-los de maneira correta. Inicialmente, é checado se existem dados para serem lidos. Se sim, o primeiro dado (tamanho dos vetores) é lido, assim é possível saber a quantidade de números que estarão contidos em cada um dos vetores. Assim, os 3 vetores são recebidos e preenchidos.

Deslocamento do robô: Primeiramente, é confirmado se todos os dados foram devidamente recebidos. Se sim, os dados são impressos no monitor serial do Arduino e o robô começa a se deslocar com o auxílio das funções “vira” e “anda”, apresentadas anteriormente.

Na Figura 29, é apresentado um fluxograma que ilustra todo o funcionamento do projeto de maneira resumida, para facilitar a compreensão do mesmo.



Fonte: Do Autor (2020).

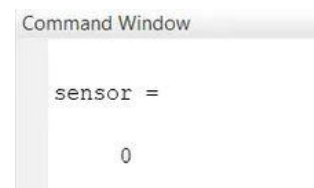
4. RESULTADOS

4.1. TESTES DE VALIDAÇÃO COM O ROBÔ MÓVEL

Com a finalização da construção e programação de todos os elementos do projeto, foram realizados alguns testes para validá-lo. Esta validação é importante para garantir que não haja nenhum erro de sintaxe ou na lógica do algoritmo.

Teste 1: Rodou-se o arquivo “Amain.m” no Matlab com todos os sensores na terra úmida. O valor 0 (zero) foi mostrado na tela, o que indica que todos os sensores estão úmidos, como esperado.

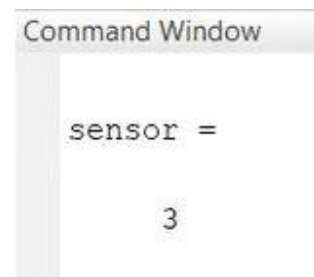
Figura 30 – Indicação de que todos os sensores estão úmidos



Fonte: Do Autor (2020).

Após isso, o sensor 3 foi retirado do solo úmido e foram mostradas na tela as seguintes informações.

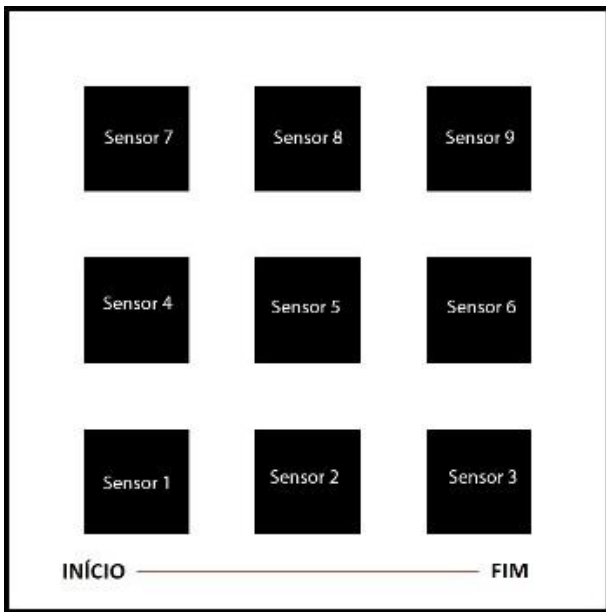
Figura 31 – Indicação de que o sensor 3 está seco



Fonte: Do Autor (2020).

Após a identificação de umidade baixa no sensor 3, foi gerado o caminho até o mesmo.

Figura 32 – Caminho gerado para o Teste 1



Fonte: Do Autor (2020).

Observa-se que o caminho possui início no Sensor 1, como explicado anteriormente.

Os vetores “distancias”, “ângulos” e “sinal” também são mostrados na tela.

Figura 33 – Vetores contendo as informações de deslocamento do robô para o Teste 1

```
distancias =  
    138.0145      0  
  
angulos =  
     0     0  
  
sinal =  
     0     0
```

Fonte: Do Autor (2020).

Observa-se que existe apenas uma distância, isso indica que o robô irá percorrer um caminho em linha reta, como esperado. Na Figura 34 é mostrada a posição final do robô após se deslocar até o sensor.

Figura 34 – Posição do robô após o deslocamento do Teste 1



Fonte: Do Autor (2020).

O quadrado vermelho na imagem indica a posição final desejada, isso mostra que o algoritmo funciona de maneira correta e a posição final do robô é aceitável, uma vez que o mesmo funciona em malha aberta e não possui nenhum tipo de controle aplicado.

Teste 2: Rodou-se o arquivo “Amain.m” no MATLAB com todos os sensores na terra úmida, com exceção do sensor 5. O valor 5 foi mostrado na tela, o que indica que o sensor 5 estava seco, como esperado.

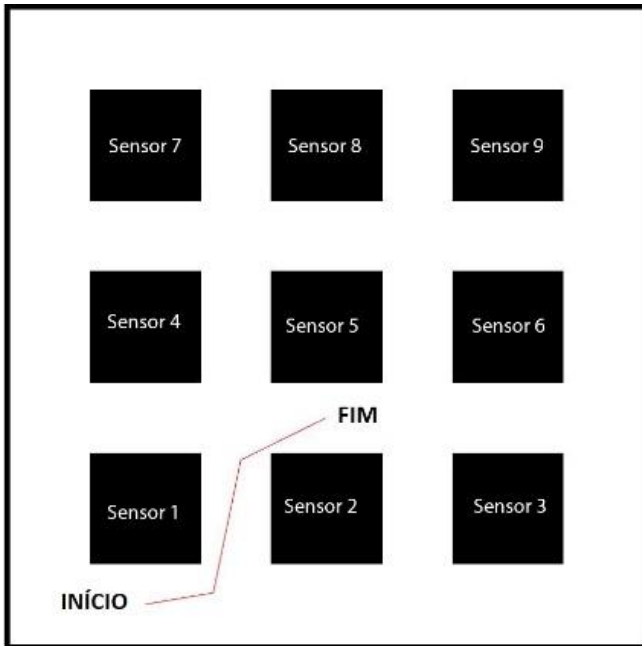
Figura 35 – Indicação de que o sensor 5 está seco

```
Command Window  
  
sensor =  
      5
```

Fonte: Do Autor (2020).

Após a identificação de umidade baixa no sensor 5, foi gerado o caminho até o mesmo.

Figura 36 – Caminho gerado para o Teste 2



Fonte: Do Autor (2020).

Observa-se que o caminho possui início no Sensor 1, como explicado anteriormente. Os vetores “distancias”, “ângulos” e “sinal” também são mostrados na tela.

Figura 37 – Vetores contendo as informações de deslocamento do robô para o Teste 2

```

distancias =
    24.3187  49.9460  34.3628  0

angulos =
    9.5332  68.7700  52.0150  26.2883

sinal =
    0  0  1  1
    
```

Fonte: Do Autor (2020).

Agora existem várias distâncias e ângulos e, de acordo com o vetor “sinal”, os dois primeiros ângulos são positivos e os dois últimos são negativos. Isso indica que o robô irá percorrer um caminho mais complexo do que no Teste 1, como esperado. Na Figura 38 é mostrada a posição final do robô após se deslocar até o sensor.

Figura 38 – Posição do robô após o deslocamento do Teste 2



Fonte: Do Autor (2020).

O quadrado vermelho na imagem indica a posição final desejada, isso mostra que o algoritmo funciona de maneira correta e a posição final do robô é aceitável, uma vez que o mesmo funciona em malha aberta e não possui nenhum tipo de controle aplicado.

5. CONCLUSÃO

Este trabalho teve o objetivo de construir uma plataforma móvel que interpreta os dados obtidos do algoritmo Mapa de Rotas Probabilístico (MRP) e, assim, se movimenta de forma autônoma em um cenário que representa uma casa de vegetação. Existem nove sensores de umidade distribuídos pelo cenário. Quando um sensor detecta umidade baixa, o algoritmo MRP calcula a melhor rota até o mesmo, e essas informações são enviadas ao robô por meio de comunicação Bluetooth. Os resultados obtidos foram extremamente satisfatórios, uma vez que os códigos desenvolvidos são capazes de calcular corretamente, a partir da trajetória gerada pelo algoritmo MRP, todas as distâncias e ângulos que devem ser percorridos pelo robô, para que o mesmo chegue ao local desejado. Este projeto é uma abordagem inicial para aplicações de robôs autônomos em casas de vegetação e possibilita uma continuidade nos estudos a respeito do assunto, podendo ocorrer avanços significativos e, assim, se obter um projeto capaz de ser implementado no setor agrícola da Universidade Federal de Lavras. Como dito nos referenciais teóricos deste artigo, este tipo de tecnologia é capaz de otimizar e aumentar a produtividade de um processo agrícola, o que é extremamente importante para atender à crescente demanda de alimentos, proveniente do grande aumento demográfico mundial.

5.1. PERSPECTIVAS FUTURAS

Para estudos posteriores, ficam algumas sugestões de melhorias que podem ser feitas a fim de se atingir um projeto funcional e aplicável em um ambiente agrícola real:

- Sensores: a instalação de sensores no robô, baseados em ultrassom, imagem ou luz, aumentariam a segurança do mesmo, evitando possíveis colisões com outros objetos.
- Monitoramento da trajetória: é extremamente importante que a trajetória real do robô seja monitorada, para que a mesma seja a mais próxima possível da trajetória gerada pelo algoritmo. Por exemplo, pode-se usar uma câmera para monitorar a trajetória do robô, calcular o erro e, através da aplicação de técnicas de controle, executar comandos nos atuadores a fim de corrigir o erro em tempo real.
- Comunicação V2I: para que o projeto possa ser utilizado em ambientes extensos, deve-se utilizar um dispositivo Bluetooth de classe 1 ou outro modo de comunicação sem fio, dependendo da extensão do ambiente.
- Plataforma móvel mais robusta: o robô utilizado neste projeto é um kit didático para estudos de robótica móvel. Portanto, sugere-se o desenvolvimento de uma plataforma com componentes mais robustos e confiáveis, com espaço reservado para a instalação de um manipulador responsável por realizar tarefas no ambiente de cultura.
- Manipulador: para que o projeto seja funcional, é essencial o desenvolvimento e instalação de um manipulador capaz de realizar irrigação, colheita e, possivelmente, outras tarefas imprescindíveis no meio agrícola.

7. REFERÊNCIAS BIBLIOGRÁFICAS

- BEKKELIEN, A. **Bluetooth Indoor Positioning**; University of Geneva: Geneva, Switzerland, 2012.
- BENTO, F. A. P. **Determinação de Rotas Otimizadas para Robótica Móvel em uma Casa de Vegetação**. 2019. 70 p. Monografia (Bacharelado em Engenharia de Controle e Automação) – Universidade Federal de Lavras, Lavras, MG, 2019.
- DEY, K. C. et al. “**Vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication in a heterogeneous wireless network - Performance evaluation**,” *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 2016.
- 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.
- GHEORGHIU, R. A. et al. (2017). **Bluetooth Latency Analysis for Vehicular Communications in a Wi-Fi Noisy Environment**. 10.1109/TSP.2017.8075956.
- HAYASHI, S. et al. (2005). **Robotic Harvesting Technology for Fruit Vegetables in Protected Horticultural Production**.
- KALA, R. (2014) **Code for Robot Path Planning using Probabilistic Roadmap**, Indian Institute of Information Technology Allahabad, Available at: <http://rkala.in/codes.html>.
- KONG, L. et al. "Millimeter-Wave Wireless Communications for IoT-Cloud Supported Autonomous Vehicles: Overview, Design, and Challenges," in *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62-68, January 2017, doi: 10.1109/MCOM.2017.1600422CM.
- LAVALLE, S. M. **Planning Algorithms**. Cambridge University Press., 2006.
- LOZANO-PÉREZ, T.; WESLEY, M. A. **A algorithm for planning collision-free paths among polyhedral obstacle**. *Communications of the ACM*, v. 22, p. 560-570, 1979.
- MANDOW, A. et al. **The autonomous mobile robot AURORA for greenhouse operation**. *IEEE Robotics & Automation Magazine*, v. 3, p.18-28, 1996.
- MARCHANT, G. et al. “**International Governance of Autonomous Military Robots**”, *Columbia Science and Technology Law Review*, 2011.
- NEZHAD, M. A. K. B.; MASSH, J.; KOMLEH, H. E. **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, S. B. **Introdução a robótica**. Grupo Gen-LTC, 2013.
- 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.
- PIYARE, R.; TAZIL, M. "Bluetooth based home automation system using cell phone," *2011 IEEE 15th International*

Symposium on Consumer Electronics (ISCE), Singapore, 2011, pp. 192-195, doi: 10.1109/ISCE.2011.5973811.

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

RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 3a Edição. Editora Campus, 2013.

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

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, M.; CHEN, C.; KUO, Y. **Implementation of intelligent air conditioner for fine agriculture**. *Energy and Buildings*, v. 60, p. 364-371, 2013.