

# Mapeamento e Localização para o kit Robótico RoboDeck

Matheus Chaves Menezes<sup>1</sup>, Chrystian Gustavo Martins Nascimento<sup>1</sup>,  
Alexandre C. M. de Oliveira<sup>2</sup>

<sup>1</sup>Curso de Ciência da Computação – Universidade Federal do Maranhão (UFMA)  
São Luís - Brasil

<sup>2</sup>Departamento de Informática – Universidade Federal do Maranhão (UFMA)  
São Luís - Brasil

{matheuschavesmenezes, chrystiang20}@gmail.com, acmo@deinf.ufma.br

**Resumo.** *Dois fatores são determinantes para o sucesso da exploração robótica autônoma: sistema de localização e construção do mapa para navegação. Este trabalho apresenta um estudo sobre a plataforma robótica RoboDeck, em especial, no tocante às rotinas computacionais específicas necessárias para sua utilização na tarefa de exploração autônoma, com suporte do Sistema Operacional Robótico (Robot Operating System – ROS) e sensores de baixo custo. Experimentos são apresentados, nos quais mapas de um ambiente fechado e controlado são gerados satisfatoriamente.*

**Palavras-chaves:** *Robótica Móvel. Mapeamento. Localização. Sensores. ROS.*

**Abstract.** *Two factors determine the success of the autonomous robotic exploration: tracking system and construction of the map for navigation. This paper presents a study on the robotic platform RoboDeck, in particular with respect to specific computational routines necessary for use in autonomous exploration task, with support of the Robotic Operating System (Robot Operating System - ROS) and low cost sensors. Experiments are presented, which maps of a closed and controlled environment are satisfactorily generated.*

**Keywords:** *Mobile Robots. Mapping. Location Estimation. Sensors. ROS.*

## 1. Introdução

A construção de mapas possibilita a navegação autônoma a um robô móvel, bem como a exploração de ambientes desconhecidos, inexplorados ou de difícil acesso para seres humanos, seguindo objetivos pré-definidos. Durante o processo de mapeamento, informações necessárias sobre a localização, como posição e rotação do robô, são obtidas através de sistemas odométricos instalados no equipamento. A detecção de obstáculos ao longo da construção do mapa pode ser realizada por diversos sensores, desde sensores ultrassônicos de baixo custo e precisão [1] [2], aos mais avançados, como os *laser scanners* [3] [4]. *Encoders* são muito utilizados para extração desses dados de sensores, porém, alguns fatores contribuem para o acúmulo de erros da posição e orientação do robô ao longo do tempo.

Neste trabalho, a plataforma robótica RoboDeck, fabricada pela empresa nacional XBOT, é alvo de estudo visando empregá-la para a tarefa de exploração autônoma. O RoboDeck, projetado para promover o desenvolvimento educacional, possui diversos sensores, como acelerômetro, bússola, *encoders*, câmera digital, sensores infravermelhos e

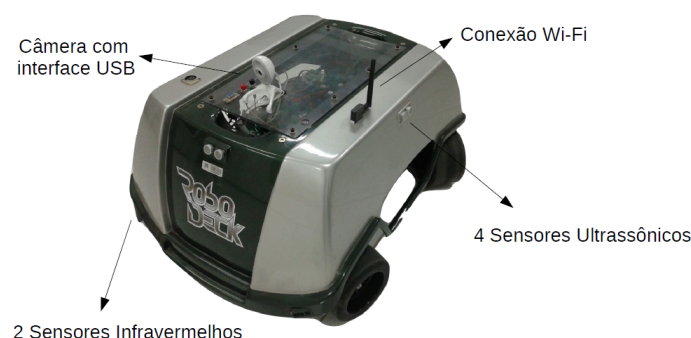
ultrassônicos [5]. O estudo da plataforma inclui o desenvolvimento de rotinas específicas para odometria e movimento, incluindo rotação e captura de dados dos sensores, com suporte do *framework* ROS (*Robot Operating System*), este último idealizado para facilitar no gerenciamento e na escrita de algoritmos robóticos, tendo compatibilidade com vários robôs no mercado [6]. Experimentos são apresentados, nos quais mapas de um ambiente fechado e controlado são gerados satisfatoriamente, como primeiro passo antes do desenvolvimento de novas rotinas voltadas para dar autonomia ao RoboDeck na exploração de ambientes desconhecidos.

Este artigo está dividido em seções. A Seção 2 apresenta os conceitos utilizados para realização deste trabalho, bem como a apresentação do robô, ferramentas e sensores. Na seção 3, explica-se as abordagens usadas para cumprir com a proposta de mapeamento e localização. Na Seção 4, são apresentadas as conclusões do trabalho.

## 2. Materiais e Métodos

O RoboDeck é uma plataforma robótica educacional de código aberto produzido pela empresa nacional Xbot [5]. O pacote educacional conta ainda com um SDK (*Software Development kit*) para desenvolvimentos de aplicações, um ambiente de programação baseado em linguagem C/C++ e um *software* para testes. O robô, ilustrado na Figura 1, possui quatro sensores ultrassônicos distribuídos no centro de cada lado. Além disso, o robô possui infravermelho, câmera USB, acelerômetro, bússola, *encoders* nos motores de tração, GPS e comunicação via *WiFi* ou *ZigBee*. Cada *encoder* possui resolução de pulsos que variam de 0 a 400 por uma volta completa da roda, ou seja, uma volta completa da roda é igual a 400 pulsos (resolução) [7].

Figura 1. Visão geral do robô RoboDeck

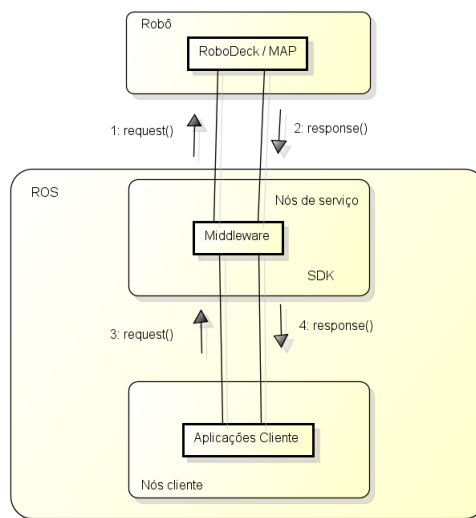


ROS é um *framework* de código aberto utilizado no gerenciamento de sistemas robóticos, fornecendo serviços de um sistema operacional, incluindo abstração de hardware, controle de dispositivos de baixo nível, e implementação de funcionalidades usualmente utilizadas, como troca de mensagens entre processos, gerenciamento de pacotes e biblioteca para escrita e execução de códigos para vários dispositivos [6]. O *Runtime* do ROS forma uma rede ponto-a-ponto de processos que são livremente acopladas utilizando a infraestrutura de comunicação nas modalidades síncrona ou assíncrona.

O *middleware* funciona como um intermediador da comunicação dos programas clientes escritos no ROS com o RoboDeck, capaz de se comunicar com a placa MAP do

RoboDeck e seu objetivo é oferecer uma API por meio do ROS para esta comunicação [8]. O ROSJava é um pacote disponível para o ROS e oferece serviços para a escrita de algoritmos (nós) na linguagem Java e criação de projetos para integração com outras plataformas, como o sistema operacional *Android* [9]. A Figura 2 mostra o diagrama de comunicação do *middleware*.

Figura 2. Diagrama de comunicação do *middleware*



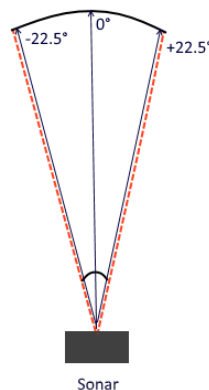
Sensores ultrassônicos, também conhecido como sonares, são vastamente utilizados em robótica móvel devido ao seu baixo custo, manuseio simples e, normalmente, leituras precisas [10]. Os sistemas de ultrassom são adequados somente em pequenas áreas de trabalho e se inexistirem obstáculos que possam interferir com a propagação do sinal [11]. Para determinar a distância do sonar para um objeto, Usa-se pulsos acústicos emitidos por um transdutor e suas reflexões. Esta técnica também é conhecida como *time-of-flight* (TOF) [12]. O cálculo da distância  $d$  retornada por um sensor ultrassônico pode ser dado por  $d = v \times t/2$ , onde  $v$  é a velocidade do som (aproximadamente  $343 \text{ m/s}$ ) e  $t$  é o tempo de viagem da partícula em segundos. Deve-se considerar o tempo de eco até o sensor, dividindo-se o tempo por dois. Apesar das suas vantagens, esses sensores sofrem de dois problemas conhecidos: medidas incorretas e incerteza angular [13]. Neste trabalho, o sensor do robô cobre um feixe cônico de aproximadamente  $45^\circ$ . A Figura 3 mostra a distribuição do feixe cônico do sonar. O intervalo de confiança de leitura do sonar foi definido para 5 e 90 cm.

Odometria é um método muito usado para posicionamento de um robô móvel, obtida a partir de sensores, como os populares *encoders* rotativos que transformam o movimento de rotação dos motores em pulsos ou sinais digitais. Segundo [14], é bem conhecido que odometria proporciona boa precisão, curto prazo, baixo custo e ainda permite taxas de amostragem muito elevadas. Porém, seu modelo gera erros inevitáveis ao longo do tempo, dependendo da distância percorrida pelo robô móvel acumulativamente.

### 3. Experimentais Computacionais

Nesta seção, experimentos são apresentados visando a obtenção de mapas de um ambiente fechado e controlado a partir de movimentos elementares e leituras dos sensores

Figura 3. Modelagem do sonar



ultrassônicos: em linha reta para frente e para trás, rotação à direita e à esquerda, e comando de parada do robô. Mapas de grades de ocupação consistem em uma matriz de duas dimensões de células, onde cada célula contém um valor de ocupação, definido para representar se a célula está livre, ocupada ou indefinida. Neste trabalho, os valores das células são definidos entre 0 e 1, onde 0 representa livre, 1 representa ocupado e 0,5 representa estado indefinido.

O ambiente de teste é formado pela versão beta do *middleware* de integração RoboDeck-ROS, Sistema Operacional Linux, distribuição Ubuntu 14.04 LTS, e ROS distribuição Indigo. O ambiente físico, utilizado para realizar o mapeamento, foi contido em placas de material MDF de dimensões  $270 \times 264 \times 45$  (largura, profundidade e altura dados em cm), revestidas de material adequado para menor interferir nos sonares.

O ambiente físico foi dividido em quadrantes, sendo que cada um deles associado a diferentes mapas, chamados de secundários. Os quatro mapas secundários formam o mapa global. As dimensões de cada mapa secundário é  $150 \times 150$  cm e as dimensões do mapa global são de  $336 \times 330$  cm. O procedimento do mapeamento do ambiente ocorreu em 9 etapas: mapeamento, movimentação linear de 100 cm, mapeamento, rotação em  $90^\circ$ , movimentação linear de 100 cm, mapeamento, rotação em  $90^\circ$ , movimentação linear de 100 cm, mapeamento.

Os *encoders* foram usados cálculo da distância percorrida pelo robô a cada movimento. Conhecido o diâmetro da roda e os 400 pulsos que representam uma revolução da roda, a distância percorrida pelo robô é dada por  $D = \frac{E2\pi r}{400}$ , onde  $E$  é a leitura efetuada por um dos *encoder* e  $r$  é o valor do raio das rodas.

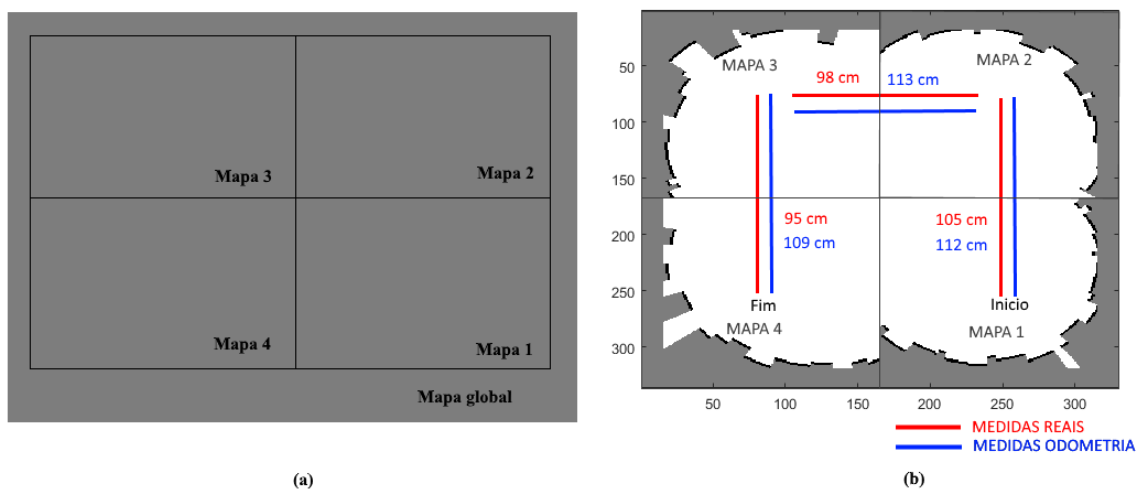
A Tabela 1 mostra as medidas reais e valores odométricos da distância percorrida pelo robô ao receber um comando de movimentação linear de 100 cm. Nota-se que locomoção é, em média, 7,2 cm a menos que o esperado, com variabilidade de 10,35 cm.

A proposta da união dos mapas secundários para a formação de um mapa global é apresentada na Figura 4. Apresentam-se as distâncias reais percorridas pelo robô e aquelas calculadas pela odometria. A modelagem estatística do erro é um fator de grande importância para a correção da distância percorrida pelo robô para o caso de navegação autônoma.

Tabela 1. Valores enviados ao robô e suas respectivas medidas reais.

Distância Enviada ao Robô (DER)	Distância Percorrida Real (DPR)	Distância Percorrida Odometria (DPO) %	Erro DER-DPO (%)	Erro DER-DPR (%)
100	91,5	100,6095	0,6058	9,2896
100	96	100,7194	0,7143	4,1667
100	86,5	101,4891	1,4673	15,6069
100	117	126,2291	20,7790	14,5299
100	80	100,4995	0,4970	25,0000
100	85	105,33	5,0603	17,6471
100	91	101,3791	1,3603	9,8901
100	100	102,2588	2,2089	0,0000
100	82	101,2692	1,2533	21,9512
100	99	100,2796	0,2788	1,0101
Média ± desvio padrão	92,8 ± 10,3542	104,0063 ± 7,9437	3,4225 ± 6,2530	11,9 ± 8,5452

Figura 4. (a) - Mapa global dividido em mapas secundários. (b) - Mapa global do ambiente a partir dos mapas secundários.



#### 4. Conclusão

Exploração autônoma não é uma tarefa trivial quando esta é destinada a robôs móveis, que necessitam integrar o sistema de localização e construção do mapa do ambiente com um processo cognitivo capaz de guiar o robô usando as informações incrementalmente colhidas durante a exploração. Este trabalho apresenta um estudo de caso sobre a plataforma robótica RoboDeck visando desenvolver rotinas computacionais específicas necessárias para sua utilização na tarefa de exploração autônoma, com suporte do Sistema Operacional Robótico (*Robot Operating System* – ROS) e sensores de baixo custo. Experimentos são apresentados para a construção de mapas de ambientes fechados e controlados. Os resultados são considerados satisfatórios dadas as características dos sensores ultrassônicos

que equipam a plataforma. Os mapas produzidos apresentaram instabilidades à proporção que as distâncias envolvidas entres o robô e os obstáculos se tornaram maiores. Faz-se necessário, como trabalho futuro, a modelagem do erro na odometria, assim como estudo de métodos para melhorar os mapas gerados com os sensores, visando equipar o sistema para navegação autônoma.

## Referências

- [1] V. Varveropoulos, “Robot localization and map construction using sonar data,” *The Rossum Project*, vol. 10, pp. 1–10, 2005.
- [2] L. Yenilmez and H. Temeltas, “Map building for mobile robots by ultrasonic and infrared sensor data fusion,” in *Optomechatronic Systems III*, pp. 444–449, International Society for Optics and Photonics, 2002.
- [3] J. G. Juang and J. A. Wang, “Indoor map building by laser sensor and positioning algorithms,” in *Applied Mechanics and Materials*, vol. 764, pp. 752–756, Trans Tech Publ, 2015.
- [4] J. Guivant, E. Nebot, and S. Baiker, “Autonomous navigation and map building using laser range sensors in outdoor applications,” *Journal of robotic systems*, vol. 17, no. 10, pp. 565–583, 2000.
- [5] XBOT, “Apostila de software do robodeck versão 1.1,” 2011.
- [6] “ROS introduction.” <http://wiki.ros.org/ROS/Introduction>, 2014. Acessado: 10/07/2016.
- [7] L. Zanolla, D. R. de Sousa, R. M. Furlaneto, W. T. Botelho, and M. d. G. B. Marietto, “Implementação com validação real de um controle proporcional, integral e derivativo na plataforma robótica robodeck,” *Simpósio de Tecnologia da Informação*, vol. V, 2014.
- [8] T. Y. Gashu, “Integração do sistema operacional ros com o módulo de alta performance robodeck,” 2014.
- [9] “ROS java.” [https://github.com/rosjava/rosjava\\_core](https://github.com/rosjava/rosjava_core), 2013. Acessado: 10/07/2016.
- [10] E. Araujo and R. A. Grupen, “Feature detection and identification using a sonar-array,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, pp. 1584–1589, IEEE, 1998.
- [11] H. A. SECCHI, “Uma introdução aos robôs móveis,” *Instituto de Automática–INAUT. Universidad Nacional de San Juan–UNSJ–Argentina*, 2012.
- [12] L. Kleeman and R. Kuc, “Sonar sensing,” in *Springer Handbook of Robotics*, pp. 491–519, Springer, 2008.
- [13] K. Lee and W. K. Chung, “Effective maximum likelihood grid map with conflict evaluation filter using sonar sensors,” *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 887–901, 2009.
- [14] J. A. Zaki and O. Arafa, “Mobile robot positioning using odometry and ultrasonic sensors,” *Journal of Cybernetics and Informatics*, vol. 13, 2012.