V SIINTEC INTERNATIONAL SYMPOSIUM ON INNOVATION AND TECHNOLOGY
SIMPÓSIO INTERNACIONAL DE INOVAÇÃO E TECNOLOGIA
CIRCULAR ECONOMY | ECONOMIA CIRCULAR

DOOGIE MOUSE: UMA PLATAFORMA *OPEN SOURCE* PARA APLICAÇÃO DE ALGORITMOS INICIAIS DE INTELIGÊNCIA ARTIFICIAL EM ROBÓTICA MÓVEL

Caio Alves Amaral[1]; Mateus dos Santos Meneses;

[1] Centro Universitário SENAI CIMATEC; Salvador/BA;
[2] Centro Universitário SENAI CIMATEC; Salvador/BA;
mateusmenezes95@gmail.com

Resumo: Um estudo realizado no Brasil sobre o atual cenário de pesquisas na robótica demonstra que 53,48% dos robôs usados em pesquisas nacionais são construídos no país. Assim, para que pesquisadores eliminem a etapa de projeto de um novo robô em cada nova pesquisa, este trabalho busca realizar a concepção de uma plataforma open source do tipo micromouse, utilizando o framework ROS e o simulador Gazebo, em que conceitos primários de inteligência artificial podem ser aplicados. Para isso, foi formulado um benchmarking e, porconseguinte, foi elaborada a arquitetura geral e eletrônica e idealizado o modelo mecânico do Doogie Mouse. Os resultados preliminares da simulação expressaram como o tipo de malha do modelo mecânico influenciam na estabilidade e realidade da simulação.

Palavras-Chave: *Open source*, ROS, Inteligência Artificial, Robótica Móvel, Micromouse.

DOOGIE MOUSE: A OPEN SOURCE PLATFORM TO APPLICATION OF ARTIFICIAL INTELLIGENCE INITIAL ALGORITHMS ON MOBILE ROBOTIC

Abstract: A survey carried in Brazil about the current research scenario in robotics shows that 53,48% of the robots used in national research are built in the country. Thus, for researchers to eliminate the design stage of a new robot in each new research, this work seeks to realize the design of an open source micromouse platform, using the ROS framework and the Gazebo simulator, whereby primary concepts of artificial intelligence can be applied. For this purpose, was made a benchmarking and, as consequence, the general and electronic architecture were elaborated and the mechanic model of the Doogie Mouse designed. Preliminary simulation results expressed how the mesh type of the mechanical model influences the stability and reality of the simulation.

Keywords: Open source, ROS, Artificial Intelligence, Mobile Robotic, Micromouse.

.1. INTRODUÇÃO

As aplicações em IA (Inteligência Artificial) tem crescido nos últimos anos [1]-[2]. Entre tantas aplicações, a IA tem sido bastante utilizada na construção de soluções

dentro da robótica móvel, seja a partir de mecanismo de reconhecimento do ambiente em que está inserido o robô [3-4] ou mesmo através do planejamento de trajetórias de movimento.

A competição Micromouse realizada pelo Institute of Electrical and Electronics Engineers (IEEE) faz uso de alguma dessas soluções, na qual um robô diferencial é posto na entrada de um labirinto e deve encontrar uma determinada posição final a partir de seus algoritmos de inteligência no menor tempo possível.

Este trabalho, portanto, propõe a concepção de uma plataforma open source do tipo micromouse [6], permitindo o desenvolvimento de aplicações de conceitos primários de inteligência artificial, sobre forma de algoritmos de busca em plataformas móveis. A integração entre a inteligência artificial e a plataforma móvel será intermediada pelo Robot Operating System (ROS), um dos mais utilizados frameworks de robótica [5], assim, permitindo o uso de sua vasta biblioteca de pacotes open source produzidos por sua comunidade de usuários, reduzindo o tempo gasto "reinventando a roda" ao utilizar-se de ferramentas já desenvolvidas no framework [5]. Além disso, o ROS possui integração com o ambiente de simulação Gazebo [8], que será utilizado na concepção do robô (e de seus algoritmos de controle, navegação e percepção) e na realização de testes dentro do ambiente que serão confrontados futuramente com os testes realizados com o protótipo físico.

Assim, propõe-se uma solução para o atual cenário de pesquisas na robótica realizadas no Brasil, no qual 53,48% dos robôs usados em pesquisas nacionais são construídos no país, devido aos altos custos na importação das plataformas de robótica disponíveis no mercado [7]. A plataforma robótica permitirá aos pesquisadores um maior foco em aplicações e soluções na robótica, ao eliminar a etapa de projeto de um novo robô e de sua interface de comunicação em cada nova pesquisa, contribuindo com o crescimento tecnológico e tornando o desenvolvimento da robótica mais acessível no país.

2. MATERIAIS E MÉTODOS

A priori, este robô faz parte de um trabalho multidisciplinar de conclusão de curso de graduação que utiliza a metodologia TheoPrax, envolvendo alunos de Engenharia de Controle e Automação e Engenharia Mecânica. Assim, os tópicos subsequentes descrevem como a plataforma foi projetada, explorando a arquitetura geral do sistema, seu modelo mecânico e sua arquitetura eletrônica.

2.1 Matriz de Comparação

Ao início do projeto, foi realizado um benchmarking visando comparar o que já existia dentro do estado da arte dos micromouse. Através dele, montou-se uma matriz de comparação de forma a quantificar atributos considerados mais significativos para o robô. Levou-se em conta, portanto, a existência da documentação disponível e seu nível de clareza; o uso de algum framework de robótica; se faz uso ou suporta algum

ambiente de simulação; diversidade de linguagens que a plataforma pode ser programada; como é realizada a interface do usuário; quantidade de diferentes sensores e se a plataforma é expansível, podendo acrescentar a ela outros recursos (seja em hardware ou em software).

Tabela 1. Matriz de Comparação

Área	Peso	GreenGiant	Kumamoto National College	Smartmouse	WolfieMouse	Raspberry Pi Mouse V2
Documentação	1	0,5	2	1	2	2.5
Framework de Robótica	1	0	0	0	0	2
Ambiente de Simulação	0,8	0	0	2	1	3
Linguagens de Programação	0,5	1	1	2	2	2
Interface do Usuário	0,5	3	1	1	2	3
Sensores	0,5	2	1	1	3	1
Expansível	0,8	1	0	0	0	1
	Somatório:	**4,3**	**3,5**	**4,6**	**6,3**	**10.5**

Assim, mapeou-se 5 principais modelos concorrentes, conforme visto na Tabela 1. O modelo que possui maior nível de detalhamento e mais pontuou em relação a todas as áreas foi o Raspberry Pi Mouse V2. Contudo, toda a documentação desse projeto encontra-se em língua japonesa, sendo inacessível à maior parte do público.

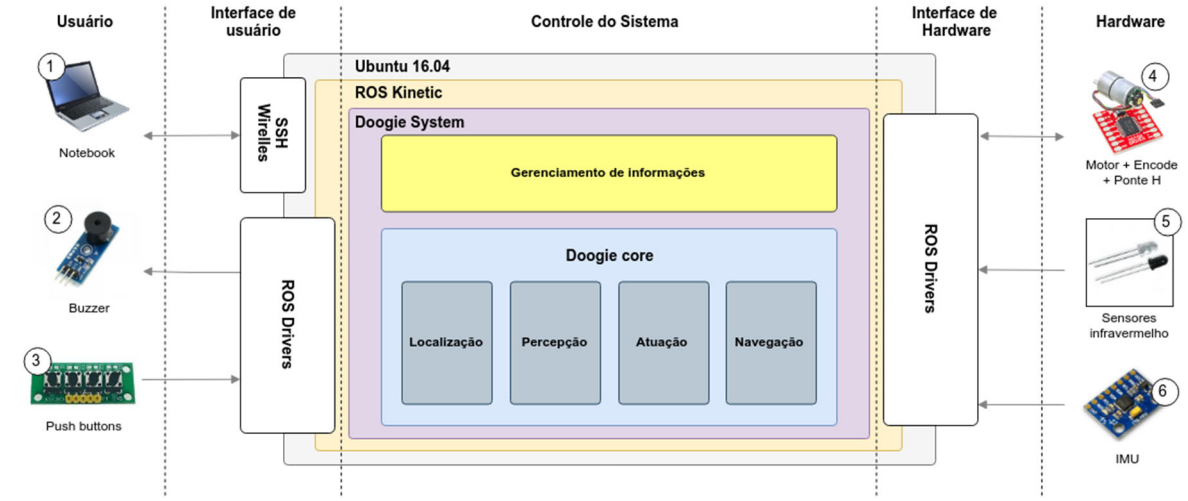
A partir disso, pode-se levantar requisitos para que o projeto do Doogie Mouse contemple seu objetivo. Observou-se que existe uma falta generalizada de documentação acessível dos projetos e pouco uso dos framework de robótica e de ambientes de simulação. O próprio Raspberry Pi Mouse V2, mesmo sendo capaz de rodar em ambiente de simulação Gazebo a partir do ROS, não possui nenhum modelo de labirinto ou qualquer cenário em que se possa simular o robô, nem possui exemplos de implementação de algoritmos de busca. Portanto, essas funções deverão estar disponíveis no modelo final do Doogie Mouse.

2.2 Arquitetura Geral

Inicialmente, foi elaborada uma arquitetura geral, expressa na Figura 1, para demonstrar de forma macro o funcionamento do Doogie Mouse. Embora não explicitado neste diagrama, o robô possui uma unidade central de processamento responsável pelo seu controle. Diante disso, foi especificado para essa função o microcomputador Raspberry PI Zero W.

Conforme Figura 1, o usuário pode interagir com o sistema de controle de três modos: acessando o robô remotamente utilizando o protocolo de rede Secure Shell (SSH) sobre uma rede Ethernet sem fio; percebendo alertas sonoros emitidos por um Buzzer; interagindo fisicamente através de dois botões momentâneos (push button). Para a mobilidade, foram escolhidos dois micro motores de corrente contínua com caixa de redução modelo HP-6V que são acionados por um driver de potência com dois canais independentes, também conhecido como ponte H. O dispositivo selecionado para o acionamento dos motores foi o TB6612FNG. Já para fornecer informações (posição, orientação e velocidade) para o controle de trajetória da plataforma, foram especificados dois sensores: um par de encoder magnético com sensores de efeito hall TLE4946-2K e uma Inertial Measurement Unit (IMU) MPU6050. Do mesmo modo, para orientar o sistema de controle nas tomadas de decisões de movimentação, foi adotado sensores ópticos reflexivos compostos por um Light Emitter Diode (LED) infravermelho (emissor) e um fototransistor (receptor). Os componentes escolhidos para isso foram o SFH4545 e o TFT4300 respectivamente.

Figura 1: Arquitetura geral do Doogie Mouse



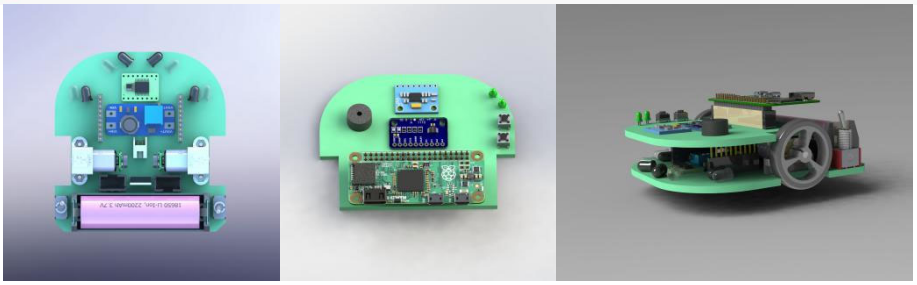
Por fim, para o controle do sistema, foi definido a utilização do framework ROS versão Kinetic Kame e o sistema operacional Ubuntu 16.04, ambos instalados na Raspberry Pi. Dessa forma, foram estabelecidos 4 módulos vinculados ao ROS que compõem o núcleo do robô: localização, que dispõe a célula do labirinto onde o robô se encontra; percepção, que fornece informações das paredes existentes em volta da plataforma; atuação, responsável pelo acionamento e controle dos motores; navegação, cujo o objetivo é escolher o caminho ótimo de movimentação, utilizando algoritmos introdutórios de inteligência artificial como busca em largura [6], busca em profundidade [8], Flood Fill [9-10] e A* [11]. Ademais, a modularidade do sistema permite que o desenvolvedor implemente outros algoritmos de inteligência. Por conseguinte, o módulo Gerenciamento de Informações atua como um interlocutor entre o ROS e o usuário. Esses módulos são interfaceados através de ROS drivers, realizando conversões de protocolos de comunicação e comandos de controle em dados reconhecidos pelo ambiente ROS.

2.3 Modelo Mecânico

Para o design do robô, foi utilizado como um ponto de partida o TON-BOT V1.1, plataforma desenvolvida pela Ioton Technology [12]. Além disso, uma vez que o robô a ser desenvolvido é do tipo micromouse, ele deve ter suas dimensões não superiores a uma seção retangular de 25 x 25 cm [6].

A partir dessas premissas e da análise feita na subseção 3.1, buscou-se um design mecânico simples e de maior leveza. Dessa forma, será utilizado como frame do robô as próprias PCB, buscando posicionar suas rodas de forma a mantê-las alinhadas ao centro de massa de todo o conjunto mecânico. Para tanto, uma modelagem em CAD inicialmente foi realizada através de ferramenta Solidworks, buscando iterativamente a melhor disposição de seus elementos físicos (rodas, sensores e demais componentes eletrônicos das placas). O modelo mecânico final pode ser visualizado na Figura 2.

Figura 2. Arquitetura geral do Doogie Mouse

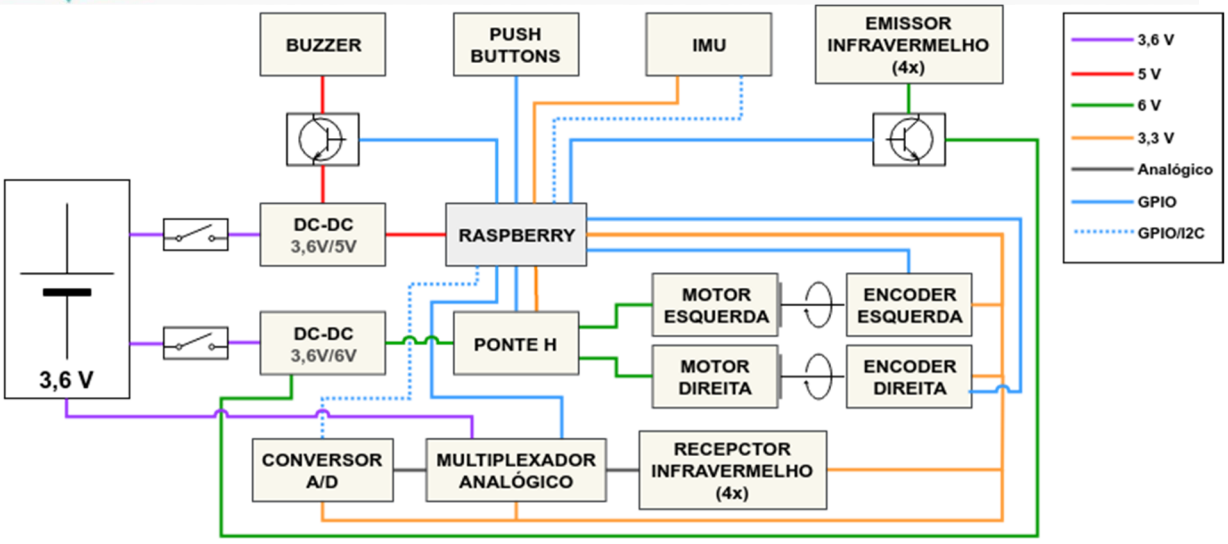


Da esquerda para direita visualiza-se as placas inferior, superior além do modelo do robô visto em perspectiva. A placa inferior possui 98 mm de comprimento e 92,90 mm de largura, enquanto a placa superior possui 75 mm de comprimento e 92,90 mm de largura. Foi necessário o uso de duas placas para melhor adequação dos componentes eletrônicos sem atrapalhar eventuais manutenções no dispositivo nem dificultar sua montagem.

2.4 Eletrônica

Com intuito de facilitar a replicação do robô por usuários que queiram utilizá-lo, optou-se pelo uso de breakout boards, que são placas eletrônicas pré-montadas. O Doogie possui seis breakout boards: dois conversores de tensão DC-DC, modelos MT3608 e U1V10F5, um conversor Analógico/Digital (A/D) ADS1115, uma IMU, um multiplexador analógico 74HC4051 e uma ponte H. Os demais componentes como resistores, transistores, LEDs infravermelho, fototransistores e conectores, são soldados diretamente na Placa de Circuito Impresso (PCI). Esses componentes foram distribuídos nas PCIs superior e inferior do robô, conforme Figura 2. A fim de fornecer energia elétrica para esses componentes, foi especificado uma bateria do tipo Li-Ion modelo 18650 com tensão nominal igual à 3,6 V. A arquitetura elétrica na Figura 3 demonstra como esses componentes descritos estão interligados eletricamente. A propósito de melhor visualização, o referencial de tensão (Ground - GND) foi omitido.

Figura 3. Representação elétrica do Doogie Mouse



3. RESULTADOS PRELIMINARES

A simulação foi realizada no Gazebo 7.0.0, a partir do ROS através de suas bibliotecas do ros_control em um Inspiron 15 7548.Para o melhor uso do ambiente, tanto o FPS quanto o Real Time Factor foram levados em conta, para uma visualização pelo desenvolvedor mais estável e realista, respectivamente. Diante disso, foi necessário realizar simplificações na malha do robô para maior fluidez da simulação, suprimindo alguns dos elementos presentes no modelo mecânico. A fim de atingir o maior valor de FPS e um Real Time Factor mais próximo de um, utilizou-se de testes iterativos, avaliando como o número de faces das malhas afetam a simulação. Uma vez que o Gazebo possui tanto malhas para efeito visual quanto malhas de “collision box” (o envoltório do robô capaz de simular contato físico com outros objetos dentro da simulação), os dois foram incluídos no teste. Além disso, comparou-se a simulação com o uso de malhas carregadas a partir de arquivos COLLADA com as carregadas a partir de STL.

Tabela 2. Comparativo entre malhas carregadas no Gazebo

	Visual (Faces)	Colisão (Faces)	FPS	Real Time Factor
Doogie_lite_dae	70.410	70.410	~1,5	0,35
Doogie_real_dae	1.091.337	1.091.337	~0,6	0,03
Doogie_lite_stl	70.398	70.398	~4.2	0,33

Tabela 3. Comparativo com simplificação da malha de colisão

	Visual (Faces)	Colisão (Faces)	FPS	Real Time Factor
Doogie_lite_dae	70.410	6	~8,0	0,52
Doogie_real_dae	1.091.337	6	~3,5	0,52

Os testes foram realizados em um Dell Inspiron 15 7548, com o Ubuntu Xenial (16.04 LTS) e ROS Kinetic. Através deles, conforme a Tabela 2, é evidente como é essencial a simplificação da malha de colisão. Sem nenhum tipo de otimização, a taxa de quadros do Gazebo chega a 0,6 FPS, com a versão mais completa do robô (idêntico ao modelo mecânico gerado no SolidWorks), enquanto que quando a collision box é simplificada como um prisma de 6 faces, sua taxa de quadros aumenta em quase 6 vezes, atingindo uma média de 3,5 FPS.

Quanto a diferença entre as malhas em STL e COLLADA, nota-se uma melhora no número de quadros por segundo com o uso do formato STL: 4,2 FPS contra 1,5 FPS. Por outro lado, o Real Time Factor entre COLLADA e STL não diferiram significativamente: 0,35 e 0,33 respectivamente. Os resultados possivelmente são decorrentes da existência de menos arestas na malha, exigindo menor poder computacional para a simulação.

4. CONCLUSÃO (ARIAL 12)

O presente trabalho apresentou o desenvolvimento de um robô open source, explicando como um plataforma baseada na competição Micromouse foi projetada utilizando novas concepções de construção como o uso do framework ROS, da Raspberry PI Zero W e da ferramenta de simulação Gazebo. Os resultados preliminares apontam uma melhor otimização da simulação ao se reduzir o número de faces presentes nas malhas renderizadas na simulação, atingindo na melhor configuração, 8,0 FPS com o Real Time Factor de 0,52. Posteriormente, um labirinto também será gerado no Gazebo, utilizando uma configuração de malha semelhante a obtida nos resultados, e a partir disso serão simulados os algoritmos de busca dentro da simulação. Além disso, como próximo passo para a concepção do Doogie Mouse, será realizada a montagem do protótipo e, consequentemente, os testes em um labirinto físico. Assim, poderão ser confrontados os resultados em simulação e em ambiente real, verificando a acurácia da simulação. Ademais, por se tratar de um projeto open source, todos os documentos de hardware e software bem como um guia do usuário, estarão disponíveis para que o projeto possa ser replicados por estudantes e pesquisadores que queiram utilizar a plataforma.

5. REFERÊNCIAS

[1]LI, Yangyuan. **Stable Modeling on Resource Usage Parameters of MapReduce Application**. Brain, Budapeste, v. 9, n. 2, p. 45-62, 1 maio 2018. DOI 10.5281/zenodo.1245887. Disponível em: https://zenodo.org/record/1245887#.XVsH-3VKi00. Acesso em: 17 ago. 2019.

[2]PENSTEIN, C. R. **A social spin on language analysis**. Nature, Pittsburgh, n. 545, p. 166-167, 10 maio 2017. Disponvel em: https://www-nature.ez148.periodicos.capes.gov.br/articles/545166a#notes. Acesso em: 17 ago. 2019.

[3]NASCIMENTO, Matheus. **Detecção e classificação de obstáculos para robô de inspeção de linhas de transmissão**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia de Controle e Automação) - Faculdade Área 1, Salvador, 2018.

[4]ALCANTARA, Pedro. **Classificação de Obstáculos Baseada no Classificador k-Nearest Neighbors Aplicada a um Robô de Inspeção de Linha de Transmissão**. 2018. Dissertação (Mestrado em Engenharia Elétrica) - Escola Politécnica, Universidade Federal da Bahia (UFBA), Salvador, 2018. Disponível em: http://www.ppgee.eng.ufba.br/teses/cb28211baf41ec8222af61ab8a6d059.pdf. Acesso em: 11 ago. 2019.

[5]TELLEZ, Ricardo. **A History of ROS (Robot Operating System)**. [S. l.], 9 jul. 2019. Disponível em: http://www.theconstructsim.com/history-ros/. Acesso em: 11 ago. 2019.

[6]KIBLER, S. G. et al. **IEEE Micromouse for mecHistoryhatronics research and education**. 2011 IEEE International Conference on Mechatronics, ICM 2011 - Proceedings, p. 887–892, 2011.

[7]NETO, R. P. et al. **Robótica na educação: Uma revisão sistemática dos Últimos 10 anos**. Simpósio Brasileiro de Informática na Educação, n. XXVI, p. 8, 2015.

[8]KOENIG, N.; HOWARD, A. **Design and use paradigms for Gazebo, an open-source multi-robot simulator**. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Los Angeles, v. 3, p. 2149-2154, 28 set. 2004. Disponvel em: https://ieeexplore.ieee.org/abstract/document/1389727. Acesso em: 17 ago. 2019.

[9]RUSSELL, Stuart. **Inteligência artificial**. 3. ed. Rio de Janeiro: Elservier, 2013. xxi, 988 p. ISBN 9788535237016.

[10]TJIHARJADI, S.; SETIAWAN, E. **Design and implementation of a path finding robot using flood fill algorithm**. International Journal of Mechanical Engineering and Robotics Research, v. 5, n. 3, p. 180–185, 2016.

[11]TJIHARJADI, S.; WIJAYA, M. C.; SETIAWAN, E. **Optimization maze robot using A* and flood fill algorithm**. International Journal of Mechanical Engineering and Robotics Research, v. 6, n. 5, p. 366–372, 2017.

[12]**IOTON** Technologie. Disponível em: https://github.com/iotontech. Acesso em 17 ago. 2019.