



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, 28 de novembro de 2017

Sumário

Agradecimentos	iii
Resumo	v
Abstract	vii
Lista de Figuras	ix
Lista de Tabelas	xi
Lista de Abreviaturas e Siglas	xii
Lista de Símbolos	xiii
1 Introdução	1
2 Revisão Bibliográfica	5
2.1 Fundamentação Teórica	5
2.2 Estado da Arte	6
3 Especificação e montagem do protótipo	9
4 Desenvolvimento Teórico	11
5 Implementação dos algoritmos	15
6 Avaliação experimental	17
7 Resultados	19
8 Conclusão e Trabalhos Futuros	21
9 Referências	23
Apêndices	27
A Custo dos componentes utilizados	28

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

1	Projeção de crescimento do mercado de robótica móvel até 2020.	1
2	Diagrama de um robô móvel com três rodas omnidirecionais.	2
3	Desenho de uma roda omnidirecional.	3
4	Chassi projetado. TROCAR A IMAGEM.	10
5	Sistemas de coordenadas global I e relativo ao centro do robô R.	11
6	Vista superior do robô, mostrando as convenções adotadas. v_x e v_y estão no sistema de coordenadas do robô.	12

Lista de Tabelas

1	Custo dos componentes utilizados no projeto.	28
---	--	----

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? (Siegwart et al. (2011)) 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 (Markets and Markets (2015)).

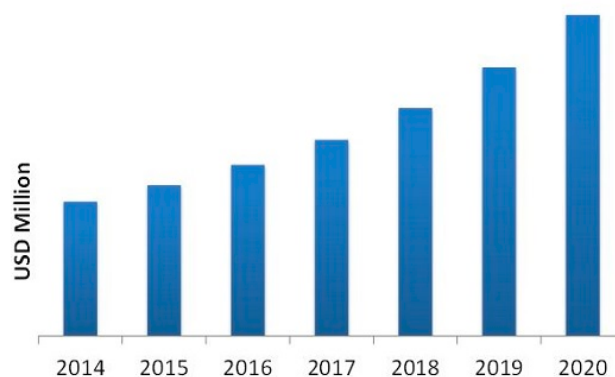


Figura 1: Projeção de crescimento do mercado de robótica móvel até 2020.

Fonte: Markets and Markets (2015)

Segundo Lynch and Park (2017), 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 2. 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 Siciliano and Khatib (2016). As rodas omnidirecionais utilizadas podem ser vistas em detalhe na Figura 3. O sistema de movimentação é controlado por software processado em um computador embarcado a partir dos sinais fornecidos por sensores inerciais e de odometria.

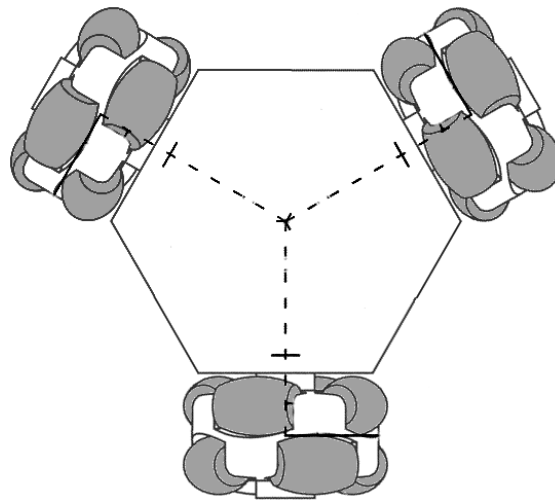


Figura 2: Diagrama de um robô móvel com três rodas omnidirecionais.

Fonte: adaptado de Ritter (2016)

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



Figura 3: Desenho de uma roda omnidirecional.

Fonte: adaptado de Nexus (2016)

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.
- **VERIFICAR. TEM UMA SESSÃO A MAIS AQUI.. DE ONDE SAIU CONCLUSÃO VS RESULTADOS**

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 compreender o comportamento de um robô e desenvolver aplicações, é necessário obter o modelo matemático do 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. Nesse caso, não é necessário analisar as restrições de movimento adicionadas ao sistema por cada roda, simplificando a obtenção da cinemática direta do TOMR. Tal análise é apresentada por diversos autores, como Siegwart et al. (2011), por exemplo, e será abordada nas próximas seções. Ainda de acordo com estes autores, conforme a velocidade de operação aumenta, se torna importante realizar a análise dos efeitos dinâmicos do sistema. A **modelagem dinâmica** do robô em questão ainda é foco de pesquisa, variando conforme a aplicação desejada, sendo uma das opções apresentada por Kim and Kim (2014). FALAR UM POUCO DESSA MODELAGEM

Em Lynch and Park (2017) são apresentados 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 diversas técnicas de se implantação de trajetórias, com diversos níveis de complexidade computacional. Implementações mais simples podem ser uma linha reta de um ponto a outro, com atuação relativamente brusca, causando picos de aceleração, ou mais refinadas, realizado interpolações polinomiais 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 uma operação eficiente para as características dinâmicas de cada aplicação.

Segundo Lynch and Park (2017), 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 uma certa velocidade v (translacional e rotacional) para o robô. Siegwart et al. (2011) sugerem a utilização de um controlador com realimentação de estados. Nesta estratégia, o algoritmo de geração de trajetória divide o caminho a ser percorrido em diversos pontos, enquanto que 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 trabalhos 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, Siciliano and Khatib (2016) 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 Indiveri (2009), o controle por torque computador configura um sistema de controle centralizado, enquanto a utilização de um controlador PID para cada roda seria um exemplo de aplicação de uma

estratégia de controle descentralizado.

A localização do robô no ambiente é essencial para o bom funcionamento de um robô móvel. Conforme Lynch and Park (2017). Uma estratégia para se realizar a **odometria** – a medição da distância percorrida – é através da integração das velocidades das rodas. Como o sensoramento das rodas é muito comum, através de *encoders* de quadratura, por exemplo, tornando-se a odometria barata 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 como acelerômetros e giroscópios também tendem a acumular o mesmo tipo de erro. Assim, conforme Siegwart et al. (2011), se recomenda utilizar métodos de localização absolutos de tempo em tempo, como magnetômetros, GPS e marcadores fixos no ambiente, ou, conforme complementado por Lynch and Park (2017), se pode realizar uma conjugação das leituras de diversos sensores, por meio do método conhecido como **fusão de dados**.

Poucos trabalhos detalham a implantação dos sistemas desenvolvidos. De acordo com Craig (2017), tal fato se deve às constantes mudanças e atualizações tecnológicas em *hardware* e linguagens de programação. Os autores tratam, então, de apresentar os fundamentos básicos da implementação, cabendo a cada pesquisador definir suas soluções a partir do que está disponível. Para Siciliano and Khatib (2016), a escolha da arquitetura utilizada é muito subjetiva, sendo comum tratar o robô como um conjunto de subsistemas. Tal abordagem busca melhorar a modularidade do sistema, facilitar a reutilização de módulos em outros projetos e facilitar testes, validações e manutenção. Arquiteturas mais voltadas à robótica tendem a incluir capacidades de processamento em tempo real, controle de atuadores e sensores, e, por vezes, operam em diferentes camadas de prioridade, com escalas de tempo diferentes para diferentes tarefas.

Craig (2017) ainda apresenta diversos paradigmas para as linguagens de programação utilizadas em robótica, sendo que uma das grandes tendências hoje é a utilização de linguagens gerais já existentes, com a adição de bibliotecas voltadas para o desenvolvimento de aplicações robóticas (ou o desenvolvimento das próprias bibliotecas). A escolha da linguagem a ser utilizada, conforme Siciliano and Khatib (2016), deve ser realizada com cuidado, com o objetivo de manter o desenvolvimento do sistema mais fácil, seguro e flexível. Assim, essa escolha pode ser baseada nas experiências anteriores do desenvolvedor, no tipo de robô e nas tarefas que se deseja executar, sem haver um consenso da comunidade acadêmica sobre qual a melhor abordagem.

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 2 –, a exemplo de Ritter (2016), Samani et al. (2007), Williams et al. (2002) e Indiveri (2009), entre outros. Alguns autores, como Krinkin et al. (2015) e Rojas and Förster (2006) 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. Williams et al. (2002) apresentam um estudo sobre os coeficientes de atrito de rodas omnidirecionais em diversas superfícies e um modelo dinâmico que leva tal efeito em consideração. Samani et al. (2007) utilizam 3 rodas motrizes e outras 3 rodas livres ligadas a *encoders*, para que deslizamentos devido a torque excessivo não afetem a odometria.

Jung and Kim (2001) 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. Ritter (2016) 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. Samani et al. (2007) utilizam 3 controladores PID, para controlar posição e orientação do robô, e relatam em detalhes o desenvolvimento de tais controladores. Em contraste, Rojas and Förster (2006) e Indiveri (2009) também utilizam PID, porém para o controle de individual de cada motor, utilizando apenas o modelo cinemático do sistema. Indiveri (2009) também sugere estratégias para evitar saturação dos atuadores.

Tanto Treesatayapun (2011) quanto Oubbati et al. (2005) 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. Oubbati et al. (2005) 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. Ginzburg and Nokleby (2013) 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. Rojas and Förster (2006) utilizam a leitura dos *encoders* das rodas e uma câmera externa, enquanto que Garcia-Saura (2015) utilizam apenas um giroscópio e um sensor de distância. Röhrig et al. (2010), por outro lado, utilizam medições de distância utilizando sensores laser (aplicados em um AGV).

Carrasco and da Silva (2016) mostram que é possível executar um algoritmo de determinação de atitude a partir de uma IMU (ADICIONAR NO GLOSSÁRIO) utilizando TRIAD (EXPLICAR ISSO AQUI), filtros de Kalman, covariância de Allen (ISSO AQUI TBM) e a plataforma Arduino UNO, com razoável precisão. Park et al. (1996) 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 Jimenez et al. (2009), que implementam três métodos de localização baseados em INS (COLOCAR NO GLOSSÁRIO) para trajetórias de pedestres, concluindo que os resultados podem ser melhorados

quando há mais qualidade na detecção da orientação. Steinhoff and Schiele (2010) obtiveram resultados similares na mesma área.

Dos trabalhos mencionados, poucos entram em detalhes quanto ao *hardware* utilizado. Oubbati et al. (2005) utilizam um computador embarcado de 2,6 GHz, uma grande evolução em relação a Feng et al. (1989), que utilizavam um computador Motorola 68000 com aproximadamente um milésimo da capacidade computacional daquele. Takemura et al. (2007) e Loh et al. (2003) utilizam computadores externos, envolvendo atrasos na comunicação entre tais computadores e o robô. Carrasco and da Silva (2016) apresentam um sistema que implementa um filtro de Kalman em um microprocessador Arduino UNO, uma alternativa de baixo custo. Diversos trabalhos mais recentes, como Krinkin et al. (2015), 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, Loh et al. (2003) 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

Conforme mencionado nas seções anteriores, o robô construído possui três rodas em uma configuração simétrica. Apesar da falta de redundância – pois se alguma das rodas falhar se perde a holonomicidade –, robôs omnidirecionais com 3 rodas (TOMR) são utilizados com mais frequência por serem mais simples de se implementar, apresentarem custo mais baixo (pois motores e rodas são responsáveis por x% do custo do projeto, conforme o Apêndice A), e uma certa economia de peso.

As rodas utilizadas medem 58 mm de diâmetro, com estrutura em plástico e dez roletes emborrachados, mostrando boa capacidade de carga para os fins de demonstração do projeto. Cada roda é acionada por um motor de corrente contínua com caixa de redução de relação 1:34, com uma velocidade nominal no eixo de saída de 210 rpm. A fixação das rodas no motor foi realizada conforme o Apêndice B **n sei se vai ter essa frase aqui**. Incluso no motor está um *encoder* de quadratura, que permite a leitura da velocidade da roda e da direção de rotação. Com a relação de redução, se tem que para cada revolução da roda se tem 341.2 pulsos do sensor Banggood.com (2017).

Além da utilização dos *encoders* para implementação da odometria, também foi instalada na estrutura uma bússola, para garantir uma medida absoluta da orientação do robô maior parte do algoritmo de localização. O modelo utilizado é a HMC5883L, em um módulo integrado (breakout board?), que utiliza comunicação pelo protocolo I2C e tem precisão de 2 graus (Honeywell (2013)). Devido ao baixo custo dos componentes utilizados e da modularidade do protocolo de comunicação (Semiconductors (2000)), também foi adicionada uma unidade de medidas inerciais MPU6050, que possui acelerômetro e giroscópio em torno dos três eixos utilizados (Invensense (2013)).

falar da bateria falar da bateria falar da bateria falar da bateria falar da bateria falar da bateria. Ligados à bateria, se tem dois reguladores de tensão MH-MINI-360 CITARDATASHEET, um trabalhando em 5V para alimentar o computador principal e os sensores, e outro com saída configurada em 6 V, para alimentar os motores e os drivers.

O acionamento dos motores se dá por um circuito de pontes H. Há duas destas placas, e cada uma pode acionar dois motores. Assim, se tem a possibilidade de utilizar mais um motor em trabalhos futuros. Talvez eu devesse citar as especificações dessas coisas, certo? Cada driver é alimentado com a tensão regulada de 6 V, e o computador utilizado comanda o chaveamento via PWM.

Power pins The maximum permitted current draw from the 3.3 V pins is 50 mA.

Maximum permitted current draw from the 5 V pin is the USB input current (usually 1 A) minus any current draw from the rest of the board.[18]

Model A: 1000 mA - 500 mA -> max current draw: 500 mA Model B: 1000 mA - 700 mA -> max current draw: 300 mA

Todo o processamento é realizado por um **Raspberry Pi**, 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 (Upton and Halfacree (2014)).

Para unir todos os componentes descritos, se projetou uma estrutura central, como um chassi. Tal estrutura pode ser vista na Figura 4. No centro geométrico da estrutura e na periferia, próximo a uma das rodas, foram feitos dois orifícios que devem acomodar uma caneta hidrográfica cada. Assim, durante a fase de testes, se pode acompanhar graficamente a evolução cinemática do robô. Devido a localização central de uma das canetas, todos os componentes foram instalados na periferia da estrutura. Se tomou ainda o cuidado de instalar o CI de acelerômetro e giroscópio o mais próximo ao centro possível, para que as componentes de aceleração centrípeta dos movimentos com componentes de rotação não influenciassem demasiado nos resultados. A IMU poderia ter sido colocada no centro geométrico, e este erro poderia ser introduzido no traço da caneta. No entanto, como a odometria e localização dependem muito mais dos sensores montados nos motores do que da IMU, se preferiu manter a caneta no centro, mantendo o MPU6050 o mais próximo possível. A bússola também foi montada relativamente próxima ao centro do robô, se tomando o cuidado de manter a mesma orientação dos eixos.

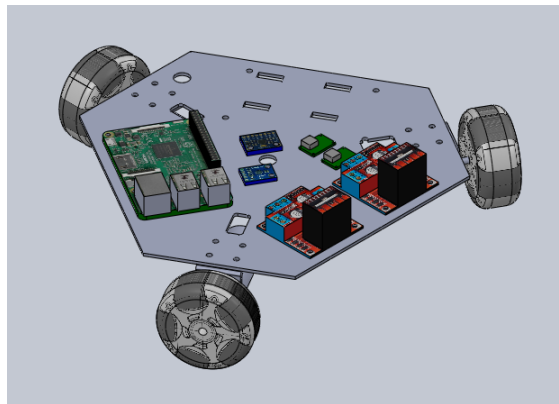


Figura 4: Chassi projetado. TROCAR A IMAGEM.

O custo de aquisição dos componentes relatados pode ser visto detalhado no [hyperref\[sec:custo\]](#) Apêndice A. Cabe ressaltar que todos os itens foram comprados em dobro, para realizar a montagem de dois robôs para futuros trabalhos no LAMECC (Laboratório de Mecatrônica e Controle), e que o chassi foi usinado no próprio laboratório com equipamento próprio, sendo computado apenas o custo do material utilizado.

Siegwart et al. (2011), pg 97: sensores.

4 Desenvolvimento Teórico

Primeiramente, se definem dois sistemas de coordenadas. O primeiro, (x_I, y_I) , é o sistema de coordenadas global, fixo no ambiente. O segundo, (x_R, y_R) , está centrado no próprio robô. Ainda se pode definir o ângulo θ como a orientação do robô – ou seja, o ângulo entre os dois sistemas de coordenadas. Tal relação pode ser vista na Figura 5, e a transformação de um sistema para o outro é descrita na Equação 1, conforme Siegwart et al. (2011) e Ritter (2016).

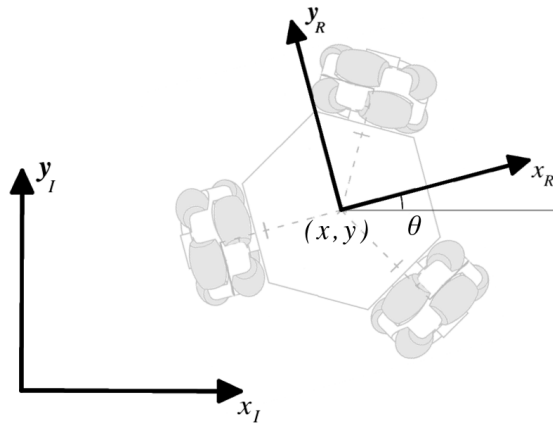


Figura 5: Sistemas de coordenadas global I e relativo ao centro do robô R.

Fonte: Adaptado de Ritter (2016)

$$\begin{pmatrix} x_I \\ y_I \\ \theta \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_R \\ y_R \\ \theta \end{pmatrix} \quad (1)$$

O último termo da Equação 1 também pode ser descrito como q_R , e o vetor de velocidades $[v_x, v_y, \omega_z]^T$, centrados no sistema de coordenadas do robô, é \dot{q}_R . Com o objetivo de mapear a velocidade de giro das rodas $\dot{\phi} = [\dot{\phi}_1, \dot{\phi}_2, \dot{\phi}_3]^T$ às velocidades \dot{q}_R , se utiliza a modelagem cinemática apresentada por Siegwart et al. (2011), com as referências apresentadas na Figura 6. Na figura, A mesma modelagem é utilizada por Ritter (2016), porém com outra sequência e sentido de giro para as rodas.

Assim, para um robô com 3 rodas dispostas em simetria radial em torno do centro da estrutura, a cinemática direta é dada pela Equação 2. Diversos autores utilizam variações da mesma modelagem (Rojas and Förster (2006), Pin and Killough (1994), entre outros). 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).

EM CONSTRUÇÃO DAQUI PRA BAIXO:

ODOMETRIA: Lynch and Park (2017)

Samani et al. (2007): 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. 90% encoders, 10% visão computacional. Adicionam um modelo de ruído dos encoders à estimativa.

TRAJETÓRIA: Lynch and Park (2017), capítulo 9.

CONTROLE: Rojas and Förster (2006): 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: Lynch and Park (2017)

Samani et al. (2007): 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.

Siegwart et al. (2011): 5.6.3.1 Introduction to Kalman filter theory

5 Implementação dos algoritmos

SEÇÃO EM CONSTRUÇÃO

O código de Ritter (2016) foi desmembrado em módulos, para separar a implementação já realizada da cinemática direta e inversa dos módulos de comunicação com o simulador utilizado. Foram escritos novos módulos para realizar a interface da Raspberry com os motores, sensores e periféricos em geral, e com essa modularidade se pode até fazer uma biblioteca para arduino blabalbalabballab.

COLOCAR DIAGRAMA DAS BIBLIOTECAS

Foi utilizada pigpio Joan (2017).

Além de utilizar os comandos fornecidos por Ritter (2016), foram implementados os modos de movimentação citados por Loh et al. (2003): 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.

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 (Upton and Halfacree (2014)).

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 (Upton and Halfacree (2014)) quanto pelos componentes em geral utilizados (Invensense (2013) e a bússola e o arduino se eu usar). Com este protocolo, descrito em Semiconductors (2000), 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.

O *fast mode* é suportado pelo Raspberry Pi 3 B+.

6 Avaliação experimental

SEÇÃO EM CONSTRUÇÃO

Se notou que o acionamento dos motores depende de alguns fatores. Variando a frequência dos PWMs mudou o q?

O protótipo foi acionado sobre um papel enorme, com duas canetas de cores distintas instaladas nos orifícios destinados a tal. Seguem os resultados:

7 Resultados

DEVE TER EM TORNO DE DEZ PAGINAS

Se pode usar isso aqui se pá: XinuPi is the port of Embedded Xinu to the Raspberry Pi. XinuPi provides a simple, lightweight operating system for the Raspberry Pi that contains several orders of magnitude fewer lines of code than the Linux-based software stacks that normally run on the device. Its purpose is to provide an inexpensive, convenient platform for various areas of computer science education at a University level, including operating systems, embedded systems, networking, and programming languages. Another goal of XinuPi is to document some of the Raspberry Pi's hardware that has, until this point, been poorly documented or even undocumented. This includes the documentation here as well as XinuPi source code and the documentation generated from comments in it.

8 Conclusão e Trabalhos Futuros

EM CONSTRUÇÃO cd Processamento na GPU.

9 Referências

- Banggood.com, 2017. Specifications for 6v 210rpm dc motor with encoder and geared reduction. <https://www.banggood.com/6V-210RPM-Encoder-Motor-DC-Gear-Motor-with-Mounting-Bracket-and-Wheel-p-1044064.html>, acessado em 03/10/2017.
- Campion, G., Bastin, G., Dandrea-Novel, 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.
- Honeywell, 2013. Hmc5883l 3-axis digital compass ic. Acessado em 17/11/2017.
URL https://cdn-shop.adafruit.com/datasheets/HMC5883L_3-Axis_Digital_Compass_IC.pdf
- 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
- 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.
- Joan, 2017. Reference for pigpio c interface. Acessado em 27/11/2017.
URL <http://abyz.me.uk/rpi/pigpio/cif.html>
- 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 1 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\$ 120,00	1	R\$ 120,00
Reguladores de Tensão	R\$ 4,98	4	R\$ 19,96
Parafusos	R\$ 14,00	X	R\$ 14,00
Diversos	R\$ 30,00	X	R\$ 30,00
Custo Total:			R\$ 946,05

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