

SUBMETIDO 18/03/2022

APROVADO 02/06/2022

PUBLICADO ON-LINE 19/07/2022

PUBLICADO 10/01/2024

EDITOR ASSOCIADO
Gilberto Reynoso Meza

DOI: <http://dx.doi.org/10.18265/1517-0306a2022id6745>

ARTIGO ORIGINAL

Desenvolvimento de um modelo de simulação de um robô a quatro rodas utilizando ROS e Gazebo

 Gerberson Felix da Silva ^[1]
 Raphaell Maciel de Sousa ^{[2]*}
 Daniel Almeida dos Santos ^[3]
 Álefe Emanuel de Carvalho Azevedo ^[3]

[1] gerbersonfdasilva@gmail.com

[2] raphaell.sousa@ifpb.edu.br

[3] yopenbey934@gmail.com

[4] alefe.emanuel@academico.ifpb.edu.br

Grupo de Pesquisa em Controle, Automação e Robótica (GPCAR), Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB), Campus Cajazeiras, Brasil

RESUMO: O desenvolvimento de sistemas computacionais capazes de processar mais dados a custos cada vez menores tem estimulado o uso de *softwares* de modelagem tridimensional que permitem obter resultados gráficos cada vez mais realistas. Na robótica, simulações tridimensionais têm possibilitado o estudo de robôs de alto custo de aquisição, por meio de modelos computacionais que podem ser desenvolvidos em computadores pessoais, os quais são mais baratos e acessíveis. Nessa direção, o principal objetivo deste trabalho foi modelar computacionalmente um protótipo de robô móvel desenvolvido pelo Grupo de Pesquisa em Controle, Automação e Robótica (GPCAR) do IFPB – Campus Cajazeiras. O modelo computacional proposto pode ser utilizado tanto para testar algoritmos antes da implementação no robô real quanto para o ensino e a pesquisa em robótica móvel autônoma. A implementação computacional proposta neste trabalho foi realizada por meio do Autodesk Inventor e do Blender, em seguida, exportada para o ambiente de simulação do Gazebo. Além disso, o robô virtual foi integrado ao *Robot Operating System* (ROS). Resultados experimentais são apresentados para demonstrar o funcionamento do modelo computacional e sua integração com o protótipo desenvolvido, apontando a aproximação entre os resultados obtidos por meio do robô real e seu gêmeo digital.

Palavras-chave: construção sustentável; casa ecológica; educação; práticas experimentais; sustentabilidade.

Development of a simulation model of a four-wheel robot using ROS and Gazebo

ABSTRACT: The development of computer systems capable of processing more data at lower and lower costs has stimulated the use of three-dimensional modeling software that allows increasingly realistic graphic results. In robotics, three-dimensional simulations have enabled the study of high-cost robots through computational models that can be developed on personal computers, which are cheaper and more accessible. In this direction, the main objective of this work was to computationally model a mobile robot prototype developed

*Autor para correspondência.

by the Research Group in Control, Automation, and Robotics (GPCAR) from IFPB – Cajazeiras Campus. The proposed computational model can be used both to test algorithms before implementation in the real robot and for teaching and research purposes in autonomous mobile robotics. The computational implementation proposed in this paper was performed using Autodesk Inventor and Blender, and then exported to the Gazebo simulation environment. In addition, the virtual robot has been integrated into the Robot Operating System (ROS). The experimental results are presented to demonstrate the functioning of the computational model and its integration with the proposed prototype, pointing out the approximation between the results obtained using the real robot and its digital twin.

.....
Keywords: : digital twin; Gazebo; robot simulation; ROS.

1 Introdução

A impossibilidade do desenvolvimento de atividades presenciais em instituições de ensino em função da pandemia causada pela covid-19 impôs restrições de acesso a equipamentos laboratoriais. Consequentemente, isso também causou limitações no que se refere à aprendizagem baseada em experimentos práticos.

As atividades de ensino baseadas em experimentos proporcionam ao discente a possibilidade de estudar na prática fenômenos teóricos abordados em sala de aula ou vistos apenas nos livros didáticos. Desse modo, unir os estudos de teoria, de simulação e de prática é uma atividade importante que potencializa a consolidação da aprendizagem (Logar; Peklaj; Savec, 2017).

A robótica educacional tem sido um tipo de abordagem utilizada por vários autores como método de ensino capaz de potencializar as habilidades dos discentes. Na robótica, estimula-se a criatividade e a resolução de problemas integrando conceitos de mecânica, elétrica e programação (Nanou *et al.*, 2022).

Nessa direção, o robô Wally foi desenvolvido pelo Grupo de Pesquisa em Controle, Automação e Robótica (GPCAR) para o ensino e pesquisa da robótica móvel. O referido robô possui uma diversidade de sensores, tais como o LiDAR, e uma câmera estéreo e *encoders* nas quatro rodas; além de estar equipado com uma placa de processamento Nvidia Jetson TX2, um Arduino Due, um *power bank* para alimentação das placas de controle e para sensores e uma bateria de LiPO para alimentação dos motores. Com relação ao custo, Wally possui diversos sensores que não são comercializados no Brasil, portanto, sujeitos às flutuações de preço e limitações de tempo e taxas de importação. Portanto, o robô Wally consiste em uma ferramenta de alta tecnologia que pode ser usada para ensino, pesquisa e aplicações de robótica móvel autônoma.

Se o robô Wally é uma demonstração da prática, por outro lado a simulação computacional tem se tornado uma ferramenta estratégica em inúmeros campos, usada por muitos pesquisadores, visto que permite testar os algoritmos em ambientes virtuais, consequentemente, reduzindo o custo de erros e acelerando os processos de desenvolvimento. Porém, isso requer a modelagem adequada de um robô em um ambiente virtual (Moskvin *et al.*, 2020).

Com relação ao desenvolvimento de modelos de simulação computacional, diversas ferramentas computacionais estão disponíveis para aplicações específicas ou não da

robótica móvel. Entre essas ferramentas, destaca-se o *Robot Operating System* (ROS), que consiste em uma coleção de ferramentas e bibliotecas que acelera o desenvolvimento de protótipos de robôs móveis (Ahamed; Tewolde; Kwon, 2018), pois integra alguns *softwares* de simulação e ferramentas visuais, tais como o Gazebo e o RViz, que permitem ao usuário simular e monitorar seus robôs durante a execução de tarefas, entre outras possibilidades. Dessa forma, o ROS possibilita testar uma aplicação robótica diretamente do simulador Gazebo para os robôs reais, sem grandes alterações (Roa-Borbolla *et al.*, 2018). Logo, o ROS, junto ao Gazebo, fornece uma estrutura de simulação padrão com base em sua arquitetura modular, facilitando a integração de contribuições de outros pesquisadores (Zhang *et al.*, 2015).

Nessa direção, este trabalho apresenta o desenvolvimento de um modelo de simulação computacional (gêmeo digital) de um protótipo de robô móvel dotado de sensores e placas de processamento de dados, de modo que seja possível enviar e receber comandos do modelo computacional, tal qual ocorre com o protótipo real.

Tendo isso em vista, o restante deste trabalho está dividido da seguinte forma: na seção 2, destacam-se os trabalhos relacionados com a presente proposta; na seção 3, a metodologia utilizada no trabalho é apresentada em detalhes; e, por fim, os resultados experimentais e as conclusões são apresentados nas seções 4 e 5, respectivamente.

2 Referencial teórico

Nesta seção são apresentados o referencial teórico, com a introdução de alguns trabalhos relacionados à robótica móvel, bem como alguns conceitos importantes utilizados nesta pesquisa.

2.1 Robótica móvel

Atualmente, existem diversos tipos de robôs móveis interagindo com seres humanos e coexistindo na execução de tarefas do cotidiano. Cada vez mais se tem a tendência de que tais dispositivos móveis utilizem técnicas de controle inteligente para desempenhar tarefas sem o auxílio de seres humanos, de forma totalmente autônoma (Romero *et al.*, 2014). Desse modo, pode-se dizer que um robô móvel é um agente autônomo capaz de extrair informações do ambiente e utilizar esse conhecimento do mundo para deslocar-se com segurança de modo significativo e intencional, atuando e executando tarefas (Pio; Castro; Castro Júnior, 2006).

Inicialmente, os robôs foram utilizados para a automação de processos de produção industrial (Edwards, 1984). Logo mais, os robôs começaram também a ser utilizados para outros propósitos, tais como: brinquedos e entretenimento, medicina e cirurgia, e realização de tarefas em ambientes perigosos (Lima *et al.*, 2017; Watanabe *et al.*, 2021). Assim, a nova geração de robôs deve trabalhar de forma robusta, interagindo em ambientes complexos, imprevisíveis e realizando uma variedade de tarefas mais complexas que as de seus antecessores.

Para Wolf *et al.* (2009), um dos grandes desafios da robótica é como integrar as informações vindas de vários sensores, de modo a gerar comandos e controlar os diferentes dispositivos de atuação do robô, atuando de forma segura tanto para o robô quanto para aqueles que o cercam. Para isso, o desenvolvimento de pesquisas e aplicações de robótica móvel autônoma e inteligente, seja no Brasil ou no exterior, tem crescido de forma ampla e consistente nos últimos anos.

Siegwart, Nourbakhsh e Scaramuzza (2011) ressaltam que, embora os robôs móveis tenham um amplo conjunto de aplicações e mercados, um projeto que envolve a integração de diferentes áreas do conhecimento é um fato verdadeiro para praticamente todos os robôs móveis de sucesso, o que faz da robótica móvel um campo tão interdisciplinar quanto possível. Em um contexto educacional, Trentin *et al.* (2015) destacam o potencial da robótica como uma possível estratégia permanente e abrangente de aprendizagem interdisciplinar, possibilitando a interação entre várias áreas do conhecimento e em diferentes níveis de ensino, dos anos iniciais à pós-graduação.

2.2 Simulação de robôs

Segundo Megalingam *et al.* (2018), todos os dias vários robôs apresentam problemas durante uma série de testes, e uma das principais razões por trás disso é o hábito de testar diretamente o robô sem uma estimativa precisa de seu comportamento em uma plataforma de simulação. Outro fator levantado pelos mesmos autores são os controladores defeituosos, que podem dar comandos incorretos aos robôs, possibilitando a ocorrência de danos a eles mesmos e, se o comportamento for inesperado, pode acontecer algo ainda pior, como, por exemplo, danos aos operadores ou a terceiros.

Em muitas situações, no desenvolvimento de sistemas de controle robótico inteligente, faz-se necessário o uso de sistemas simulados, pois eles possibilitam projetar e ajustar o sistema até o ponto de ser implementado em robôs reais e posto em funcionamento na prática, conseqüentemente, simplificando o processo de verificação e validação de *software* (Kim; Lee; Lim, 2020).

Para Chikurtev (2020), um modelo de robô em um ambiente de simulação pode ser usado por muitas pessoas sem a necessidade de todos possuírem um robô físico, além de possibilitar uma série de experimentos e estudos que podem ser conduzidos de uma forma que não comprometa a segurança dos humanos, do robô e dos objetos com os quais o robô tem contato. Wolf *et al.* (2009) citam algumas vantagens acerca do uso de simulações de robôs, tais como economia de recursos financeiros e de tempo, prevenção de danos ao robô e de acidentes, e aumento da segurança e do aperfeiçoamento do *hardware* e do *software*. Já Choi *et al.* (2020) destacam a importância da simulação no projeto, controle e aprimoramento de desempenho de robôs, observando que, no futuro, os robôs estarão cada vez mais presentes no nosso cotidiano.

2.3 Robot operation system

O conceito de um sistema operacional de robô começou na Universidade de Stanford, quando alunos de pós-graduação notaram que seus colegas também perdiam bastante tempo em pesquisas refazendo algo que alguém já havia feito. Para eles, a falta de uma plataforma padrão frequentemente impedia que diferentes tipos de *softwares* robóticos se comunicassem uns com os outros, o que limitou a capacidade dos desenvolvedores de usar vários aplicativos em um único robô. Isso também impediu que outros pesquisadores pudessem reutilizar o *software* e colaborar de outra forma.

Visto isso, o ROS elimina a necessidade de criar programas do zero para cada projeto, o que reduz o tempo e os custos de desenvolvimento de aplicativos do robô. Não só isso, o uso e a modificação constantes de aplicativos por desenvolvedores levam a melhorias contínuas.

Com a adoção pelos usuários facilitada pelo seu código aberto (Garber, 2013), o ROS é um *framework* que hoje em dia é amplamente aceito e usado na comunidade de robótica. Seu objetivo principal é facilitar o desenvolvimento e o compartilhamento de vários componentes de um sistema robótico (Mahtani *et al.*, 2016). Por isso, fornece bibliotecas e ferramentas para ajudar os desenvolvedores de *software* a criarem algoritmos robóticos, como abstração de hardware, *drivers* de dispositivos, bibliotecas, simuladores, gerenciamento de pacotes e muito mais (Zhi; Xuesong, 2018).

Esse *framework* usa o conceito de pacotes, nós, tópicos, mensagens e serviços. Um nó é um programa executável que adquire dados dos sensores do robô e os passa para outros nós, sendo a informação que se move de um nó para outro chamada de mensagem. As mensagens sempre viajam por portas especiais chamadas de tópicos. Um nó que envia mensagens em um tópico é chamado de *publisher* e o nó receptor precisa se inscrever no tópico para receber essa mensagem, por isso, ele é chamado de *subscriber*. Todos os nós relacionados são combinados em um pacote que pode ser facilmente compilado e transferível para outros computadores (Zaman; Slany; Steinbauer, 2011). Então, cada pacote no ROS é claramente rotulado com seus termos de licenciamento para que os pesquisadores e desenvolvedores possam saber imediatamente quais componentes podem ser incorporados em seu trabalho (Cousins, 2010).

Para Quigley *et al.* (2009), o ROS foi projetado para apoiar a filosofia de desenvolvimento de *software* modular baseado em ferramentas. Seus idealizadores esperavam que seu projeto aberto poderia ser estendido e desenvolvido por outras pessoas, para que criassem sistemas de *software* de robô passíveis de serem úteis para uma variedade de plataformas de *hardware*, tipos de pesquisas e requisitos de tempo de execução. O ecossistema ROS agora consiste em dezenas de milhares de usuários em todo o mundo, trabalhando em domínios que vão desde projetos de passatempo a grandes sistemas de automação industrial (Zhang *et al.*, 2017).

2.4 Gazebo

O uso de um simulador visual é útil nos casos em que se deseja testar o funcionamento de determinado algoritmo em desenvolvimento. No Gazebo, não apenas o robô, mas também um mundo ou terreno no qual o ele deve ser implantado pode ser criado (Megalingam *et al.*, 2018). Desse modo, o Gazebo é um simulador que nos permite testar rapidamente algoritmos, projetar robôs, realizar testes e treinar sistemas de inteligência artificial usando cenários realistas.

O Gazebo pode ser usado para simular robôs com parâmetros reais, de modo que o comportamento do robô nessas situações possa ser emulado e antecipado. O *hardware* simulado do Gazebo é projetado para refletir o comportamento de seu equivalente na realidade (Kumar *et al.*, 2020). Em sua biblioteca, o Gazebo dispõe de robôs que são comumente usados pela comunidade; além disso, é possível modelar o robô em *software* CAD, tais como o Autodesk Inventor, e em seguida, pode-se importar o modelo desenvolvido para o Gazebo por meio de um arquivo *Unified Robot Description Format* (URDF), que possui a descrição dos sensores, *links* e juntas. Todos os objetos simulados têm massa, velocidade, fricção e vários outros atributos que permitem que se comportem de forma realista quando empurrados, puxados, derrubados ou carregados. Assim sendo, quase todos os aspectos da simulação são controláveis, desde as condições de iluminação até os coeficientes de atrito (Koenig; Howard, 2004).

Logo, o Gazebo é capaz de simular com precisão e eficiência diversos robôs em ambientes internos e externos complexos. Além de ser gratuito e possuir uma comunidade

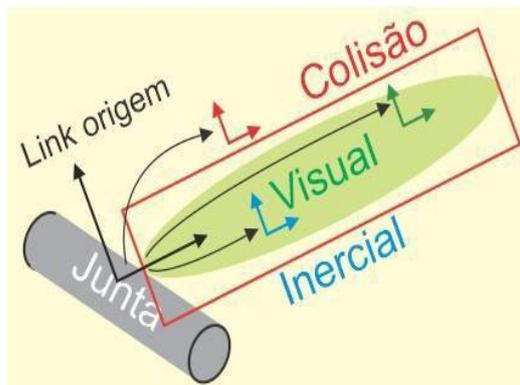
ativa, um outro aspecto relevante é que o usuário pode desenvolver uma aplicação diretamente no ambiente de simulação e depois implantar no robô físico com poucas ou nenhuma alteração no código (Cheng *et al.*, 2022).

2.5 URDF

O URDF é um formato XML que descreve todos os elementos de um robô (suas partes, sensores, juntas, dimensões, etc.). Apenas estruturas em árvore podem ser representadas, descartando todos os robôs paralelos. Além disso, o formato assume que o robô consiste em *links* rígidos conectados por juntas, bem como que os elementos flexíveis não são suportados (Peters, 2022).

Conforme pode ser observado na Figura 1, o *link* URDF é dividido em nome (que define o *link* em si), inercial, visual e colisão.

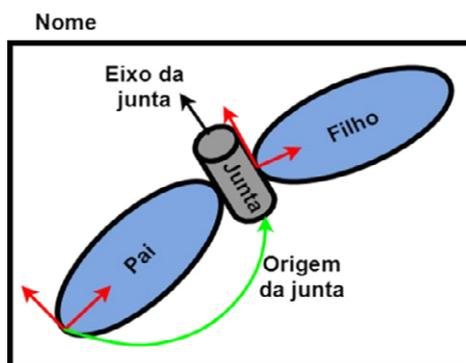
Figura 1 ▶
Link de um arquivo URDF. Fonte: adaptada de Peters (2022)



Na propriedade de inércia do *link*, é definida a origem em xyz e rpy (com ângulos em radianos), a massa em kg, e a inércia por meio do ixx , ixy , ixz , iyy , iyz e izz , que são elementos de uma matriz de inércia. Na propriedade visual do *link*, é definida a origem em xyz e rpy (com ângulos em radianos) e a geometria do objeto, que pode ser no formato de caixa, cilindro, esfera ou de malha, por meio dos formatos Collada (.dae) ou StereoLithography (.stl). Também pode ser definido o material, incluindo sua cor e textura. Na propriedade de colisão do *link* é definida a origem e a geometria, assim como na propriedade visual.

Conforme é observado na Figura 2, a junta URDF é dividida em nome (que define a própria junta) e em tipo (que define o tipo de junta).

Figura 2 ▶
Junta de um arquivo URDF. Fonte: elaborada pelos autores (2021)



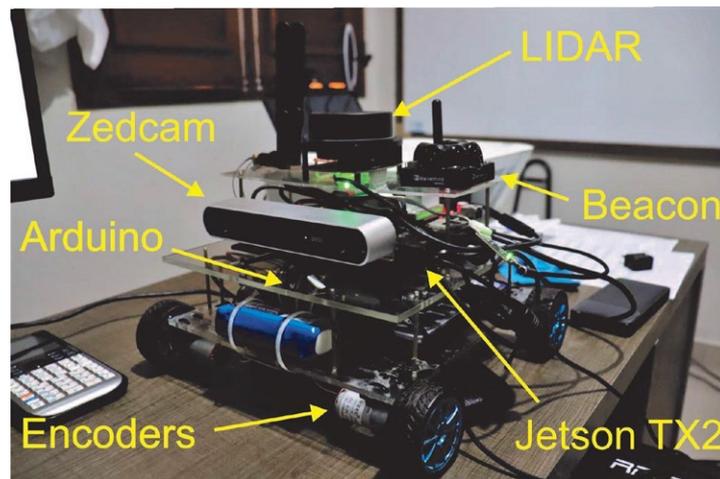
Na propriedade de tipo de junta, ela pode ser definida como rotacionária, contínua, prismática, fixa, flutuante ou plana. Nesses tipos, podem ser especificados a origem, o *link* pai, o *link* filho, o eixo, a calibração, a dinâmica, os limites (inferior, superior, de esforço ou velocidade) e os controladores de segurança, que especificam limites opcionais de posição e velocidade da junta.

Existem dois métodos para construir o modelo URDF. No primeiro, escreve-se diretamente o arquivo XML, ou seja, os *links* podem ser desenhados como gráficos fundamentais ou gráficos introduzidos salvos como um arquivo binário desenhado por CAD. E o outro método é por meio de um *plugin* no *software* Solidworks (Zhang *et al.*, 2018).

3 Metodologia

O objetivo deste trabalho é desenvolver um modelo computacional (gêmeo digital) do robô Wally (Figura 3), desenvolvido pelo GPCAR (Grupo de Pesquisa em Controle, Automação e Robótica). O referido modelo pode ser utilizado para o ensino e pesquisas relacionadas com robótica móvel, visto que consiste em um modelo computacional com praticamente todos os recursos de sensores e atuadores presentes no modelo real.

Figura 3 ►
Robô Wally
desenvolvido. Fonte:
arquivo dos autores
(2021)



Para o desenvolvimento desta proposta, foram escolhidos alguns *softwares* de modelagem 3D que serão apresentados nesta sessão. Além disso, os passos para a integração do modelo computacional com o protótipo real também são aqui descritos.

Neste trabalho, o modelo tridimensional foi desenvolvido por meio do Autodesk Inventor (versão educacional), porém, outras ferramentas de modelagem 3D – como Fusion 360, SolidWorks e FreeCAD – podem ser utilizadas. Então, após a modelagem realizada no Autodesk Inventor, usou-se o Blender, visto que é um *software* livre. Além do referido programa de computador possibilitar colorir o modelo computacional desenvolvido previamente, é possível aplicar com ele texturas, reduzir a complexidade das malhas do modelo, e, conseqüentemente, reduzir o tamanho do arquivo, que, por fim, foi exportado em Collada (.dae), de modo que seja possível visualizar as cores definidas no Blender.

Depois da etapa do Blender, consultou-se a documentação disponível para a criação do arquivo URDF, além de ser realizada a escolha dos plugins para o Gazebo. O Gazebo foi escolhido para ser utilizado devido a sua documentação ser acessível, além de sua popularidade e por ser desenvolvido pela mesma organização que criou e mantém o ROS, não havendo, portanto, problemas de compatibilidade.

4 Resultados

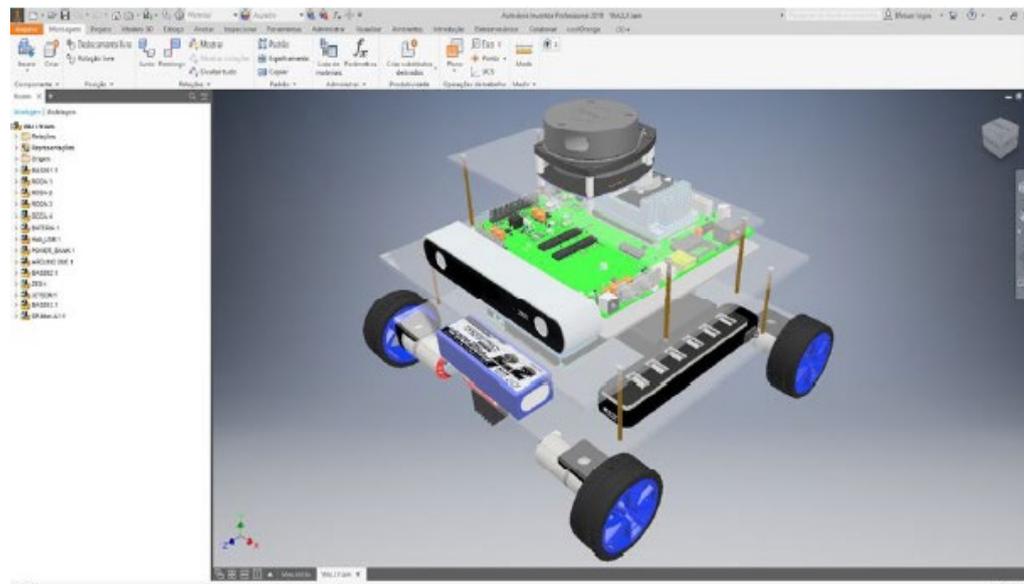
Nesta seção são apresentados os resultados obtidos utilizando o robô Wally, com a utilização de um *notebook* Dell Inspiron 14 3437, com processador Intel Core i5-4200U, 8 GB de memória RAM e o simulador Gazebo, por meio das ferramentas do ROS de visualização e depuração, além do modelo computacional desenvolvido. Os códigos desenvolvidos podem ser encontrados em um repositório dos autores^[1].

[1] https://github.com/GerbersonFelix/wally_description.

4.1 Resultados experimentais

Na Figura 4, é possível observar o modelo computacional inicial desenvolvido no Autodesk Inventor, no qual algumas das peças foram modeladas por meio de medições do robô real, e outras foram obtidas em um repositório de peças chamado Grabcad, como é o caso da placa do modelo digital da NVIDIA Jetson TX2.

Figura 4 ▶
Robô Wally modelado no Autodesk Inventor.
Fonte: arquivo dos autores (2021)



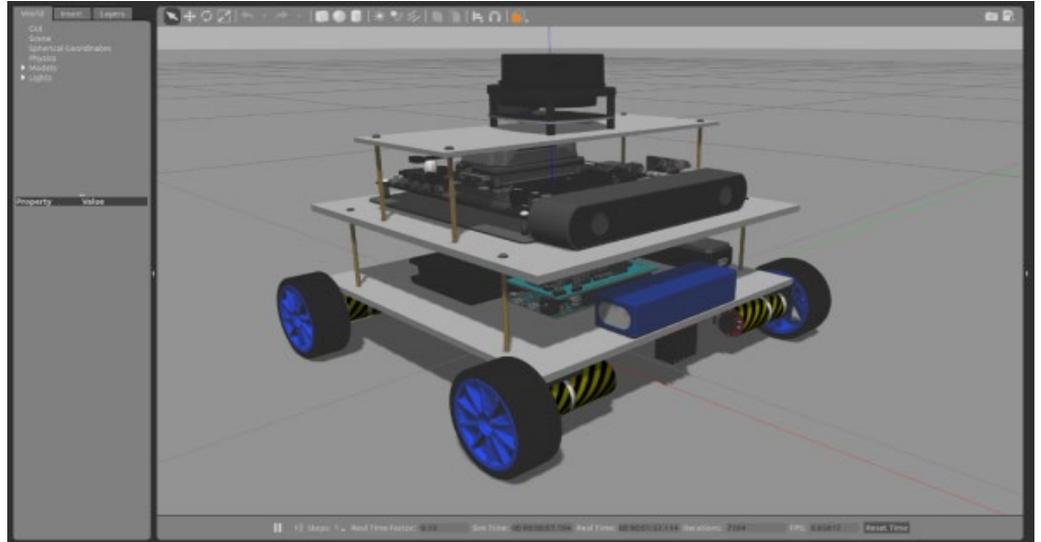
Após a modelagem inicial, iniciou-se a aplicação de coloração e texturas, de modo que o robô possa ter uma aparência mais realista. Na Figura 5, pode ser observado o modelo computacional após a etapa de coloração e textura realizada no Blender. O modelo computacional da Figura 5 já apresenta uma aparência realista, com texturas próximas ao protótipo real. O arquivo gerado no Blender será exportado e carregado dentro do ambiente de simulação (Gazebo).

Figura 5 ▶
Robô Wally modelado
no Blender
. Fonte: arquivo dos
autores (2021)



Na Figura 6, pode ser observado o gêmeo digital proposto neste trabalho, já inserido dentro do ambiente Gazebo. O Gazebo faz a interpretação do arquivo URDF desenvolvido previamente, sendo o resultado um modelo bem realista, com todas as cores e texturas aplicadas nos passos anteriores. Em relação ao acionamento dos motores e à comunicação com os sensores, cada elemento sensor/atuador está encapsulado dentro do arquivo URDF, de modo que já é possível publicar e ler dados por meio do ROS.

Figura 6 ▶
Robô Wally modelado
no Gazebo
. Fonte: arquivo dos
autores (2021)



Após ser carregado dentro do ambiente de simulação (Gazebo), alguns testes foram realizados para validação da comunicação com sensores e atuadores. Na Figura 7, o Wally encontra-se em uma situação em que está posto entre obstáculos (parte mais escura da imagem). Nesse teste, pretendeu-se verificar o funcionamento do LiDAR, bem como o acesso às câmeras do robô. Na Figura 8, é possível notar que as câmeras conseguem mostrar os referidos obstáculos (imagem obtida por meio do RViz); além disso, observa-se em rosa e vermelho os dados do LiDAR com os obstáculos identificados. Esses dados correspondem à nuvem de pontos construída a partir da indicação de cada feixe de *laser* do sensor.

Figura 7 ▶
Wally no ambiente
de simulação.
Fonte: arquivo dos
autores (2021)

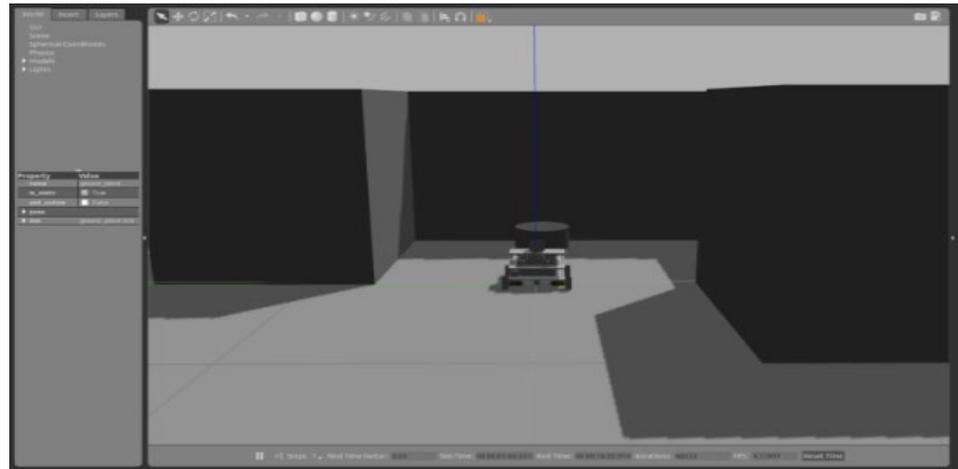


Figura 8 ▶
Visualização do modelo
do Wally com dados da
simulação.
Fonte: arquivo dos
autores (2021)

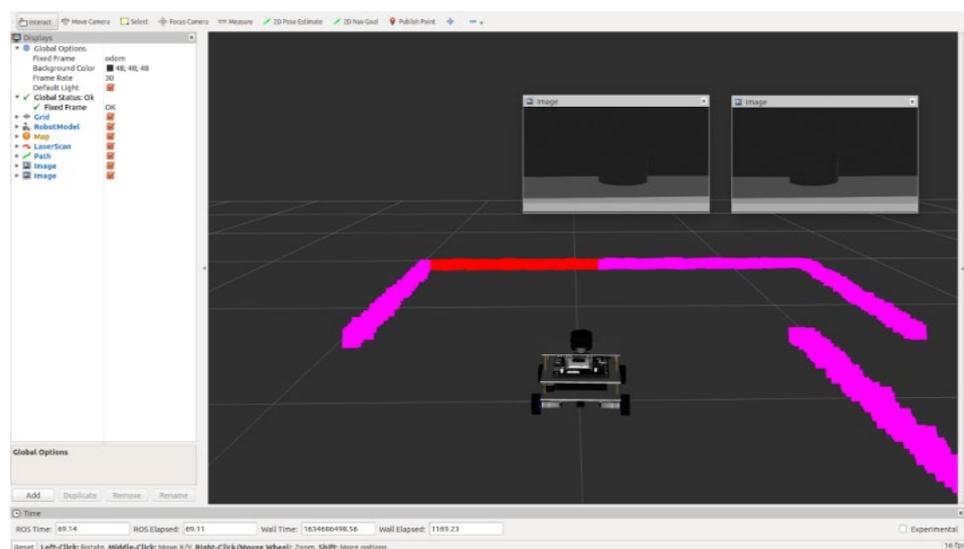
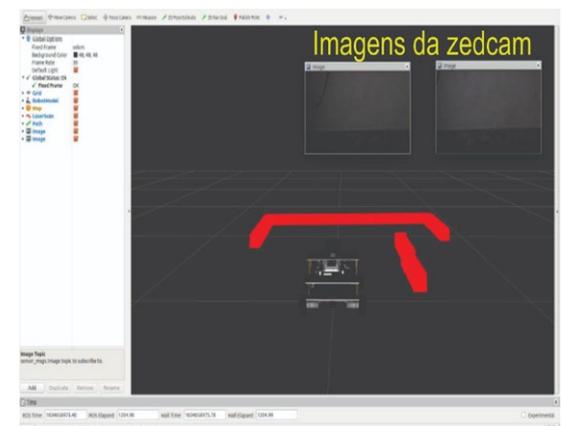


Figura 9 ▼
Wally com sensores
ligados (a) e
visualização do modelo
do Wally com os dados
dos sensores (b).
Fonte: arquivo dos
autores (2021)

De modo similar, os experimentos com obstáculos (Figuras 7 e 8) foram reproduzidos no robô real (Figuras 9a). Na Figura 9a pode-se observar o robô Wally com seus sensores ligados, com suas respectivas leituras (Figura 9b) por meio do RViz, que está sendo executado no *notebook*. Na Figura 9b, observa-se pelas imagens de cada lado da câmera estéreo, bem como pelas superfícies detectadas ao redor do robô pelo LiDAR, que há uma integração entre o gêmeo digital e o protótipo real, por meio do acesso aos sensores reais.



(a)



(b)

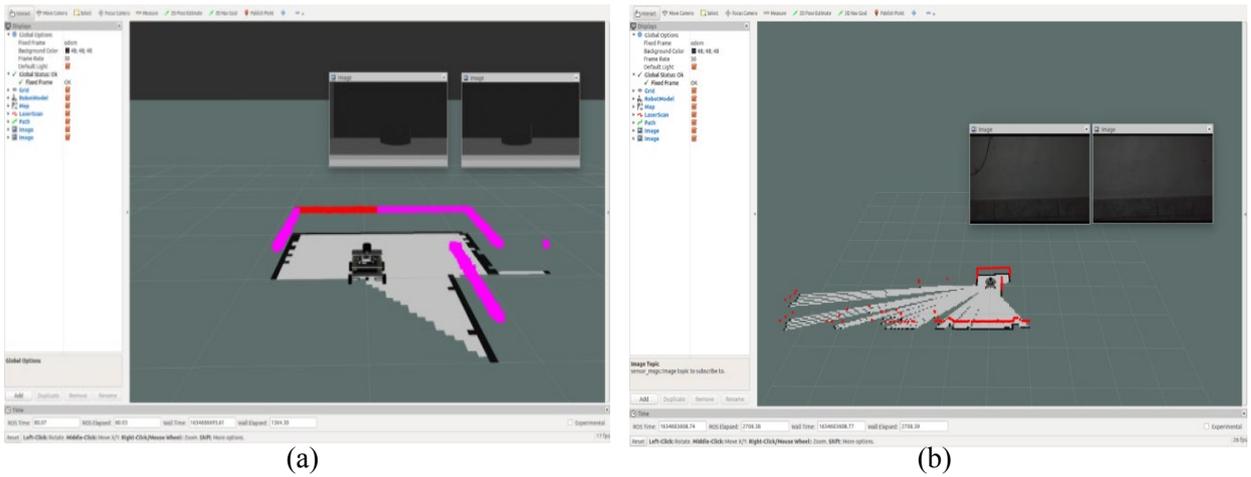
Figura 10 ▼

Visualização do RViz de um processo de mapeamento.

(a) Ambiente de simulação. (b) Com o robô experimental.

Fonte: arquivo dos autores (2021)

Nas Figuras 10 e 11 são apresentados os resultados experimentais para o processo de mapeamento. Na Figura 10a, observa-se um tom de cinza mais claro, que diz respeito ao mapa que está sendo gerado pelo pacote *GMapping* do ROS. O mesmo pode ser visualizado no robô experimental, na Figura 10b.



Nas Figuras 11a e 11b são apresentados os resultados dos mapas obtidos pelo modelo de simulação e pelo robô real, respectivamente.

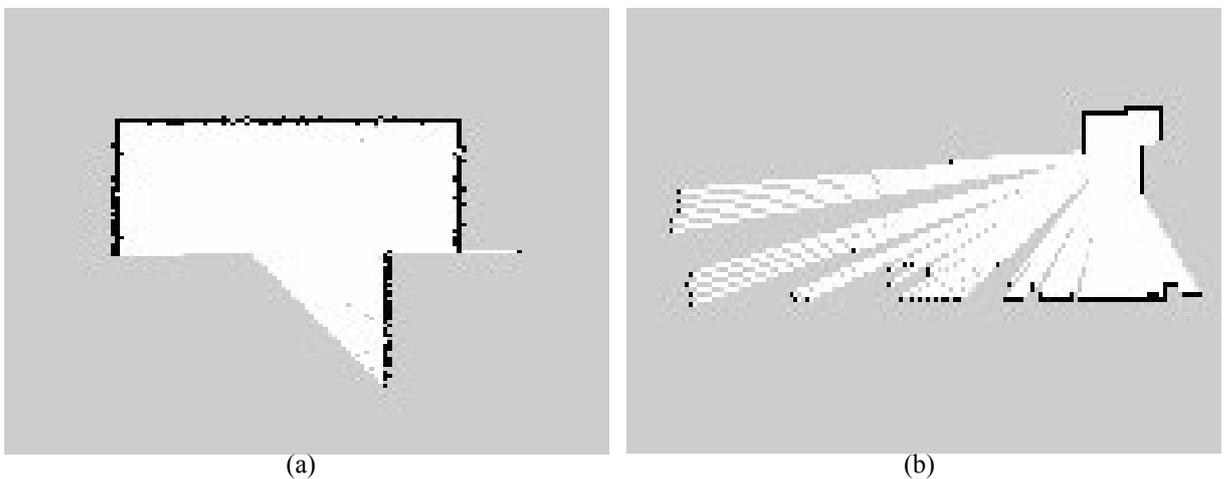


Figura 11 ▲

Mapa gerado por meio do *GMapping*.

(a) Simulação. (b) Robô experimental.

Fonte: dados da pesquisa (2021)

A partir dos resultados obtidos, pode-se observar que o modelo de simulação computacional desenvolvido pode ser utilizado em diversas aplicações no contexto de ensino e pesquisa da robótica móvel. Diversos sensores são utilizados no protótipo real (ver Figura 3), de modo que, para instituições de ensino ou grupos de pesquisa, pode ser difícil a aquisição e confecção de um protótipo similar. O desenvolvimento do modelo de simulação possibilita que o protótipo seja testado e até incorporado em diferentes trabalhos ou cursos de robótica móvel, permitindo o estudo de técnicas de mapeamento, planejamento de trajetórias, visão computacional, controle, entre outros métodos.

5 Conclusões

Este trabalho apresentou o desenvolvimento de um modelo computacional (gêmeo digital) de um robô móvel autônomo baseado em um protótipo real desenvolvido pelo GPCAR. Foram realizados diversos testes com o modelo de simulação e o protótipo. A partir dos resultados apresentados, pode-se constatar que os resultados obtidos por meio do robô real se aproximam dos resultados de simulação.

Com relação ao uso do ROS, algumas soluções prontas colaboraram para o desenvolvimento do modelo computacional, o que fez reduzir significativamente o tempo de produção do referido modelo, coleta e análise dos resultados.

As sugestões para trabalhos futuros são: adicionar outros tipos de sensores ao modelo computacional e utilizar outros pacotes do ROS para robótica autônoma, tais como o *navigation stack*.

Financiamento

Este trabalho foi financiado pelos editais de pesquisa do IFPB, com recursos do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ), e pelo programa de concessão de hardware acadêmico da NVIDIA (*NVIDIA Academic Hardware Grant Program*).

Conflito de interesses

Os autores declaram não haver conflito de interesses.

Referências

AHAMED, M. F. S; TEWOLDE, G.; KWON, J. Software-in-the-loop modeling and simulation framework for autonomous vehicles. *In: 2018 IEEE INTERNATIONAL CONFERENCE ON ELECTRO-INFORMATION TECHNOLOGY (EIT), 2018, Rochester. Proceedings [...]. Rochester: IEEE, 2018. p. 305-310. DOI: <https://dx.doi.org/10.1109/EIT.2018.8500101>.*

CHENG, J.; ZHU, L.; CAI, X.; WU, H. Mapping and path planning simulation of mobile robot slam based on ROS. *In: 2022 INTERNATIONAL SEMINAR ON COMPUTER SCIENCE AND ENGINEERING TECHNOLOGY (SCSET), 2022, Indianapolis. Proceedings [...]. Indianapolis: IEEE, 2022. p. 10-14. DOI: <https://doi.org/10.1109/SCSET55041.2022.00012>.*

CHIKURTEV, D. Mobile Robot Simulation and Navigation in ROS and Gazebo. *In: 2020 INTERNATIONAL CONFERENCE AUTOMATICS AND INFORMATICS (ICAI), 2020, Varna. Proceedings [...]. Varna: IEEE, 2020. p. 1-6. DOI: <https://doi.org/10.1109/ICAI50593.2020.9311330>.*

CHOI, H.; CRUMP, C.; DURIEZ, C.; ELMQUIST, A.; HAGER, G.; HAN, D.; HEARL, F.; HODGINS, J.; JAIN, A.; LEVE, F.; LI, C.; MEIER, F.; NEGRUT, D.; RIGHETTI,

L.; RODRIGUEZ, A.; TAN, J.; TRINKLE, J. On the use of simulation in robotics: opportunities, challenges, and suggestions for moving forward. *PNAS*, v. 118, n. 1, e1907856118, 2020. DOI: <https://doi.org/10.1073/pnas.1907856118>.

COUSINS, S. ROS on the PR2 [ROS Topics]. *IEEE Robotics & Automation Magazine*, v. 17, n. 3, p. 23-25, 2010. DOI: <https://doi.org/10.1109/MRA.2010.938502>.

EDWARDS, M. Robots in industry: An overview. *Applied Ergonomics*, v. 15, n. 1, p. 45-53, 1984. DOI: [https://doi.org/10.1016/S0003-6870\(84\)90121-2](https://doi.org/10.1016/S0003-6870(84)90121-2).

GARBER, L. Robot OS: a new day for robot design. *Computer*, v. 46, n. 12, p. 16-20, 2013. DOI: <https://doi.org/10.1109/MC.2013.434>.

KIM, Y.; LEE, S.-Y.; LIM, S. Implementation of PLC controller connected Gazebo-ROS to support IEC 61131-3. *In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETFA), 25., 2020, Vienna. Proceedings [...].* Vienna: IEEE, 2020. p. 1195-1198. DOI: <https://dx.doi.org/10.1109/ETFA46521.2020.9212096>.

KOENIG, N.; HOWARD, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. *In: 2004 IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS (IROS), 2005, Sendai. Proceedings [...].* Sendai: IEEE, 2004. p. 2149-2154. DOI: <https://doi.org/10.1109/IROS.2004.1389727>.

KUMAR, K.; AZID, S. I.; FAGIOLINI, A.; CIRRINCIONE, M. Erle-copter simulation using ROS and Gazebo. *In: IEEE MEDITERRANEAN ELECTROTECHNICAL CONFERENCE (MELECON), 20., 2020, Palermo. Proceedings [...].* Palermo: IEEE, 2020. p. 259-263. DOI: <https://doi.org/10.1109/MELECON48756.2020.9140476>.

LIMA, A. S.; RIBEIRO, S. A.; ALMEIDA, L. F.; FUSCHILO, C. Um sistema de comunicação via socket em uma rede wi-fi para controle de um robô de inspeção. *Holos*, v. 2, p. 424-435, 2017. DOI: <https://doi.org/10.15628/holos.2017.5737>.

LOGAR, A.; PEKLAJ, C.; SAVEC, V. F. Effectiveness of student learning during experimental work in primary school. *Acta Chimica Slovenica*, v. 64, n. 3, p. 661-671, 2017. DOI: <http://dx.doi.org/10.17344/acsi.2017.3544>.

MAHTANI, A.; SÁNCHEZ, L.; FERNÁNDEZ, E.; MARTINEZ, A. **Effective robotics programming with ROS**. 3. ed. Birmingham: Packt Publishing, 2016.

MEGALINGAM, R. K.; NAGALLA, D.; PASUMARTHI, R. K.; GONTU, V.; ALLADA, P. K. ROS based, simulation and control of a wheeled robot using gamer's steering wheel. *In: INTERNATIONAL CONFERENCE ON COMPUTING COMMUNICATION AND AUTOMATION (ICCCA), 4., 2018, Greater Noida. Proceedings [...].* Greater Noida: IEEE, 2018. p. 1-5. DOI: <https://doi.org/10.1109/CCAA.2018.8777569>.

MOSKVIN, I.; LAVRENOV, R.; MAGID, E.; SVININ, M. Modelling a crawler robot using wheels as pseudo-tracks: model complexity vs performance. *In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL ENGINEERING AND APPLICATIONS (ICIEA), 7., 2020, Bangkok. Proceedings [...].* Bangkok: IEEE, 2020. p. 1-5. DOI: <https://dx.doi.org/10.1109/ICIEA49774.2020.9102110>.

NANOU, A.; TSIOMI, E.; OIKONOMOU, A.; KARAMPATZAKIS, D. The SAS strategy training for children with ASD in inclusive educational robotics activities. **Education. Innovation. Diversity (EID)**, v. 2, n. 3, p. 34-52, 2022. DOI: <http://dx.doi.org/10.17770/eid2021.2.6723>.

PETERS, S. **urdf/XML/link**. 2022. Disponível em: <https://wiki.ros.org/urdf/XML/link>. Acesso em: 16 jan. 2024.

PIO, J. L. S.; CASTRO, T. H. C.; CASTRO JÚNIOR, A. N. A robótica móvel como instrumento de apoio à aprendizagem de Computação. *In*: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE), 17., 2006, Brasília. **Anais [...]**. Brasília: SBIE, 2006. p. 497-506. Disponível em: <http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/510/496>. Acesso em: 16 jan. 2024.

QUIGLEY, M.; GERKEY, B.; CONLEY, K.; FAUST, J.; FOOTE, T.; LEIBS, J.; BERGER, E.; WHEELER, R.; NG, A. ROS: an open-source Robot Operating System. *In*: ICRA WORKSHOP ON OPEN SOURCE SOFTWARE (ICRA), 2009, Kobe. Disponível em: <http://robotics.stanford.edu/~ang/papers/icraoss09-ROS.pdf>. Acesso em: 19 jun. 2022.

ROA-BORBOLLA, A. G.; MARIN-HERNANDEZ, A.; HERNANDEZ-BELMONTE, U. H.; AYALA-RAMIREZ, V.; ROA-F., K. Realistic and automatic map generator for mobile robots. *In*: 2018 INTERNATIONAL CONFERENCE ON MECHATRONICS, ELECTRONICS AND AUTOMOTIVE ENGINEERING (ICMEAE), 2018, Cuernavaca. **Proceedings [...]**. Cuernavaca: IEEE, 2018. p. 50-55. DOI: <https://doi.org/10.1109/ICMEAE.2018.00016>.

ROMERO, F. A. R.; PRESTES, E.; OSÓRIO, F.; WOLF, D. **Robótica móvel**. 1. ed. Rio de Janeiro: LTC, 2014. v. 1.

SIEGWART, R.; NOURBAKHSI, I. R.; SCARAMUZZA, D. **Introduction to autonomous mobile robots**. 2. ed. Cambridge: MIT Press, 2011.

TRENTIN, M. A. S.; ROSA, C. W.; ROSA, A. B.; TEIXEIRA, A. C. Robótica educativa livre no ensino de Física: da construção do robô à elaboração da proposta didática de orientação metacognitiva. **Revista Brasileira de Ensino de Ciência e Tecnologia**, v. 8, n. 3, p. 274-292, 2015. DOI: <http://dx.doi.org/10.3895/rbect.v8n3.1950>.

WATANABE, H.; TANZAWA, T.; KOTANI, S.; SHIMIZU, T. Remote-controlled rust assessment system for weathering steel. *In*: 2021 IEEE INTERNATIONAL CONFERENCE ON MECHATRONICS (ICM), 2021, Kashiwa. **Proceedings [...]**. Kashiwa: IEEE, 2021. p. 1-6. DOI: <https://doi.org/10.1109/ICM46511.2021.9385643>.

WOLF, D. F.; SIMÕES, E. V.; OSÓRIO, F. S.; TRINDADE, O. J. Robótica móvel inteligente: da simulação às aplicações no mundo real. *In*: CARVALHO, A. C. P. L. F.; KOWALTOWSKI, T. **Atualizações em informática**. Rio de Janeiro: PUC-Rio, 2009. Disponível em: http://osorio.wait4.org/publications/2009/CL_JAI2009_Completo.pdf. Acesso em: 19 jun. 2022.

ZAMAN, S.; SLANY, W.; STEINBAUER, G. ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues. *In*: 2011 SAUDI INTERNATIONAL ELECTRONICS, COMMUNICATIONS AND PHOTONICS

CONFERENCE (SIEPC), 2011, Riyadh. **Proceedings** [...]. Riyadh: IEEE, 2011. p. 1-5. DOI: <https://dx.doi.org/10.1109/SIEPC.2011.5876943>.

ZHANG, L.; MERRIFIELD, R.; DEGUET, A.; YANG, G.-Z. Powering the world's robots: 10 years of ROS. **Science Robotics**, v. 2, n. 11, 2017. DOI: <https://doi.org/10.1126/scirobotics.aar1868>.

ZHANG, M.; QIN, H.; LAN, M.; LIN, J.; WANG, S.; LIU, K.; LIN, F.; CHEN, B. M. A high fidelity simulator for a quadrotor UAV using ROS and Gazebo. *In*: ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON 2015), 41., 2015, Yokohama. **Proceedings** [...]. Yokohama: IEEE, 2015. p. 2846-2851. DOI: <https://dx.doi.org/10.1109/IECON.2015.7392534>.

ZHANG, Y.; LU, Z.; WANG, C.; LIU, C.; WANG, Y. Voice control dual arm robot based on ROS system. *In*: 2018 IEEE INTERNATIONAL CONFERENCE ON INTELLIGENCE AND SAFETY FOR ROBOTICS (ISR), 2018, Shenyang. **Proceedings** [...]. Shenyang: IEEE, 2018. p. 232-237. DOI: <https://dx.doi.org/10.1109/IISR.2018.8535942>.

ZHI, L.; XUESONG, M. Navigation and control system of mobile robot based on ROS. *In*: IEEE ADVANCED INFORMATION TECHNOLOGY, ELECTRONIC AND AUTOMATION CONTROL CONFERENCE (IAEAC), 3., 2018, Chongqing. **Proceedings** [...]. Chongqing: IEEE, 2018. p. 368-372. DOI: <https://doi.org/10.1109/IAEAC.2018.8577901>.