



UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
TRABALHO DE CONCLUSÃO EM ENGENHARIA DE  
CONTROLE E AUTOMAÇÃO



# **Desenvolvimento de plataforma robótica omnidirecional**

*Autor: Emílio Dolgener Cantú*

*Orientador: Eduardo Perondi*

Porto Alegre, 13 de novembro de 2017



---

## **Sumário**



## **Agradecimentos**

Agradeço a todo mundo que ajudou.



## **Resumo**

O presente trabalho apresenta a implementação de uma base robótica omnidirecional holonômica, utilizando 3 omniwheels. Foram implementados um sistema de controle e um método de odometria em malha aberta. Os resultados foram xxxx.





## **Abstract**

This work shows the implementation of a holonomic omnidirectional robotic platform, using 3 omniwheels. Were also implemented a control system and a method for open loop odometry. As a result, xxxx.



## **Lista de Figuras**



## **Lista de Tabelas**









# 1 Introdução

O interesse na área da robótica se dá pela multi-disciplinaridade do tema, que abrange um espectro de conhecimentos que vai desde mecânica estrutural até a aplicação de teorias de controle sofisticadas. A área ainda se estende por eletrônica, elétrica, computação e até mesmo psicologia. Assim, considera-se que esta área é bastante adequada a um trabalho de conclusão de um curso igualmente abrangente, que é a Engenharia de Controle e Automação.

Em aplicações industriais, a maioria dos robôs utilizados são manipuladores, que realizam tarefas repetitivas – como soldagem ou montagem de peças – com precisão e rapidez adequados a cada aplicação. Estes robôs, no entanto, são em geral fixos, e daí surge o estudo da robótica móvel: como um robô pode se mover sem supervisão e interagir com o mundo real? (?) Além do interesse acadêmico, existe um significativo interesse comercial, visto que o mercado de robôs móveis, que estava em torno de 4,5 bilhões de dólares americanos em 2014, tende a duplicar até 2020 (?).

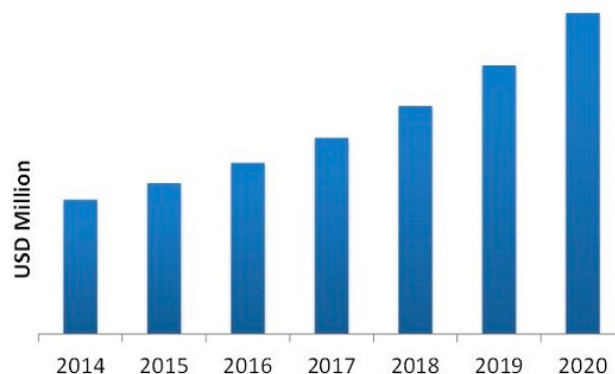


Figura 1: Projeção de crescimento do mercado de robótica móvel até 2020.

Fonte: ?

Segundo ?, robôs móveis são divididos em robôs não-holonômicos e robôs omnidirecionais, com diferenças significativas em planejamento de trajetória, controle e modelagem dos dois tipos de robô. Diante da natureza do presente trabalho de conclusão de curso, o escopo foi definido no âmbito dos robôs holonômicos omnidirecionais, que se destacam comercialmente pela habilidade de realizar transporte de cargas pequenas em espaços confinados – como corredores de hospital e depósitos de armazenamento que buscam o aumento da capacidade sem perder agilidade logística nem aumentar o espaço necessário nas instalações. Academicamente, o controle de rodas omnidirecionais apresenta diversos problemas interessantes, vários dos quais serão descritos ao longo do presente trabalho. O desenvolvimento de uma plataforma robótica omnidirecional holonômica se torna útil para futuras aplicações em diversas áreas de investigação em robótica, controle e automação.

O presente trabalho consiste no desenvolvimento (tanto teórico quanto experimental) de uma

plataforma robótica que possa se movimentar de maneira autônoma em qualquer direção do plano sem necessidade de reorientação – apresentando holonomicidade. Após uma avaliação na bibliografia sobre os tipos de robôs factíveis de serem construídos no tempo previsto com os recursos financeiros disponíveis, optou-se pela seguinte configuração. A plataforma considerada mais adequada utiliza 3 *omniwheels*, cada uma acionada por um motor elétrico dedicado, conforme o modelo mostrado na Figura ???. Como as rodas são montadas de maneira fixa no chassi, este tipo de robô ainda oferece a vantagem de ser construído com uma estrutura mecânica mais simples, como mencionado por ?. As rodas omnidirecionais utilizadas podem ser vistas em detalhe na Figura ???. O sistema de movimentação é controlado por software processado em um computador embarcado a partir dos sinais fornecidos por sensores inerciais e de odometria.

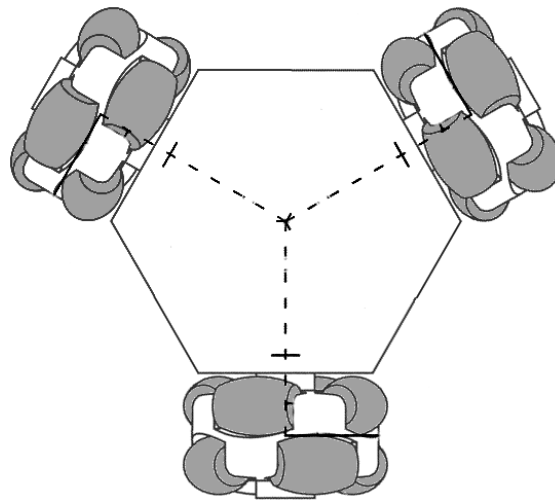


Figura 2: Diagrama de um robô móvel com três rodas omnidirecionais.

Fonte: adaptado de ?

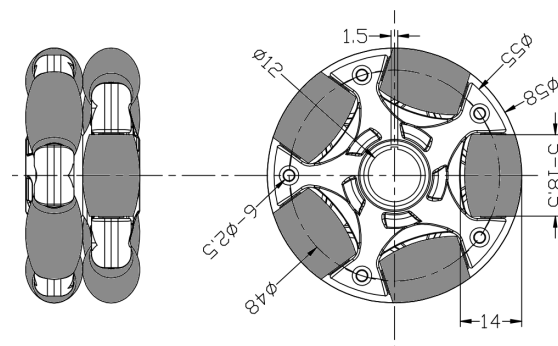


Figura 3: Desenho de uma roda omnidirecional.

Fonte: adaptado de ?

Este trabalho tem como **objetivo geral** projetar, construir, colocar em operação e testar uma plataforma robótica omnidirecional de baixo custo, mas com características semelhantes às dos sistemas comerciais. Para atingir esse objetivo, se devem alcançar os seguintes **objetivos específicos**:

- Modelagem do robô;
- Especificação e construção de um protótipo;
- Implantação de um algoritmo de controle;
- Implantação de instrumentação e algoritmo de localização;
- Realizar experimentos de seguimento de trajetórias e analisar os resultados obtidos.

O trabalho está organizado da seguinte maneira:

- Na Seção 2, é apresentada a revisão bibliográfica, versando sobre robótica móvel, técnicas de controle utilizadas na área e um breve resumo sobre métodos de localização, além de uma exposição dos trabalhos mais recentes envolvendo robôs omnidirecionais;
- Na Seção 3, especificação do hardware, estrutura mecânica e montagem do protótipo;
- Na Seção 4, é apresentado o desenvolvimento teórico da modelagem do robô e dos algoritmos de controle e localização;
- Na Seção 5 são implementados os algoritmos descritos na seção anterior;
- Na Seção 6, é desenvolvido e realizado um experimento para avaliar o desempenho do robô projetado;
- Por fim, na Seção 7, serão apresentados as análises e discussões sobre os resultados dos experimentos, a conclusão sobre o trabalho como um todo e algumas propostas para futuros trabalhos.

Além do descrito, o trabalho ainda contém os seguintes apêndices:

- Apêndice A, da descrição dos custos do projeto;
- o q mais vier.



## 2 Revisão Bibliográfica

### 2.1 Fundamentação Teórica

Para que se possa compreender o comportamento de um robô e desenvolver aplicações, é necessário obter o modelo de tal robô. Para robôs móveis, em contraste com braços robóticos, por exemplo, há um nível de complexidade adicional na estimação da posição do robô, visto que a plataforma móvel não possui nenhuma extremidade fixa em um ponto conhecido. No caso do robô omnidirecional com 3 rodas, se deseja obter um **modelo cinemático** que relacione as velocidades de cada roda à posição do robô no ambiente. Para robôs móveis em geral, primeiro se deve analisar as restrições de movimento adicionadas ao sistema por cada roda. Ao se utilizar rodas omnidirecionais, tais restrições não existem, simplificando a obtenção da cinemática direta do TOMR. Tal análise é demonstrada por diversos autores, como ?, por exemplo, e será utilizada nas próximas seções. Ainda de acordo com os autores, conforme a velocidade de operação aumenta, se torna importante realizar a análise dos efeitos dinâmicos do sistema – massa e força – durante a implantação. A **modelagem dinâmica** do robô em questão ainda é discutida, variando conforme a aplicação desejada, sendo uma das opções apresentada por ?.

Também são apresentados por ? diversos métodos de **planejamento de trajetória** para robôs, ou seja, a maneira como o robô (ou um efetuador) se movimenta de um determinado ponto até outro. Há diversos métodos de se implantar trajetórias, com diversos níveis de complexidade computacional. Implementações mais simples podem ser uma linha reta de um ponto a outro, com atuação repentina, causando picos de aceleração, ou mais refinadas, realizado interpolações polinomiais de quinta ordem para garantir velocidades e acelerações nulas nos pontos de origem e destino. Também é importante analisar o perfil de velocidade executado pelo robô, para garantir a operação mais eficiente possível para as características dinâmicas de cada aplicação.

Segundo ?, aplicar controle com realimentação em robôs omnidirecionais é relativamente simples, visto que esse tipo de plataforma robótica apresenta controlabilidade – ou seja, sempre há um conjunto de velocidades  $\dot{\phi}$  para as rodas que ocasiona numa certa velocidade  $v$  (translacional e rotacional) para o robô. ? sugerem a utilização de um controlador com realimentação de estados. Neste paradigma, o algoritmo de geração de trajetória divide o caminho a ser percorrido em diversos pontos, e o controlador implementado garante que o robô percorra tal trajeto, como se os dois sistemas – planejamento de trajetória e controle –, estivessem operando em camadas hierárquicas.

A maioria dos autores se concentra em controladores que levam em consideração apenas a modelagem cinemática do sistema. No entanto, nos casos em que o modelo do robô exija considerações dinâmicas e não-lineares, controladores mais complexos são utilizados. Dentre outros, ? descrevem o método de controle por torque computado, bastante popular, no qual o modelo dinâmico inverso é utilizado para linearizar a malha de controle. Ainda, conforme ?, o controle por torque computador configura um sistema de controle centralizado, enquanto a utilização de um PID para cada roda seria um exemplo de controle descentralizado.

A localização do robô no ambiente é essencial para o bom funcionamento de um robô móvel. Conforme [1], se realiza **odometria** – a medição da distância percorrida – através da integração das velocidades das rodas. Como o sensoriamento das rodas é muito comum, através de *encoders* de quadratura, por exemplo, tornando a odometria algo barato e conveniente. No entanto, devido às sucessivas integrações realizadas, erros de estimação tendem a se acumular ao longo do tempo de operação, devido a deslizamentos das rodas e erros numéricos, principalmente. Sensores relativos como acelerômetros e giroscópios também tendem a acumular o mesmo tipo de erro. Assim, conforme [1], se recomenda utilizar métodos de localização absolutos de tempo em tempo, como magnetômetros, GPS e marcadores fixos no ambiente, ou, conforme complementado por [2], se pode realizar uma integração das leituras de diversos sensores. Esse método é chamado de **fusão de dados**.

[1]: Capítulo 12, Arquitetura de sistemas robóticos.

[2]: pg 350, chapter 12 -> linguagens de programação para robótica

O **Raspberry Pi** é um *single board computer*, que utiliza a arquitetura ARM em seu processador, ideal para dispositivos alimentados por baterias por consumir pouca energia e gerar pouco calor. O processador possui quatro núcleos, e um *clock* de 1,2 GHz – poder computacional equivalente há um computador de mesa comum. O RPi utiliza um sistema operacional GNU/Linux, e *software* deve ser desenvolvido para ser executado nesta plataforma. Há ainda 40 pinos de GPIO que podem ser utilizados para conectar sensores, atuadores e diversos componentes, e suporte nativo a I2C ([3]).

Para a comunicação dos periféricos com este computador, é necessário utilizar algum protocolo de comunicação. O protocolo **Inter-Integrated Circuit, ou Circuito Inter-Integrado**, é geralmente utilizado em robôs, com um grande suporte tanto pela RPi ([3]) quanto pelos componentes em geral utilizados ([4] e a bússola e o arduino se eu usar). Com este protocolo, descrito em [5], dados podem ser transmitidos a 100 Kbps – ou 400 Kbps quando utilizado o *fast mode*. São utilizados duas linhas bidirecionais no barramento: SDA para os dados e SCL para os sinais de *clock*. O número de dispositivos conectados ao barramento só depende do limite de capacitância descrito na especificação. Resistores de *pull-up* são necessários para manter a linha em estado lógico alto quando não utilizada, porém estes resistores estão presentes internamente no *Raspberry Pi*, por exemplo.

ARDUINO? TEMPO REAL?

## 2.2 Estado da Arte

A grande maioria dos robôs construídos com *omniwheels* utiliza 3 rodas em uma configuração triangular simétrica – como apresentado na Figura ?? –, a exemplo de [6], [7], [8] e [9], entre outros. Alguns autores, como [10] e [11] utilizam 4 rodas, sendo que este último desenvolveu algoritmos para que o robô continuasse operando mesmo que um dos motores deixe de funcionar. Em diversos trabalhos existe uma preocupação em relação a possíveis derrapagens das rodas ao aumentar a velocidade

de operação. ? apresentam um estudo sobre os coeficientes de atrito de rodas omnidirecionais em diversas superfícies e um modelo dinâmico que leva tal efeito em consideração. ? utilizam 3 rodas motrizes e outras 3 rodas livres ligadas a *encoders*, para que deslizamentos devido a torque excessivo não afetem a odometria.

? construíram um robô omnidirecional e um sistema de transmissão mecânica que permite holonomicidade utilizando rodas convencionais. Sugerem ainda modos de operação para o robô caso algum dos motores parem de funcionar, passando a se comportar de maneira similar a um robô diferencial. Nestes casos de falha, no entanto, se perde a capacidade omnidirecional.

Há muitas técnicas de **controle** desenvolvidas para estes sistemas. Assim, é possível que cada autor utilize a que mais convenha às suas necessidades específicas. ? utiliza um controlador PID para cada roda, com parâmetros escolhidos empiricamente. Por sua vez, o robô torna-se difícil de controlar com velocidades acima de 1 m/s, segundo resultados de simulações. ? utilizam 3 PID, para controlar posição e orientação do robô, e relatam em detalhes o desenvolvimento de tais controladores. Em contraste, ? e ? também utilizam PID, porém para o controle de individual de cada motor, utilizando apenas o modelo cinemático do sistema. ? também sugere estratégias para evitar saturação dos atuadores.

Tanto ? quanto ? utilizam redes neurais para ajustar parâmetros dos controladores, sendo que no primeiro se tem uma estrutura de controle baseada em redes neurais enquanto que no segundo se utilizam as redes para calcular os parâmetros de 5 controladores PID, melhorando o desempenho mesmo levando em consideração de não-linearidades nos modelos dinâmicos utilizados. ? ainda mencionam que os resultados obtidos não foram os melhores possíveis, devido à dificuldade de se coletar dados de treinamento para as redes.

A maioria das implementações de robôs móveis hoje em dia combinam diversas técnicas de **localização e odometria** para implantar a realimentação necessária aos sistemas de controle. ? propõem um sistema de localização para robô omnidirecional baseado em odometria (localização relativa) e triangulação ativa de sensores no ambiente (localização absoluta), com fusão de dados para obter o resultado final. ? utilizam a leitura dos *encoders* das rodas e uma câmera externa, enquanto que ? utilizam apenas um giroscópio e um sensor de distância. ?, por outro lado, utilizam medições de distância utilizando sensores laser em AGV.

? mostram que é possível executar um algoritmo de determinação de atitude a partir de uma IMU utilizando TRIAD, filtros de Kalman, covariância de Allen e a plataforma Arduino UNO, com razoável precisão. ? também analisam fusão de dados utilizando um filtro de Kalman indireto para realizar *dead-reckoning* a partir da leitura de *encoders* e um giroscópio. Métodos de localização também são desenvolvidos em outras áreas, como relata ?, que implementam três métodos de localização baseados em INS para trajetórias de pedestres, concluindo que os resultados podem ser melhorados quando há mais qualidade na detecção da orientação. ? obtiveram resultados similares na mesma área.

Dos trabalhos mencionados, poucos entram em detalhes quanto ao *hardware* utilizado. ? utilizam

um computador embarcado de 2,6 GHz, uma grande evolução em relação a [1], que utilizavam um computador Motorola 68000 com aproximadamente um milésimo da capacidade computacional daquele. [2] e [3] utilizam computadores externos, envolvendo atrasos na comunicação entre tais computadores e o robô. [4] apresentam um sistema que implementa um filtro de Kalman em um microprocessador Arduino UNO, uma alternativa de baixo custo. Diversos trabalhos mais recentes, como [5], utilizam especificamente o computador embarcado **Raspberry Pi** para o processamento e um **Arduino** para a interface com sensores e atuadores – solução esta também adotada neste trabalho.

Para a avaliação dos resultados da plataforma construída, [6] implementaram 4 tipos de trajetória: translação retilínea, translação curvilínea – ambas sem alteração na orientação –, rotação pura e um caminho combinado de rotação em torno do seu centro e translação retilínea em relação às referências globais. Duas canetas foram montadas no robô, uma no centro e outra na periferia, para avaliar o resultado das movimentações executadas.



### 3 Especificação e montagem do protótipo

#### SEÇÃO EM CONSTRUÇÃO

Para o projeto, foi realizada a compra dos componentes no Apêndice A (COMO CITAR APÊNDICES??)

Apesar da falta de redundância, robôs omnidirecionais com 3 rodas são utilizados com mais frequência por serem mais simples de se implementar, apresentarem custo mais baixo, e

Foram escolhidas rodas suecas de 90 graus para permitir o uso de apenas 3 rodas, gerando economia de atuadores, peso reduzido, menor custo total, etc.

motores: <https://www.banggood.com/6V-210RPM-Encoder-Motor-DC-Gear-Motor-with-Mounting-Bracket-and-Wheel-p-1044064.html?p=970719369296201312SG>

?, pg 97: sensores.



## 4 Desenvolvimento Teórico

Coordenadas de referência na Equação ?? (?), também utilizado por ?:

$$\begin{pmatrix} x \\ y \\ \theta \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_w \\ y_w \\ \theta \end{pmatrix} \quad (1)$$

Cinemática direta, para um robô com 3 rodas dispostas em simetria radial em torno do centro da estrutura, é dada pela Equação ??. Diversos autores utilizam variações da mesma modelagem (?, ?, ?, entre outros).

$$\begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} = \frac{r}{3R} \begin{pmatrix} -\frac{3R}{\sqrt{3}} & 0 & \frac{3R}{\sqrt{3}} \\ R & -2R & R \\ 1 & 1 & R \end{pmatrix} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix}. \quad (2)$$

Cinemática inversa, obtida realizando a inversão da matriz 3x3, é dada pela Equação ??. Nota-se que esta inversão é simplificada no caso do robô com 3 rodas, visto que quando há mais rodas é formada uma matriz 3xn, e se deve realizar a pseudo-inversa, conforme demonstrado por ?. Nas equações apresentadas,  $r$  é o raio de cada roda e  $R$  o raio do robô (a distância do centro da roda ao centro da estrutura do robô).

$$\begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix} = \frac{1}{r} \begin{pmatrix} -\frac{\sqrt{3}}{2} & \frac{1}{2} & R \\ 0 & -1 & R \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & R \end{pmatrix} \begin{pmatrix} v_x \\ v_y \\ \omega \end{pmatrix} \quad (3)$$

Como pela classificação de ? um TOMR é caracterizado na categoria (3,0), o modelo cinemático das equações ?? e ?? é controlável, estável e descreve a posição, orientação e suas derivadas de forma suficiente.

?: Descrevem fórmulas para odometria, e dizem que se isso não for muito bom, nem adianta ter um controlador massa. Utilizam três rodas passivas com os encoders, para evitar problemas. Solução a meio enjambrada.

Para se realizar a odometria do robô com uma precisão melhor, ? ainda fornece as equações relacionadas ao **body twist**  $v_b = (v_{bx}, v_{by}, \omega_b)^T$ , que relaciona a velocidade do centro geométrico do robô com as velocidades de cada uma das rodas, que diferem caso o robô realize rotações durante a translação. Tal relação é dada pela Equação ??.

$$v_b = No - entendi - nada. \quad (4)$$

Modelagens dinâmicas também podem ser utilizadas, relacionando o comportamento do robô não às velocidades das suas rodas, mas sim ao torque aplicado a cada uma pelos motores. No entanto, no

caso do TOMR, se podem utilizar as velocidades das rodas como entradas, desde que haja um *loop* de controle que garanta tais velocidades durante o acionamento (CITAR AS NOTAS DE AULA DO WALTER??? control.pdf).

?, capítulo 9.

?: Sugerem que utilizar um controlador para cada roda é melhor do que para cada grau de liberdade. No nosso caso, é tranquilo pois temos apenas 3 rodas, mantendo o mesmo número de controladores. Devido à realimentação externa lenta, utilizam um preditor no robô. Não entram em detalhes. motores: <https://www.banggood.com/6V-210RPM-Encoder-Motor-DC-Gear-Motor-with-Mounting-Bracket-and-Wheel-p-1044064.html?p=970719369296201312SG>

Given a desired trajectory  $q_d(t)$ , we can adopt the feedforward plus proportional feedback linear controller (11.1) of Chapter 11 to track the trajectory: ?

?: Definir os coeficientes dos PIDs é uma novela, pois devemos levar em consideração parâmetros que possuem muita variação, como o coeficiente de atrito do solo, características das baterias, entre outros. Controle deles é bem legal.

?: 5.6.3.1 Introduction to Kalman filter theory

## **5 Implementação dos algoritmos**

SEÇÃO EM CONSTRUÇÃO



## **6 Avaliação experimental**

SEÇÃO EM CONSTRUÇÃO





## **7 Resultados**

DEVE TER EM TORNO DE DEZ PAGINAS



## **8 Conclusão e Trabalhos Futuros**

EM CONSTRUÇÃO



## 9 Referências

- Campion, G., Bastin, G., Dandrea-Novet, B., 1996. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE transactions on robotics and automation* 12 (1), 47–62.
- Carrasco, M. M., da Silva, A. L., 2016. Attitude determination for low cost imu and processor board using the methods of triad, kalman filter and allan variance. *Revista Brasileira de Iniciação Científica* 3 (2), 26–41.
- Craig, J., 2017. *Introduction to Robotics: Mechanics and Control*. Pearson.  
URL <https://books.google.com.br/books?id=JblZuwAACAAJ>
- Feng, D., Friedman, M. B., Krogh, B. H., 1989. The servo-control system for an omnidirectional mobile robot. In: *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*. IEEE, pp. 1566–1571.
- Garcia-Saura, C., 2015. Self-calibration of a differential wheeled robot using only a gyroscope and a distance sensor. *arXiv preprint arXiv:1509.02154*.
- Ginzburg, S., Nokleby, S., 2013. Indoor localization of an omni-directional wheeled mobile robot. *Transactions of the Canadian Society for Mechanical Engineering*. Canada 37, 1043–1056.
- Indiveri, G., 2009. Swedish wheeled omnidirectional mobile robots: Kinematics analysis and control (1), 164–171.
- Invensense, 2013. *Mpu6050. Revisão 3.4*.  
URL [https://store.invensense.com/datasheets/invensense/MPU-6050\\_DataSheet\\_V3%204.pdf](https://store.invensense.com/datasheets/invensense/MPU-6050_DataSheet_V3%204.pdf)
- Jimenez, A. R., Seco, F., Prieto, C., Guevara, J., 2009. A comparison of pedestrian dead-reckoning algorithms using a low-cost mems imu. In: *Intelligent Signal Processing, 2009. WISP 2009. IEEE International Symposium on*. IEEE, pp. 37–42.
- Jung, M.-J., Kim, J.-H., 2001. Fault tolerant control strategy for omnikity-iii. In: *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. Vol. 4. IEEE, pp. 3370–3375.
- Kim, H., Kim, B. K., 2014. Online minimum-energy trajectory planning and control on a straight-line path for three-wheeled omnidirectional mobile robots. *IEEE Transactions on industrial electronics* 61 (9), 4771–4779.
- Krinkin, K., Stotskaya, E., Stotskiy, Y., 2015. Design and implementation raspberry pi-based omni-wheel mobile robot. In: *Artificial Intelligence and Natural Language and Information Extraction*,

- Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT), 2015. IEEE, pp. 39–45.
- Loh, W., Low, K. H., Leow, Y., 2003. Mechatronics design and kinematic modelling of a singularityless omni-directional wheeled mobile robot. In: Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on. Vol. 3. IEEE, pp. 3237–3242.
- Lynch, K., Park, F., 2017. Modern Robotics: Mechanics, Planning, and Control. Cambridge University Press.  
URL <https://books.google.com.br/books?id=86G0nQAACAAJ>
- Markets, Markets, 2015. Mobile robots market by environment (aerial, ground, and marine), component (hardware and software), application (professional service and personal service), and geography (north america, europe, asia-pacific, and the row) - global forecast to 2020. <http://www.marketsandmarkets.com/Market-Reports/mobile-robots-market-43703276.html>, acessado em 27/09/2017.
- Nexus, 2016. 58mm plastic omniwheel. Acessado em 8/10/2017.  
URL <http://www.nexusrobot.com/product/58mm-plastic-omni-wheel-for-lego-nxt-and-servo-motor-14135.html>
- Oubbati, M., Schanz, M., Buchheim, T., Levi, P., 2005. Velocity control of an omnidirectional robocup player with recurrent neural networks. In: RoboCup. Springer, pp. 691–701.
- Park, K., Chung, D., Chung, H., Lee, J. G., 1996. Dead reckoning navigation of a mobile robot using an indirect kalman filter. In: Multisensor Fusion and Integration for Intelligent Systems, 1996. IEEE/SICE/RSJ International Conference on. IEEE, pp. 132–138.
- Pin, F. G., Killough, S. M., 1994. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. IEEE transactions on robotics and automation 10 (4), 480–489.
- Ritter, G. A., 2016. Modelo genérico de plataforma robótica omnidirecional em código aberto. Ph.D. thesis, UNISC, Santa Cruz do Sul, RS, trabalho de Conclusão de Curso de Engenharia de Computação.
- Röhrig, C., Heß, D., Kirsch, C., Künemund, F., 2010. Localization of an omnidirectional transport robot using ieee 802.15. 4a ranging and laser range finder. In: Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on. IEEE, pp. 3798–3803.
- Rojas, R., Förster, A. G., 2006. Holonomic control of a robot with an omnidirectional drive. KI-Künstliche Intelligenz 20 (2), 12–17.
- Samani, H. A., Abdollahi, A., Ostadi, H., DaneshPanah, M., 2007. Comprehensive omni-directional soccer player robots. International Journal of Advanced Robotic Systems.
- Semiconductors, P., 2000. The i2c-bus specification. Philips Semiconductors 9397 (750), 00954.

- Siciliano, B., Khatib, O., 2016. Springer handbook of robotics. Springer.
- Siegwart, R., Nourbakhsh, I., Scaramuzza, D., 2011. Introduction to Autonomous Mobile Robots. Intelligent robotics and autonomous agents. MIT Press.  
URL <https://books.google.com.br/books?id=4of6AQAAQBAJ>
- Steinhoff, U., Schiele, B., 2010. Dead reckoning from the pocket-an experimental study. In: Pervasive Computing and Communications (PerCom), 2010 IEEE International Conference on. IEEE, pp. 162–170.
- Takemura, Y., Sanada, A., Ichinose, T., Nakano, Y., Nassiraei, A. A., Azeura, K., Kitazumi, Y., Ogawa, Y., Godler, I., Ishii, K., et al., 2007. Development of “hibikino-musashi” omni-directional mobile robot. In: International Congress Series. Vol. 1301. Elsevier, pp. 201–205.
- Treesatayapun, C., 2011. A discrete-time stable controller for an omni-directional mobile robot based on an approximated model. Control Engineering Practice 19 (2), 194–203.
- Upton, E., Halfacree, G., 2014. Raspberry Pi user guide. John Wiley & Sons.
- Williams, R. L., Carter, B. E., Gallina, P., Rosati, G., 2002. Dynamic model with slip for wheeled omnidirectional robots. IEEE transactions on Robotics and Automation 18 (3), 285–293.





---

## **Apêndices**

## A Custo dos componentes utilizados

Há no mercado uma variada gama de componentes a serem utilizados em projetos robóticos. Para o projeto em questão, se utilizaram componentes que mostrassem um preço de mercado competitivo e grande disponibilidade. Dessa forma, se pode manter o projeto viável, mesmo que por vezes sacrificando um possível incremento de desempenho que se daria ao utilizar um componente mais robusto, por exemplo.

Na Tabela ?? se pode ver a lista de componentes adquirida e os respectivos custos. Nota-se que no caso das 3 rodas há – integrado ao valor apresentado – as taxas de importação e conversão de moedas, visto que esses componentes foram importados dos Estados Unidos. Também é importante mencionar que o chassi foi produzido sem custo BLABLABLAB LAMECC.

Tabela 1: Custo dos componentes utilizados no projeto.

Item:	Valor por unidade:	Quantidade:	Valor total:
Omniwheel	R\$ 46,76	3	R\$ 140,29
Motor c/ encoder	R\$ 119,00	3	R\$ 357,00
Driver	R\$ 15,00	2	R\$ 30,00
Raspberry Pi	R\$ 148,00	1	R\$ 148,00
microSD 16GB	R\$ 34,00	1	R\$ 34,00
Arduino Mega	R\$ 40,00	1	R\$ 40,00
IMU MPU6050	R\$ 9,00	1	R\$ 9,00
Magnetômetro HMC5883	R\$ 12,80	1	R\$ 12,80
Chassi	R\$ 5,00	1	R\$ 5,00
Bateria	R\$ 100,00	1	R\$ 100,00
Reguladores de Tensão	R\$ 4,98	4	R\$ 19,96
Diversos	R\$ 30,00	X	R\$ 30,00
<b>Custo Total:</b>			<b>R\$ 926,05</b>

Numa escala de preços em robótica, se percebe que (COMPARAR OS PREÇOS DE OUTROS PROTÓTIPOS??)