



Pedro Ramon de Mello Silva

**Localização de Robô em Ambiente Interno
Utilizando um Dispositivo Móvel Baseado
no Sistema Operacional *Android*
para Navegação Inercial**

47/2015

CAMPINAS
2015



UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA

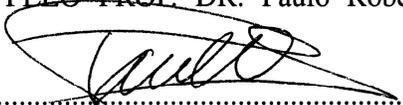
Pedro Ramon de Mello Silva

**Localização de Robô em Ambiente Interno
Utilizando um Dispositivo Móvel Baseado
no Sistema Operacional *Android*
para Navegação Inercial**

Dissertação de Mestrado apresentada à Faculdade de Engenharia Mecânica da Universidade Estadual de Campinas como parte dos requisitos exigidos para obtenção do título de Mestre em Engenharia Mecânica, na Área de Mecânica dos Sólidos e Projeto Mecânico.

Orientador: Prof. Dr. Paulo Roberto Gardel Kurka

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO Pedro Ramon de Mello Silva, E ORIENTADO PELO PROF. DR. Paulo Roberto Gardel Kurka.



.....
ASSINATURA DO ORIENTADOR

CAMPINAS
2015

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca da Área de Engenharia e Arquitetura
Elizangela Aparecida dos Santos Souza - CRB 8/8098

Si38L Silva, Pedro Ramon de Mello, 1987-
Localização de robô em ambiente interno utilizando um dispositivo móvel baseado no sistema operacional android para navegação inercial / Pedro Ramon de Mello Silva. – Campinas, SP : [s.n.], 2015.

Orientador: Paulo Roberto Gardel Kurka.
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de Engenharia Mecânica.

1. Robótica. 2. Sistema inercial de navegação. 3. Android (Programa de computador). I. Kurka, Paulo Roberto Gardel, 1958-. II. Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. III. Título.

Informações para Biblioteca Digital

Título em outro idioma: Localization of a robot in indoor environment using a mobile device based on android operating system for inertial navigation

Palavras-chave em inglês:

Robotics

Inertial navigation

Android (Electronic resource)

Área de concentração: Mecânica dos Sólidos e Projeto Mecânico

Titulação: Mestre em Engenharia Mecânica

Banca examinadora:

Paulo Roberto Gardel Kurka [Orientador]

Gilmar Ferreira Batalha

Niederauer Mastelari

Data de defesa: 20-02-2015

Programa de Pós-Graduação: Engenharia Mecânica

UNIVERSIDADE ESTADUAL DE CAMPINAS
FACULDADE DE ENGENHARIA MECÂNICA
COMISSÃO DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA
DEPARTAMENTO DE SISTEMAS INTEGRADOS

DISSERTAÇÃO DE MESTRADO ACADÊMICO

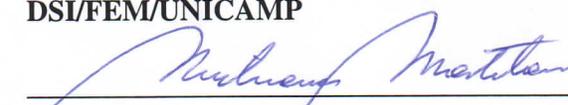
**Localização de Robô em Ambiente Interno
Utilizando um Dispositivo Móvel Baseado
no Sistema Operacional *Android*
para Navegação Inercial**

Autor: Pedro Ramon de Mello Silva
Orientador: Paulo Roberto Gardel Kurka

A Banca Examinadora composta pelos membros abaixo aprovou esta Dissertação:



Prof. Dr. Paulo Roberto Gardel Kurka, Presidente
DSI/FEM/UNICAMP



Prof. Dr. Niederauer Mastelari
DSI/FEM/UNICAMP



Prof. Dr. Gilmar Ferreira Batalha
POLI/USP

Campinas, 20 de Fevereiro de 2015.

Dedicatória

Dedico este trabalho aos meus professores de graduação, Max Trindade, José Real e Alex Guedes, por terem me inspirado a entrar no mundo da ciência, e ao meu falecido avô, Roberto Pedreira de Mello.

Agradecimentos

À Deus, minha fonte de força, fé e determinação. E como em todos os dias, agradeço pelo dom da vida.

Ao meu orientador, Prof. Dr. Paulo Roberto Gardel Kurka.

A toda minha família.

Aos amigos e companheiros de laboratório.

A CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo indispensável apoio financeiro.

The scientific man does not aim at an immediate result. He does not expect that his advanced ideas will be readily taken up. His work is like that of the planter — for the future. His duty is to lay the foundation for those who are to come, and point the way.

Nikola Tesla

Resumo

Mello Silva, Pedro Ramon de. Localização de Robô Móvel em Ambiente Interno Utilizando um Dispositivo Android para Navegação Inercial. 2015. 55p. Dissertação (Mestrado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

A navegação autônoma de robôs tem sido alvo de pesquisa. Um dos problemas a ser resolvido em navegação é a localização de um robô móvel enquanto este se locomove, ou seja, estimar o deslocamento e orientação para calcular a trajetória realizada. Um sistema de navegação inercial é um método de realizar a localização do robô utilizando sensores inerciais (acelerômetro, giroscópio e magnetômetro). Este trabalho apresenta a utilização de um dispositivo móvel baseado no sistema operacional *Android*, como UMI (Unidade de Medida Inercial), para um desenvolvimento de um sistema de navegação inercial de um robô terrestre. Para o cálculo de orientação, foi aplicado o filtro complementar para melhorar o desempenho da estimativa de orientação calculada pelo acelerômetro/magnetômetro combinado com o giroscópio. Para estimativa de deslocamento, aplicou-se integral dupla nos dados de aceleração e propõe-se uma metodologia de movimento modulado para minimizar os erros de posição e desvios no sinal de velocidade devido ao processo de integração numérica e erros de devido a componente de corrente contínua do acelerômetro.

Palavras-chave: Robótica, Sistema inercial de navegação, Android (Programa de computador).

Abstract

Mello Silva, Pedro Ramon de. Localization of a Mobile Robot in Indoor Environment Using an Android Device for Inertial Navigation System . 2015. 55p. Dissertação (Mestrado). Faculdade de Engenharia Mecânica, Universidade Estadual de Campinas, Campinas.

The autonomous navigation of mobile robots have been the subject of many research. One of the problems to be solved in navigation, is the localization of a mobile robot as it moves around, in other words, to estimate the displacement and orientation to calculate the trajectory performed. An inertial navigation system is a method of performing the localization of the robot using inertial sensors (accelerometer, gyroscope and magnetometer). This work presents the use of a mobile device based on the Android platform, as IMU (Inertial Measurement Unit), for the development of an inertial navigation system of a robot. For guidance calculation, the complementary filter has been applied to improve the performance of guidance computed by accelerometer / magnetometer combined with the gyro. For displacement estimation, a double integral in the acceleration data was applied. Furthermore, a motion modulated methodology is proposed to minimize position errors and velocity drift due to numerical integration process and accelerometer offset errors.

Keywords: Robotics, Inertial navigation, Android (Electronic resource).

Lista de Ilustrações

1.1	Barco teleoperado de Tesla. (TESLA, 1898)	1
1.2	<i>Big Dog</i> da <i>Boston Dynamics</i> . (WOODEN ET AL., 2010)	2
2.1	Deslocamento(mm) versus tempo(t). (BARSHAN AND DURRANT-WHYTE, 1995)	6
2.2	Comparação de trajetória via encoder e via fusão sensorial (LEE ET AL., 2009)	7
2.3	Orientação filtrada(vermelho) e orientação pelo acelerômetro/magnetômetro(em azul) (AYUB ET AL., 2012)	8
2.4	IMU fixada no sapato do voluntário (FOURATI AND MANAMANNI, 2013)	9
2.5	Trajetoira percorrida (FOURATI AND MANAMANNI, 2013)	9
3.1	Sistema de coordenadas do <i>smartphone</i>	12
3.2	Sinal de aceleração amostrado no tempo	14
3.3	Velocidade estimada no tempo	15
3.4	Simulação do motor CC	18
3.5	Velocidades e acelerações do robô em x e y em relação a um referencial fixo	19
3.6	Aceleração identificada	21
3.7	Velocidade distorcida em cada ciclo	21
3.8	Fluxograma do algoritmo de estimação de deslocamento	22
3.9	Aceleração identificada e velocidade corrigida	23
3.10	Velocidade com desvio	23
3.11	Rotação em torno do eixo X	24
3.12	Rotação em torno do eixo Y	25
3.13	Rotação em torno de do eixo Z	25
3.14	Drift da orientação via giroscópio	28
3.15	Orientação via acelerômetro/magnetômetro contaminado por ruídos	28
3.16	Diagrama do filtro complementar	29
3.17	Orientação via filtro complementar	29
3.18	Orientação via filtro complementar (VARGAS, 2012)	30
4.1	Samsung <i>Galaxy S3 Mini</i> (MICHEL, 2014)	32
4.2	Placa IOIO-OTG. (BEN-TSVI, 2013)	34
4.3	Vista superior da plataforma robótica	34
4.4	Diagrama do Sistema Robótico	35

5.1	Driver de motor Sabertooth.	36
5.2	Circuito de condicionamento de sinal.	38
6.1	Plataforma Robótica	39
6.2	Trajectoria real via Kinovea	40
6.3	Aceleração em x	40
6.4	Aceleração em y	41
6.5	Orientação em torno de z	41
6.6	Velocidade em x com desvio	42
6.7	Velocidade em y com desvio	42
6.8	Trajectoria distorcida	43
6.9	Velocidade corrigida em relação ao eixo x	43
6.10	Velocidade corrigida em relação ao eixo y	44
6.11	Trajectoria real(via Kinovea) e estimada(movimento modulado)	44
6.12	Trajectoria real via Kinovea	45
6.13	Aceleração em x	46
6.14	Aceleração em y	46
6.15	Orientação com relação ao eixo z	47
6.16	Aceleração em relação ao eixo x com intervalos de integração.	47
6.17	Aceleração em relação ao eixo y com intervalos de integração.	48
6.18	Comparação da trajetória distorcida com a trajetória real	48
6.19	Velocidade corrigida relação ao eixo y	49
6.20	Velocidade corrigida relação ao eixo x	49
6.21	Comparação da trajetória estimada com a trajetória real	50

Lista de Abreviaturas e Siglas

$p(t)$	- Posição
	- Termos independentes
$v(t)$	- Velocidade
$a(t)$	- Aceleração
$\theta(t)$	- Ângulo de orientação
a_m	- Aceleração medida
a_g	- Aceleração da gravidade
A_l	- Aceleração linear
$\dot{\theta}_g$	- Velocidade angular via giroscópio
H	- Campo Magnético
R_i	- Matriz de Rotação em torno do eixo i
θ_g	- Orientação via giroscópio
θ_{am}	- Orientação via acelerômetro/magnetômetro
$\Delta_i(k)$	- Incremento de deslocamento
V_m	- Tensão média
T_{on}	- Tempo ligado
T_{off}	- Tempo desligado

SUMÁRIO

Lista de Ilustrações	xvii
Lista de Abreviaturas e Siglas	xxi
SUMÁRIO	xxiii
1 Introdução	1
1.1 Motivação	4
1.2 Objetivos	4
1.3 Organização do Texto	5
2 Revisão Bibliográfica	6
3 Sistema de Navegação Inercial para Odometria de Robô Móvel	11
3.1 Componentes da UMI	11
3.2 Técnica de Integração numérica para cálculo de deslocamento	13
3.3 Técnica de modulação de movimento	16
3.3.1 Algoritmo de Estimação de Deslocamento	20
3.4 Estimação de orientação baseada em fusão sensorial	24
3.4.1 Orientação via matrizes de rotação	24
3.4.2 Filtro Complementar para Fusão Sensorial	27
3.5 Geração da Trajetória	29
4 Interface entre plataforma <i>Android</i> e Robô Móvel	31
4.1 O Dispositivo Móvel	31
4.2 Interface do dispositivo móvel com os periféricos externos	33
4.3 Plataforma Robótica	33
4.4 Diagrama do Sistema Robótico	34
5 Sistemas de Controle de Movimento	36
5.1 <i>Driver</i> de acionamento <i>Sabertooth</i>	36
5.2 Condicionamento e Geração de Sinal	37

6 Resultados Experimentais	39
6.1 Trajetória Linear com curvas	40
6.2 Trajetória Circular	45
7 Conclusão	51
Referências	52
ANEXOS	55
A – Artigos Publicados	55

1 Introdução

A aplicabilidade da robótica móvel tem ganhado popularidade especialmente em aplicações de inspeção de ambientes hostis, em indústrias e na exploração espacial, onde estão presentes robôs teleoperados ou autônomos (SIEGWART ET AL., 2011). O primeiro robô móvel teleoperado, por ondas de rádio mostrado na Figura 1.1 foi apresentado por Tesla (1898), que controlava um pequeno barco utilizando um controle remoto.

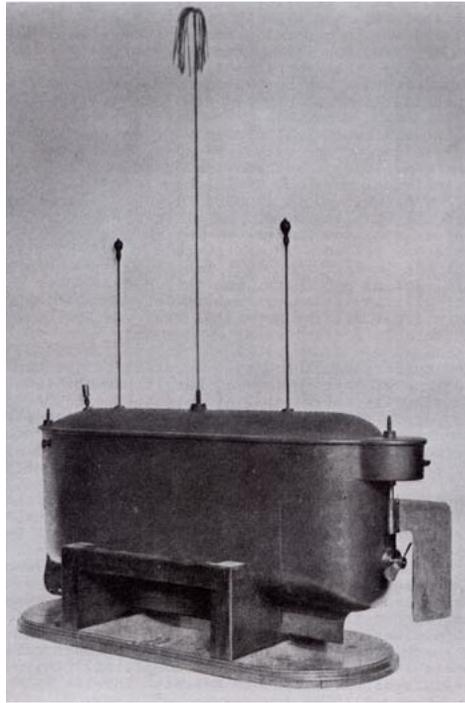


Figura 1.1: Barco teleoperado de Tesla. (TESLA, 1898)

Porém, a capacidade de um robô ser totalmente autônomo é a meta principal das pesquisas na área (JARVIS, 2010), onde ainda há poucos métodos consolidados para a realização desta tarefa. Atualmente, um dos robôs autônomos com aplicação militar é o *Big Dog* da *Boston Dynamics*, visto na Figura 1.2, que utiliza visão computacional, sensores inerciais e sensores a laser para a realização da sua tarefa de locomoção (WOODEN ET AL., 2010).

A navegação autônoma tem como requisito a capacidade de um robô móvel de identificar a sua postura (posição e orientação) no ambiente. Um dos métodos de realizar esta atividade é utilizando a odometria, que calcula a postura do robô pela integração da distancia percorrida e a taxa de variação angular. A utilização de um *encoder* nas rodas do robô é um dos métodos mais utiliza-



Figura 1.2: *Big Dog* da *Boston Dynamics*. (WOODEN ET AL., 2010)

dos em odometria, porém, sofre de erros sistemáticos devido a derrapagem e desalinhamento das rodas. Além de aplicações em robótica, a medição precisa da posição e orientação de uma trajetória desempenha um papel crítico em diversos outros campos de engenharia, incluindo aeroespacial e análise de movimento humano, onde a navegação inercial é uma alternativa para a realização dessas tarefas (FOURATI AND MANAMANNI, 2013). Navegação inercial é uma técnica de navegação autônoma onde as grandezas cinemáticas são medidas por acelerômetros, giroscópios e em alguns casos, magnetômetros, para estimar posição e/ou orientação de um objeto relativo a um ponto de partida (WOODMAN, 2007). Um dos primeiros trabalhos a utilizar navegação inercial foi de Robert Goddard, originalmente desenvolvida para foguetes utilizando sistemas de giroscópio (GODDARD, 1914). Tipicamente, unidades de medida inerciais (UMI) contém tipicamente acelerômetros, giroscópios e magnetômetros, cada um com três eixos ortogonais, a fim de medir aceleração linear, velocidade angular e campo magnético, respectivamente. Recentes avanços na construção de dispositivos MEMS (*micro-machined electromechanical systems*), tem facilitado a manufatura de sistemas de navegação inercial pequenos e leves, como os que se encontram embarcados em smartphones. O uso de uma Unidade de Medida Inercial para navegação de robôs móveis possui algumas particularidades em relação a outros tipos de odometria, como a visual e utilizando *encoder*. Primeiramente, o processamento dos dados inerciais oriundos de uma UMI para nave-

gação é menos custoso computacionalmente se comparado a um processamento de imagens (LI AND MOURIKIS, 2013). Segundo, os alguns erros sistemáticos e não-sistemáticos, como desalinhamento de rodas e derrapagem presentes na odometria com *encoder*, não existem. Além disso, apesar de uma UMI ser tipicamente mais custosa que uma câmera, ainda assim, é mais barata que outros sensores como laser e radar.

A navegação inercial é baseada na resolução de dois problemas:

- Estimação de Deslocamento Linear- O acelerômetro contido na UMI tem como saída dados de aceleração que podem ser integrados numericamente para se obter uma velocidade estimada, e integrando da mesma forma, o deslocamento. Porém, a aceleração mensurada pelo acelerômetro é contaminada por níveis CC (Corrente Contínua) e por ruídos de alta frequência, e conseqüentemente, causando erros ilimitados nos processos de integração numérica ao longo do tempo.
- Estimação de Orientação- Uma UMI que contém acelerômetros, giroscópios e magnetômetros, tem basicamente duas formas de se obter a sua orientação no espaço. A primeira, se dá pela integração numérica dos dados de velocidade angular medidos pelo giroscópio. Porém, o giroscópio é contaminado de uma componente CC que causa erros de integração ao longo do tempo. A segunda, pela combinação do acelerômetro com o magnetômetro, que apesar de não conter erros a longo prazo como na orientação via giroscópio, é consideravelmente ruidosa devido a interferências eletromagnéticas sofridas pelo magnetômetro e pelos ruídos inerentes ao acelerômetro. É possível com base nessas duas abordagens a fusão entre sensores para uma otimização na estimação de orientação, utilizando técnicas como filtro de Kalman e Filtro Complementar (WALTER T HIGGINS, 1975)

Assim, a partir dos dados de deslocamento e orientação, é possível construir a trajetória desenvolvida pela UMI embarcada em um sistema mecânico. A presente dissertação utiliza a UMI de um *smartphone* de baixo custo baseado no sistema operacional *Android* embarcado em um robô móvel, para realizar sua odometria inercial no cumprimento de tarefas de localização em ambiente de interiores.

1.1 Motivação

Para a realização da tarefa de localização em robôs móveis é comum a aplicação de sensores como *GPS*, *laser*, sonar, câmera e a combinação destes. Porém, muitas vezes essas abordagens podem ser custosas tanto economicamente quanto computacionalmente. Tendo em vista os aspectos expostos acima, este trabalho visa aplicar uma tecnologia que possa resolver um problema de localização de forma satisfatória, buscando ao mesmo tempo, a minimização de custos. Para atender este desafio, optou-se por adotar um Sistema de Navegação Inercial, usando como UMI(Unidade de Medida Inercial) um dispositivo móvel baseado na plataforma *Android*. O custo de um dispositivo móvel *Android* é consideravelmente baixo se comparado a outras tecnologias citadas acima para tarefas de localização em ambiente interno. Além disso, um dispositivo móvel *Android* pode ser programado na linguagem Java e comunicar-se facilmente com outros periféricos eletrônicos. A navegação de um robô móvel utilizando apenas uma UMI é insatisfatória para maioria das aplicações estudadas devido a alguns problemas intrínsecos, visto que apesar de um sistema de navegação inercial oferecer uma boa estimativa a curto prazo, sem uma referencia externa de correção, ele está sujeito a erros de posição e orientação sem limites (CHO ET AL., 2011). O processo de integração numérica dos dados de aceleração das UMI's, pode apresentar grandes erros de estimação da velocidade e da posição. O presente trabalho visa assim, propor uma metodologia para que os erros de estimação de velocidade e posição sejam minimizados de forma que a tarefa de localização seja realizada de forma mais acurada, atendendo ainda as especificações de baixo custo de projeto. Estabelecem-se assim os objetivos do trabalho listados abaixo.

1.2 Objetivos

Para realizar as propostas descritas na motivação acima, propõe-se cumprir nesse trabalho os seguintes objetivos:

- Criar um aplicativo para a plataforma *Android* utilizando a Linguagem Java capaz de ler os dados do acelerômetro, giroscópio e magnetômetro e os processe em tempo real a fim de obter a orientação e aceleração linear desenvolvida pelo robô, e ainda, armazenar os dados obtidos na memória do dispositivo móvel.
- Realizar a interface entre o *smarphone* com periféricos eletrônicos externos.

- Desenvolver de um circuito eletrônico que condicione o sinal de atuação do *Android* para os amplificadores de potência dos motores do robô.
- Realizar testes experimentais de locomoção do robô, aplicando a técnica proposta de movimento modulado e importar os dados inerciais armazenados para o computador.
- Reproduzir a trajetória percorrida pelo robô no software *MATLAB* da *MathWorks* e comparar com a trajetória rastreada por uma câmera.

1.3 Organização do Texto

A dissertação é organizada da seguinte forma. O Capítulo 2 apresenta a revisão bibliográfica sobre aplicações de plataformas UMI em tarefas de navegação . O Capítulo 3 apresenta o método proposto de localização de um robô móvel, utilizando sensores inerciais. O Capítulo 4 apresenta a plataforma *Android* e as tecnologias necessárias para o seu interfaceamento com o sistema robótico. O Capítulo 5 apresenta os equipamentos eletrônicos desenvolvidos para a aplicação dos métodos apresentados, em uma plataforma robótica experimental. O Capítulo 6 apresenta os resultados de estimação de movimento da plataforma robótica experimental.

2 Revisão Bibliográfica

Sistemas de navegação inercial têm sido utilizados em uma ampla gama de aplicações, não só na robótica móvel, como também, para análise de movimento humano e em sistemas de navegação militares(mísseis, aeronaves). Neste capítulo serão apresentadas diferentes aplicações de sistemas de navegação inercial na realização de diferentes tarefas. No presente trabalho um sistema de navegação inercial será aplicado para resolução da problemática de localização de um robô móvel, apresentando uma proposta para minimizar os erros de estimação de deslocamento e aplicando fusão sensorial, utilizando um *smartphone* baseado na plataforma *Android* como UMI e acionamento dos atuadores.

Um dos trabalhos de Barshan and Durrant-Whyte (1995) que motivou o uso de sistemas inercias de navegação para robôs foi desenvolvido na Universidade de Oxford, Reino Unido. A pesquisa se concentra em determinar a orientação angular e o deslocamento de um robô móvel. Os sensores utilizados para estimar a orientação são dois giroscópios, e para posição, foi utilizado um acelerômetro triaxial. O trabalho apresenta um modelo de regularização aplicando o Filtro de Kalman Extendido para compensar os erros de estimação de posição e orientação. Os resultados obtidos são comparados com um sistema de radar embarcado no robô. Um dos testes realizados para validar a estimação de posição, foi acelerar e desacelerar a plataforma robótica em uma distância de 300mm em relação ao eixo x para frente e para trás. A Figura 2.1 mostra o deslocamento estimado pelo método.

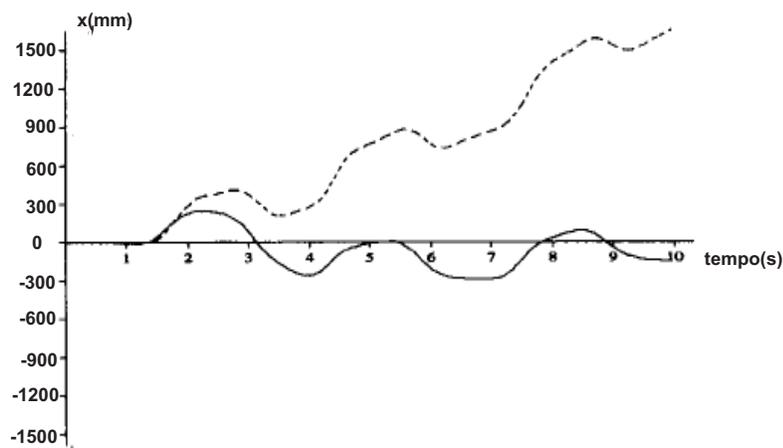


Figura 2.1: Deslocamento(mm) versus tempo(t). (BARSHAN AND DURRANT-WHYTE, 1995)

A linha tracejada mostra a posição estimada via integração dupla do sinal de aceleração e a linha contínua é o sinal de posição compensado pelo modelo de erro baseado no filtro de Kalman. No fim de dez segundos o erro de deslocamento foi de $-150.3mm$.

O trabalho de Lee et al. (2009) da Universidade Nacional de *Seoul*, Coreia do Sul, apresenta um sistema de navegação robótica em ambiente interno utilizando UMI (acelerômetro e giroscópio, ambos triaxiais), um sinalizador (*beacon*) ativo (quatro transmissores anexados no teto do ambiente, dois sensores ultrasônicos e um marcador no robô) e odômetro (*encoder*). Periodicamente, os sinalizadores anexados no teto recebem o sinal do marcador no robô para triangular sua posição absoluta. *Encoders* são aplicados para estimação de movimento a curto prazo. Erros não sistemáticos de estimação de posição devido aos *encoders* são compensados pela UMI, enquanto que os erros acumulados pela UMI são compensados utilizando o sinalizador ativo. Na Figura 2.2 mostra a rota do robô estimada via *encoder* por linhas tracejadas, enquanto que as linhas sólidas é a rota corrigida utilizando os sinalizadores e UMI. Nota-se que o erro final dado pelo *encoder* é grande comparado ao sistema de fusão sensorial aplicado.

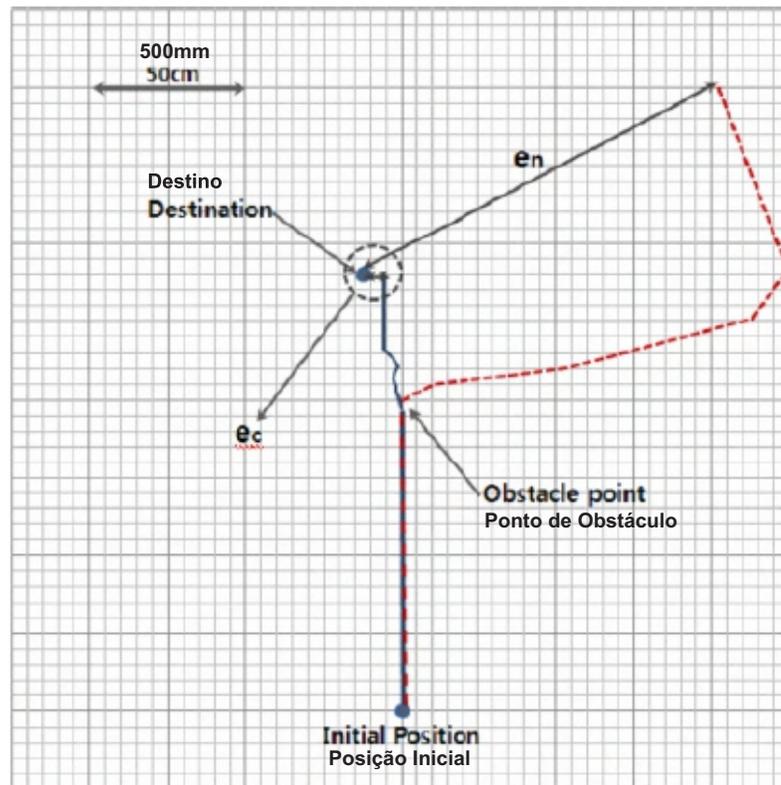


Figura 2.2: Comparação de trajetória via encoder e via fusão sensorial (LEE ET AL., 2009).

No trabalho de Ayub et al. (2012) realizado na Universidade de Lancaster, Reino Unido aplica-se a fusão sensorial para estimação de orientação angular de um *smartphone* baseado na plataforma Android. Na pesquisa, a estimação de orientação é obtida através da combinação do acelerômetro e o magnetômetro em fusão com a orientação obtida pelo giroscópio, implementando assim, um *DNRF (Drift Noise Removal Filter)*. A orientação estimada através da combinação acelerômetro/magnetômetro tem boa precisão enquanto o dispositivo móvel permanece estacionário, porém, sofre de interferência eletromagnética quando ruídos contaminam o sinal. A orientação obtida através da integração matemática dos dados do giroscópio(velocidade angular) tem boa precisão e responde bem a rápidas mudanças de orientação e não sofre de interferências eletromagnéticas, porém, sofre desvio devido aos erros propagados pela integração. A fusão sensorial aplicada minimiza o desvio e a interferência causada pelos dois métodos. Na Figura 2.3 mostra o sinal o angulo (inclinação) em azul sendo o sinal do acelerômetro/magnetômetro, e em vermelho o sinal de saída do filtro *DNRF*.

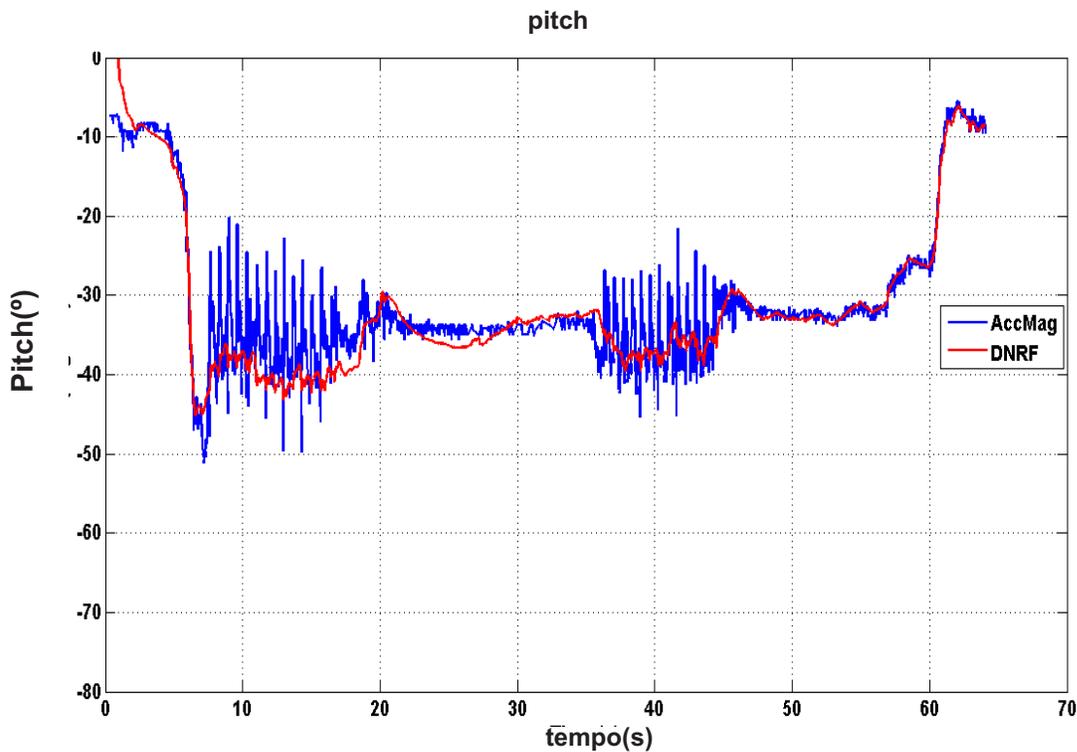


Figura 2.3: Orientação filtrada(vermelho) e orientação pelo acelerômetro/magnetômetro(em azul) (AYUB ET AL., 2012).

O sinal de saída do método *DNRF* é mais suave e tem resposta mais rápida comparado a orientação calculada pelo acelerômetro e magnetômetro.

No Instituto Nacional Politénico de Grenoble, França, foi desenvolvido um trabalho por Fourati and Manamanni (2013) para rastreamento de pedestre, utilizando uma UMI comercial. A IMU está fixada no pé do pedestre, conforme mostrado na Figura 2.4, para que seja medido seus parâmetros cinemáticos como orientação, posição, velocidade, aceleração e detecção de passo.



Figura 2.4: IMU fixada no sapato do voluntário (FOURATI AND MANAMANNI, 2013).

O algoritmo de filtragem do movimento do passo inclui métodos para estimação de posição e orientação, assim como detecção de passo. Foi implementado um filtro complementar, utilizando os sinais obtidos pelo acelerômetro, giroscópio e magnetômetro para realizar a estimação de orientação. A detecção de passo é realizada aplicando um detector de estados baseado nas medidas de aceleração. A detecção de passo é necessário para a aplicação do ZVU (*Zero Velocity Update*), um método que zera os valores de velocidade para quando o detector de estados retornar falso (pé em repouso) e integra a aceleração para quando do detector de estados retornar verdadeiro (pé em movimento). Foi realizado um experimento com um voluntário, percorrendo um trajeto retangular que pode ser visto na Figura 2.5.



Figura 2.5: Trajetória percorrida (FOURATI AND MANAMANNI, 2013).

A diferença entre o ponto de início e o ponto final foi de menos de $1m$ para $60m$ de deslocamento.

Observa-se em todos os trabalhos citados que a navegação inercial é uma ferramenta útil para o cálculo da trajetória de movimento. Ao mesmo tempo, percebe-se que existe ainda espaço para o desenvolvimento de aplicações e métodos que tornem mais robustos os resultados de navegação inercial

3 Sistema de Navegação Inercial para Odometria de Robô Móvel

Uma UMI (Unidade de Medida Inercial) é a componente fundamental em um sistema de navegação inercial para mensurar grandezas de interesse e posteriormente processá-las. Nas seguintes seções serão apresentados os fundamentos teóricos e técnicas aplicadas neste trabalho para realizar a odometria de um robô e reconstruir sua trajetória. As descrições dos sensores integrados na UMI serão mostrados na seção 3.1. A técnica de integração numérica para estimar o deslocamento partindo dos dados de aceleração será apresentada na seção 3.2. A técnica de modulação de movimento para que a estimação de deslocamento seja mais robusta será proposta na seção 3.3. O método de fusão sensorial para estimação de orientação do robô será apresentado na seção 3.4. As equações cinemáticas que descrevem trajetória do robô serão mostradas na seção 3.5.

3.1 Componentes da UMI

Uma UMI pode ser composta de acelerômetros, giroscópios e magnetômetros, representados em um sistema de coordenadas arbitrários conforme indicado pela Figura 3.1.

- Acelerômetro: Mede a aceleração da gravidade em m/s^2 aplicada a um ou mais eixos de um sistema de coordenadas, incluindo a força da gravidade. A aceleração medida pode ser representada pela Equação 3.1.

$$A_m = A_l + A_g \quad (3.1)$$

Onde A_m é a aceleração medida, A_l é aceleração linear do dispositivo e A_g é aceleração da gravidade.

- Giroscópio: Mede a velocidade angular em rad/s em torno de um ou mais eixos do sistema de coordenadas.
- Magnetômetro: Mede o campo geomagnético em μT em um ou mais eixos do sistema de coordenadas.

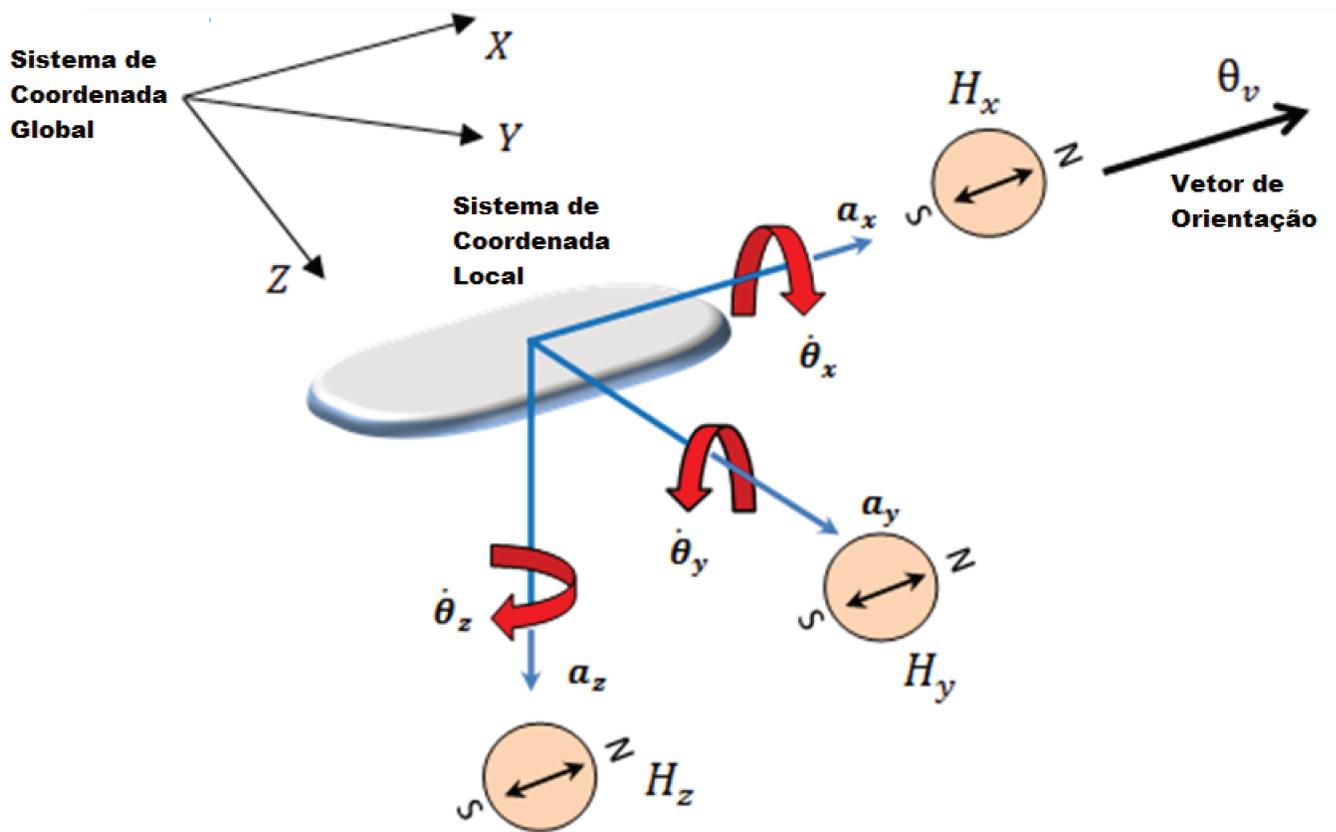


Figura 3.1: Sistema de coordenadas do *smartphone*.

As variáveis $a_x, a_y, a_z, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z, H_x, H_y$ e H_z apresentadas acima são referentes ao acelerômetro, giroscópio e magnetômetro, respectivamente.

O sinal medido pelo acelerômetro é contaminado por uma componente de corrente contínua (*offset*), ruídos e pela contribuição da aceleração da gravidade em relação ao eixo de medição (WOODMAN, 2007). Esses erros intrínsecos causam divergências dos valores reais de velocidade e posição via acelerômetro em processos de integrações numéricas. O processo de calcular a posição a partir de outro dado (aceleração, velocidade) é conhecido como *dead reckoning* (BONG-SU CHO, 2011), este é o fundamento do método aplicado neste trabalho para realização da odometria do robô. Além de deslocamento, outro objetivo da pesquisa é o cálculo dos ângulos de orientação do robô, que podem ser obtidos integrando a velocidade angular do giroscópio, ou utilizando o acelerômetro em conjunto com o magnetômetro (funcionando como uma bússola). Porém, cada método tem vantagens e desvantagens. A estratégia tomada foi realizar uma fusão sensorial das duas aproximações, para que se obtenha uma melhor estimativa da orientação angular do robô aproveitando o melhor de cada abordagem.

3.2 Técnica de Integração numérica para cálculo de deslocamento

Posição, velocidade e aceleração de uma partícula são relacionadas através das definições vistas nas Equações 3.2 abaixo

$$v = \frac{dx}{dt}, a = \frac{dv}{dt} \quad (3.2)$$

Onde $x(t)$, $v(t)$ e $a(t)$ são as posições, velocidades e acelerações de uma partícula, respectivamente.

Dessa forma, pode-se integrar as equações acima para achar a velocidade e posição em função do tempo. Porém, em processamento digital de sinais, os sinais de saída de sensores não são tratados como uma função analítica, mas como valores numéricos discretos. Dessa forma, é necessário a utilização de métodos numéricos de integração.

Considerando a função de valores discretizados na Figura 3.2

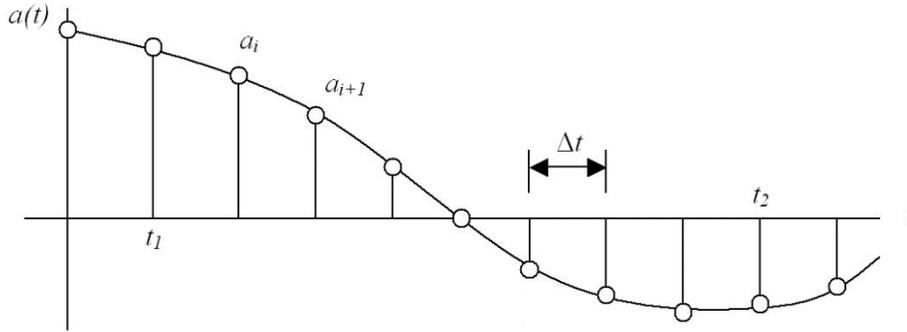


Figura 3.2: Sinal de aceleração amostrado no tempo.

É possível aproximar a área sob a curva entre dois pontos como um trapezóide de área $\frac{1}{2}(a_i + a_{i+1})\Delta t$.

Somando essas áreas a partir do tempo t_1 até t_2 , tem-se a Equação.

$$v(t_2) - v(t_1) = \int_{t_1}^{t_2} a(t)dt \simeq \sum_{t_1}^{t_2} \frac{1}{2}(a_i + a_{i+1})\Delta t \quad (3.3)$$

Similarmente, pode-se usar o mesmo procedimento para encontrar a posição pela Equação 3.4, integrando os valores de velocidade mostrados na Figura 3.3.

$$p(t_2) - p(t_1) = \int_{t_1}^{t_2} v(t)dt \simeq \sum_{t_1}^{t_2} \frac{1}{2}(v_i + v_{i+1})\Delta t \quad (3.4)$$

Os erros de integração em cada segmento(t_1 e t_2) são expressos pela Equação 3.5

$$Et = -\frac{(t_2 - t_1)^2}{12N^2} f(\delta) \quad (3.5)$$

Onde δ é um ponto dentro do intervalo t_1 e t_2 e N é o numero de intervalos. Esses erros

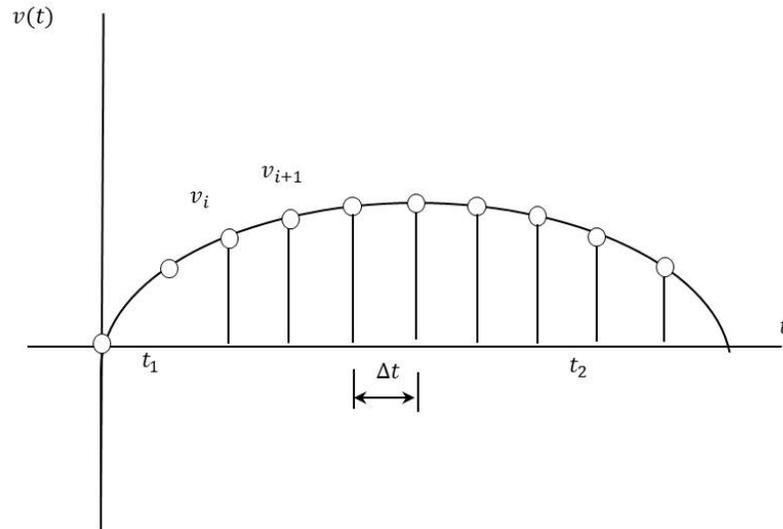


Figura 3.3: Velocidade estimada no tempo.

propagados ao longo do tempo prejudicam na estimação de velocidade e deslocamento.

O sinal de aceleração vindo de um acelerômetro é a soma da aceleração linear transiente com a aceleração da gravidade, como visto na Equação 3.1. Sendo assim, é necessário minimizar a contribuição da gravidade nas medidas de aceleração antes de aplicar os processos de integração matemática. Uma das formas de se alcançar este objetivo é aplicando um filtro passa-altas nos sinais de aceleração para atenuar suas componentes CC de baixa frequência, e assim, trabalhar apenas com a aceleração linear do dispositivo. Na Equação 3.6 é definida a função de transferência discreta de um filtro passa altas FIR (*Finite Impulse Response*) (SMITH, 1997).

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 - \sum_{k=1}^N a_k z^{-k}} \quad (3.6)$$

Porém, ainda que se minimize a contribuição de aceleração da gravidade nas medidas do acelerômetro, uma componente de corrente contínua permanecerá mesmo que muito pequena. Essa componente CC propaga erros nos processos de integração.

Considerando a integração dos dados de aceleração medidos pelo acelerômetro, definida pela Equação 3.1, tem-se a seguinte relação:

$$v(t) = \int_{t_1}^{t_2} (a_l(t) + a_g(t))dt \quad (3.7)$$

Supondo que a componente de corrente contínua devido a aceleração da gravidade, A_g , seja constante ao longo do tempo, tem-se:

$$v(t) = \int_{t_1}^{t_2} a_l(t)dt + a_g t \quad (3.8)$$

A Equação 3.8 mostra que o termo da aceleração da gravidade propaga um erro linearmente crescente ao longo do tempo para estimação de velocidade.

Equivalentemente, para a estimação de posição, tem-se:

$$p(t) = \int_{t_1}^{t_2} \int_{t_1}^{t_2} a_l(t) dt + a_g \frac{t^2}{2} \quad (3.9)$$

Logo, a perturbação pela componente A_g crescerá quadraticamente provocando erros estimação de deslocamento quando se integra a velocidade, e assim, erros na odometria do robô. As seções 3.3 e 3.4 apresentam uma proposta para minimizar os erros de localização causados pela componente CC presentes na medição do acelerômetro.

3.3 Técnica de modulação de movimento

Na prática, a integração de sinais de aceleração medidos por um acelerômetro para a obtenção de velocidade, resulta em um sinal de velocidade com tendência de desvio devido a uma componente de corrente contínua (*offset*) presente nos sinais de aceleração, causando assim, erro de estimativa de posição (LEE ET AL., 2009) . Para facilitar a correção do desvio de velocidade, e conseqüentemente, minimizar erro de estimação de deslocamento, é proposto um método de modulação AM do sinal aplicado aos atuadores do robô. O método é baseado na aplicação de um sinal

de tensão senoidal modulado pela amplitude do sinal de controle desejado, logo, a amplitude da senoide varia de acordo com o sinal de controle, segundo a equação abaixo.

$$V = V_c \sin(\omega t) \quad (3.10)$$

Onde V é a tensão aplicada nos terminais do motor CC, V_c é a tensão de controle e ω a frequência da senoide.

Considerando que os atuadores do robô em questão são motores de corrente contínua, podem então, ter seus modelos aproximados por um sistema linear invariante no tempo, conhecido na literatura (NISE, 2012), segundo a função de transferência abaixo.

$$\frac{\dot{\theta}(s)}{V(s)} = \frac{K}{(Js + b)(Ls + R) + K^2} \quad (3.11)$$

Onde b é a constante de atrito viscoso, J o momento de inercia do rotor, K a constante de torque, R a resistência da armadura e L a indutância elétrica.

A função de transferência da equação 3.11 relaciona a entrada em tensão (V) com a saída em velocidade angular ($\dot{\theta}$ rad/s). Foi realizada uma simulação no *software Matlab* da *Mathworks*, para ilustrar o comportamento de um motor de corrente contínua dado uma entrada senoidal, visto na Figura 3.4.

Como o modelo do motor CC é aproximado por um sistema linear e invariante no tempo, pode-se afirmar que para uma entrada senoidal, sua saída também será senoidal, apenas com fase e amplitude diferentes. Na Figura 3.4 é mostrado que para uma entrada de tensão senoidal no motor, tem-se uma saída senoidal da velocidade angular e da aceleração angular.

A plataforma robótica diferencial utilizada no trabalho mostrada na Seção 4.3, pode ter seu modelo cinemático que relaciona as velocidades lineares e angular do robô em relação ao um referencial fixo (SIEGWART ET AL., 2011), é mostrada na Equação 3.12

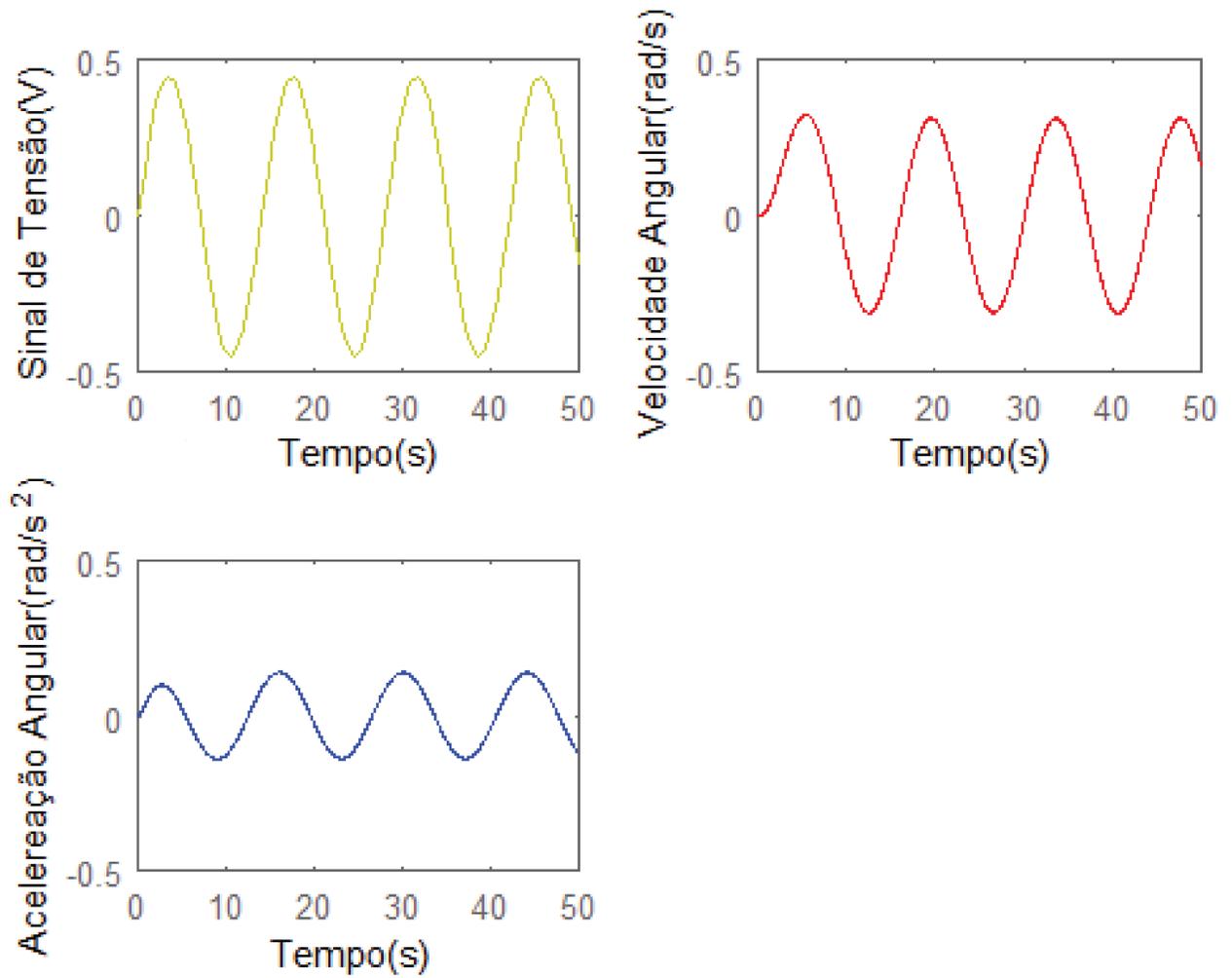


Figura 3.4: Simulação do motor CC.

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \quad (3.12)$$

Onde:

$$v(t) = \frac{1}{2} (v_r(t) + v_l(t)) \quad (3.13)$$

$$\omega(t) = \frac{1}{2} (v_r(t) - v_l(t)) \quad (3.14)$$

Visto que a derivada de um sinal senoidal é um sinal cossenoidal, é esperado que a derivada da velocidade (aceleração) também seja um sinal cossenoidal. Segundo o modelo cinemático apresentado, a aplicação de um sinal senoidal de velocidade nas rodas, devido ao sinal aplicado nos motores, faz com que a velocidade linear do robô também siga um perfil senoidal, e assim, a sua aceleração siga um perfil cossenoidal. como mostrado na Figura 3.5.

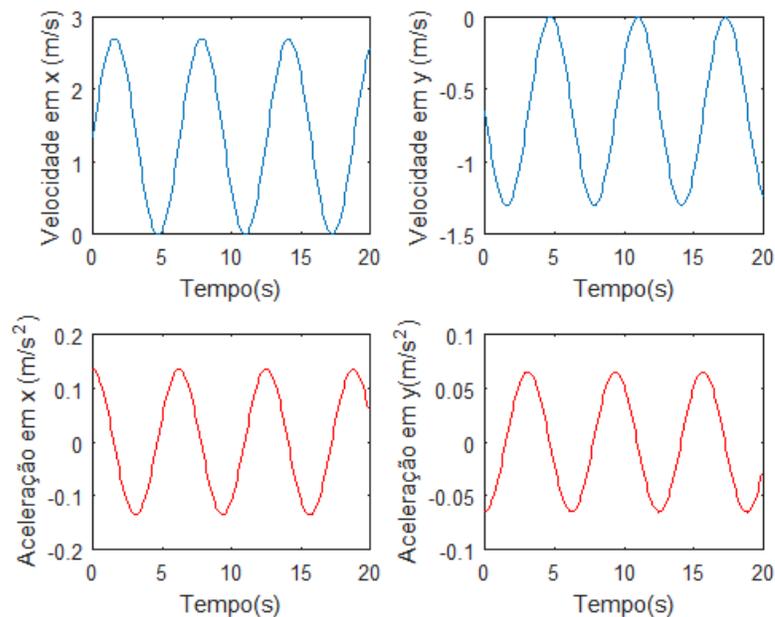


Figura 3.5: Velocidades e acelerações do robô em x e y em relação a um referencial fixo.

Dessa forma, é esperado que o acelerômetro embarcado no robô móvel meça acelerações com

um perfil cíclico na presença de uma entrada de tensão senoidal nos seus motores. A vantagem do robô desenvolver acelerações com um padrão conhecido, será aproveitada para corrigir o desvio de velocidade causado pela integração do sinal de aceleração. O algoritmo utilizado será apresentado na próxima Seção.

3.3.1 Algoritmo de Estimação de Deslocamento

Como apresentado na Seção anterior, a aplicação de um sinal de tensão senoidal nos terminais do motor CC do robô, faz com que o mesmo desenvolva um movimento com aceleração igualmente senoidal. O algoritmo de estimação de deslocamento proposto aproveita do sinal de aceleração cíclico para identificar cada passo que o robô realizou dentro de um período.

A Figura 3.6 exemplifica a identificação dos ciclos em um intervalo de aceleração. Um filtro de média móvel é aplicado a um intervalo de aceleração medida. Ao medir o sinal de aceleração, caso a média dentro do intervalo medido seja menor que um limiar definido, o algoritmo considera que dentro desse intervalo o robô está em repouso ou no fim de um ciclo de deslocamento, praticamente parado (sinal azul no gráfico). Porém, quando o robô se desloca e a média do intervalo do sinal de aceleração ultrapassar o limiar, então o ponto inicial deste intervalo é considerado como ponto inicial de integração. O limiar é aplicado para o ponto final do ciclo a fim de definir o intervalo do sinal a ser integrado (sinal vermelho no gráfico) quando o robô está em desaceleração.

A Figura 3.8 mostra o diagrama de blocos onde inclui o algoritmo apresentado acima. Ao se integrar os ciclos de aceleração identificados, obtém-se a velocidade desenvolvida dentro de cada intervalo. Porém, esta velocidade estimada é distorcida pelo desvio, como pode ser vista na Figura 3.7 devido a componente CC presente na aceleração.

Assim, esta velocidade é corrigida e armazenada em uma pilha. A correção de velocidade para cada ponto distorcido é dado pela seguinte equação.

$$\sum_{i=P_i}^{P_f} (R.([V_p(i) \quad t(i)] - [0 \quad t(p_i)])) \quad (3.15)$$

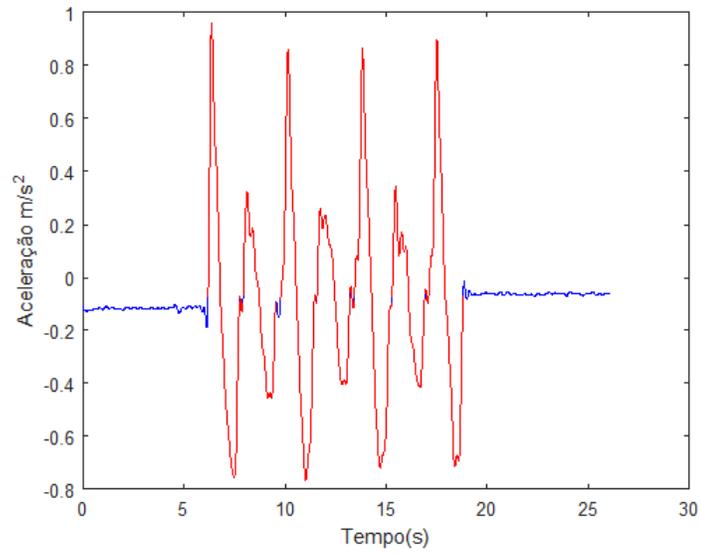


Figura 3.6: Aceleração identificada

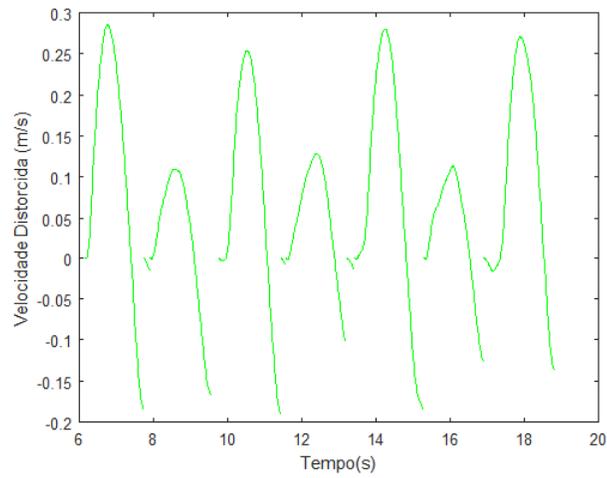


Figura 3.7: Velocidade distorcida em cada ciclo

Onde P_i é o ponto inicial de integração, P_f é ponto final, V_d é a velocidade distorcida no instante i , t o instante de tempo e R é definida pela matriz de rotação abaixo.

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (3.16)$$

Onde o ângulo θ é definido pela inclinação de cada ciclo de velocidade distorcida.

No entanto, se a média não foi identificada pelos limiares, o intervalo da média é zerado e armazenado na mesma pilha de velocidade corrigida. Finalmente, a pilha de velocidade corrigida é integrada e o deslocamento estimado é obtido.

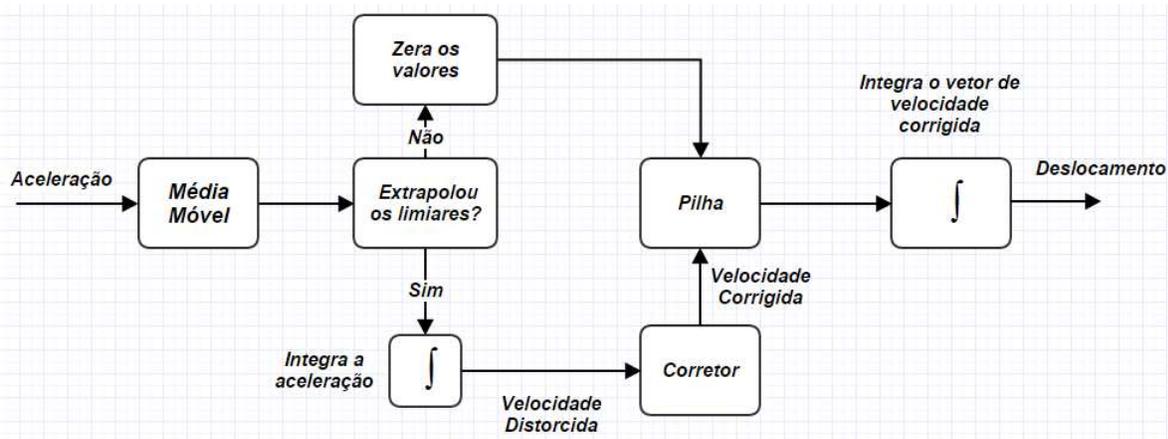


Figura 3.8: Fluxograma do algoritmo de estimação de deslocamento.

A Figura 3.9 mostra em azul, a aceleração a ser integrada no intervalo identificado, em preto, a aceleração não identificada, em vermelho, o vetor de velocidade corrigida e zerada no intervalo de aceleração preto, e em verde, a velocidade distorcida após a integração da aceleração identificada.

Na Figura 3.10, mostra o exemplo da velocidade distorcida via integral da aceleração com um *offset* de aproximadamente $0.12m/s^2$, o que geraria um erro de deslocamento ainda maior.

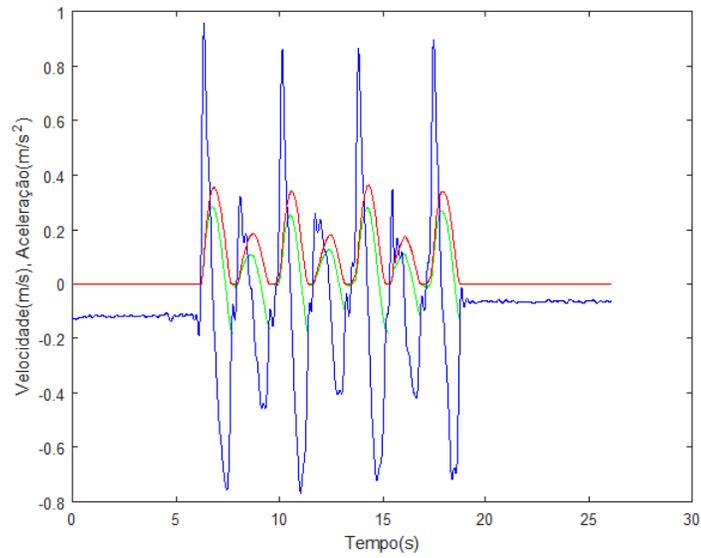


Figura 3.9: Aceleração identificada e velocidade corrigida.

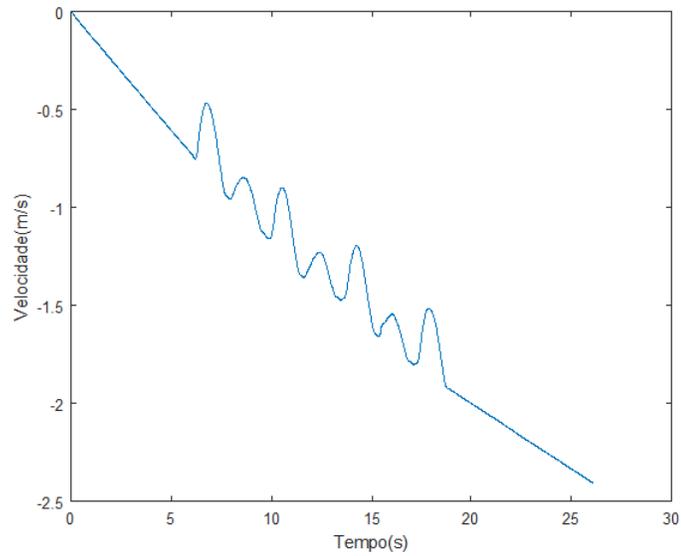


Figura 3.10: Velocidade com desvio.

3.4 Estimação de orientação baseada em fusão sensorial

3.4.1 Orientação via matrizes de rotação

Determinar os ângulos de orientação de um dispositivo móvel significa determinar os ângulos de rotação em relação a um referencial fixo arbitrário. O objetivo é relacionar o sistema de referencia de um *smartphone* a um referencial de navegação, no caso, o referencial da terra.

A transformação geométrica do referencial do *smartphone* para o referencial de navegação necessita de rotações em relação aos seus três eixos. Estes ângulos de rotação podem ser representados como ângulos de Euler (DIEBEL, 2006).

Uma das formas de se representar uma rotação é através do uso de matrizes de rotação. Definindo como guinada, a rotação em torno de z , inclinação, a rotação em torno de x e rotação, a rotação em torno de y , pode-se transformar um vetor de orientação do dispositivo móvel em relação ao referencial da terra através de multiplicação das matrizes de rotação. Os ângulos de rotação de x, y e z podem ser vistos nas Figuras 3.11, 3.12, 3.13, respectivamente, e suas matrizes de rotação são representadas pelas Equações 3.17, 3.18 e 3.19.

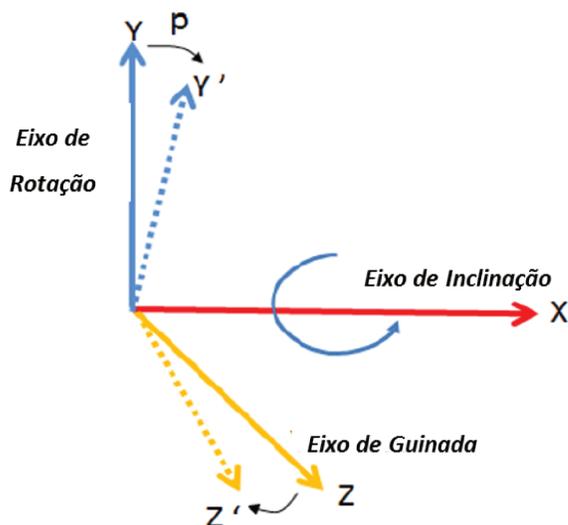


Figura 3.11: Rotação em torno do eixo X.

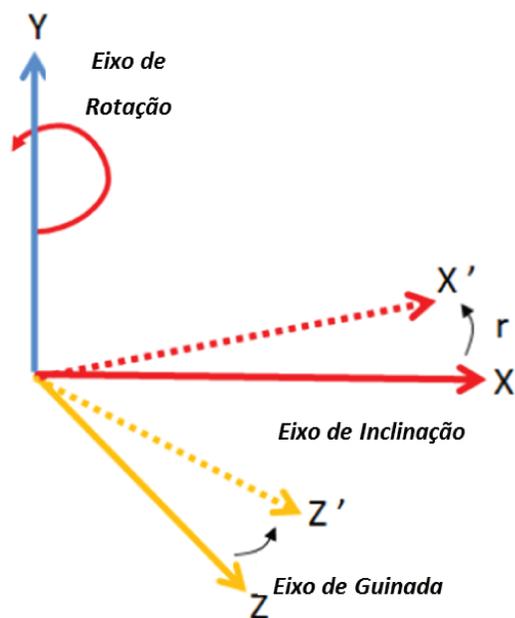


Figura 3.12: Rotação em torno do eixo Y.

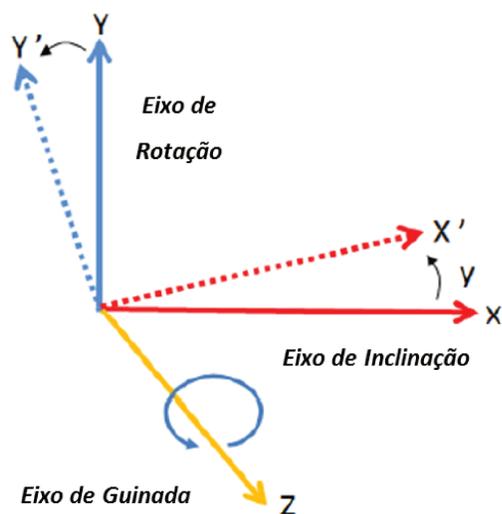


Figura 3.13: Rotação em torno de do eixo Z.

$$R_{x(k)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & \sin(\theta_x) \\ 0 & -\sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \quad (3.17)$$

$$R_{y(k)} = \begin{bmatrix} \cos(\theta_y) & 0 & -\sin(\theta_y) \\ 0 & 1 & 0 \\ \sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \quad (3.18)$$

$$R_{z(k)} = \begin{bmatrix} \cos(\theta_z) & \sin(\theta_z) & 0 \\ -\sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.19)$$

Ao multiplicar cada matriz, tem-se uma matriz de rotação que relaciona as rotações nos três eixos de um sistema de referencia em relação a um sistema de referencia fixo, Equação 3.20 abaixo.

$$R(k) = \begin{bmatrix} c_y \cdot c_z & c_y \cdot s_z & -s_y \\ s_x \cdot s_y \cdot c_z - c_x \cdot s_z & s_x \cdot s_y \cdot s_z + c_x \cdot c_z & s_x \cdot c_y \\ c_x \cdot s_y \cdot c_z + s_x \cdot s_z & c_x \cdot s_y \cdot s_z - s_x \cdot c_z & c_x \cdot c_y \end{bmatrix} \quad (3.20)$$

Onde:

$$c_x = \cos(\theta_x) \text{ e } s_x = \sin(\theta_x)$$

$$c_y = \cos(\theta_y) \text{ e } s_y = \sin(\theta_y)$$

$$c_z = \cos(\theta_z) \text{ e } s_z = \sin(\theta_z)$$

A partir da matriz de rotação final, é possível a extração dos ângulos de Euler para a determinação da orientação de um sistema de referência (*smartphone*) em relação a um referencial fixo (terra). No presente trabalho, para a estimação de orientação via acelerômetro/magnetômetro foi utilizado no aplicativo desenvolvido, os métodos *getRotationMatrix* e *getOrientation* da classe *SensorManager* do pacote para *Android* na linguagem Java. Para a estimação de orientação via giroscópio, é realizado um processo de integração em um intervalo de tempo pra conseguir o deslocamento angular dentro deste intervalo, e posteriormente, inserir em uma matriz de rotação. Ao

longo do tempo, as matrizes de rotação para cada intervalo são multiplicadas para para resultar na rotação absoluta. Para extrair os ângulos de orientação, utiliza-se o método *getOrientation*.

3.4.2 Filtro Complementar para Fusão Sensorial

A Determinação de orientação de um objeto em movimento usando UMI, tem sido desenvolvido no campo de sistemas de navegação inercial, como visto na bibliografia deste trabalho. É possível estimar a orientação de um objeto utilizando uma UMI embarcada basicamente de duas formas, utilizando um giroscópio, ou um magnetômetro combinado com um acelerômetro, considerando-se que todos os sensores são tri-axiais.

Os ângulos de orientação obtidos pelo giroscópio podem ser extraídos via integração matemática dos seus sinais de saída(velocidade angular), vista na Equação 3.21.

$$\theta_g = \int_{t_1}^{t_2} \dot{\theta}_g dt \quad (3.21)$$

Porém, o giroscópio sofre de *offset* no seu sinal medido. Essa componente CC do giroscópio é a saída média quando o mesmo não sofre alguma rotação, ou seja, o desvio da saída do valor real, (AYUB ET AL., 2012). O problema da componente CC leva ao desvio do sinal de orientação a longo prazo devido a erros de quantização e integração numérica que são propagados ao longo do tempo, similarmente ao mostrado na Equação 3.8. A orientação via giroscópio é mostrada na Figura 3.14

O magnetômetro, por sua vez, é capaz de medir o campo magnético nos eixos x,y,z . Dessa forma, é possível fazer uma combinação com o acelerômetro, funcionando como um vetor de gravidade, enquanto o magnetômetro funciona como uma bússola (LAWITZKI, 2014) .Essa combinação tem como saída os ângulos de orientação nos três eixos do sistema de referencia do UMI em relação a um referencial global. Porém, essa abordagem está sujeita a interferência magnética captada pelo magnetômetro e pelo alto sinal ruidoso do acelerômetro, prejudicando na estimação orientação. A orientação via acelerômetro/magnetômetro é vista na Figura 3.15

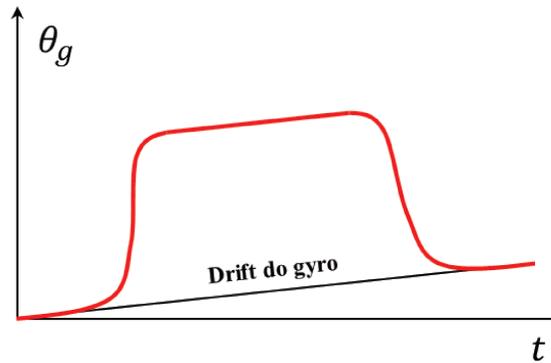


Figura 3.14: Drift da orientação via giroscópio.



Figura 3.15: Orientação via acelerômetro/magnetômetro contaminado por ruídos.

A estratégia aplicada neste trabalho, é a fusão das duas tecnologias, aproveitando o melhor de cada uma no objetivo de evitar o desvio de orientação devido ao giroscópio e o ruído devido ao acelerômetro/magnetômetro. A fusão é feita aplicando o Filtro Complementar, uma versão computacionalmente mais simples do Filtro de Kalman para realizar fusão sensorial (WALTER T HIGGINS, 1975), onde a saída do giroscópio será a orientação para curtos períodos de tempo, enquanto que a saída do acelerômetro/magnetômetro será a orientação a longo prazo. Isso equivale a aplicar um filtro passa alta no sinal de saída do giroscópio e um filtro passa baixa no sinal de saída do acelerômetro. O diagrama da Fig. 3.16 mostra a representação do Filtro Complementar no sinal dos sensores em questão.

A Equação 3.22 que define o filtro complementar é:

$$\theta = a.\theta_g + (1 - a).\theta_{am} \quad (3.22)$$

θ é a orientação filtrada, θ_g a orientação do giroscópio e θ_{am} a orientação calculada pelo acelerômetro/magnetômetro. a é o coeficiente do filtro digital, que pode ser aproximado pela Equação 3.23 abaixo.

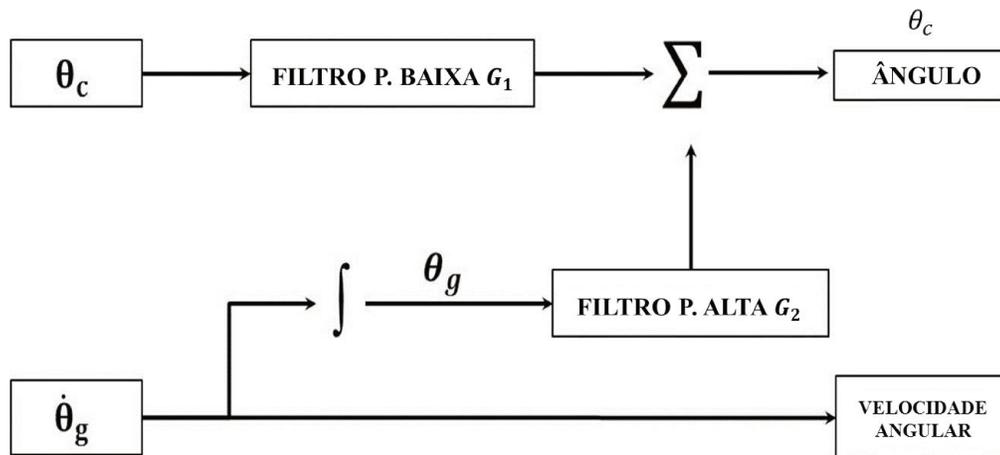


Figura 3.16: Diagrama do filtro complementar.

$$a = \frac{\tau}{\tau + T_s} \quad (3.23)$$

Onde T_s é o período de amostragem e a constante de tempo τ é relativo a duração do sinal.

Na Figura 3.17 é mostrado o sinal de orientação via fusão sensorial.



Figura 3.17: Orientação via filtro complementar.

3.5 Geração da Trajetória

Com a aquisição dos dados de orientação e deslocamento, é possível gerar uma trajetória estimada que reproduz a trajetória de executada pelo robô. O vetor de deslocamento global pode

ser calculado a partir de deslocamentos discretos $\Delta_x, \Delta_y, \Delta_z$, obtido de integrações sucessivas dos sinais de aceleração em um intervalo de tempo discreto Δ_t , tal qual o vetor de orientação discreto dado pelos ângulos $\Delta\theta_x, \Delta\theta_y, \Delta\theta_z$. A posição do robô no tempo $t = k(\Delta t)$ pode ser expresso na Equação 5.1.

$$\begin{bmatrix} X(k) \\ Y(k) \\ Z(k) \end{bmatrix} = \begin{bmatrix} X(k-1) \\ Y(k-1) \\ Z(k-1) \end{bmatrix} + \prod_{k=1}^n R(k) \begin{bmatrix} \Delta_x(k) \\ \Delta_y(k) \\ \Delta_z(k) \end{bmatrix} \quad (3.24)$$

Onde R é a matriz de rotação resultante da multiplicação sucessiva das matrizes de rotação em x, y e z , devida em 3.20. Dessa forma, pode-se desenhar a trajetória realizada pelo robô, como mostrado na Figura 3.18

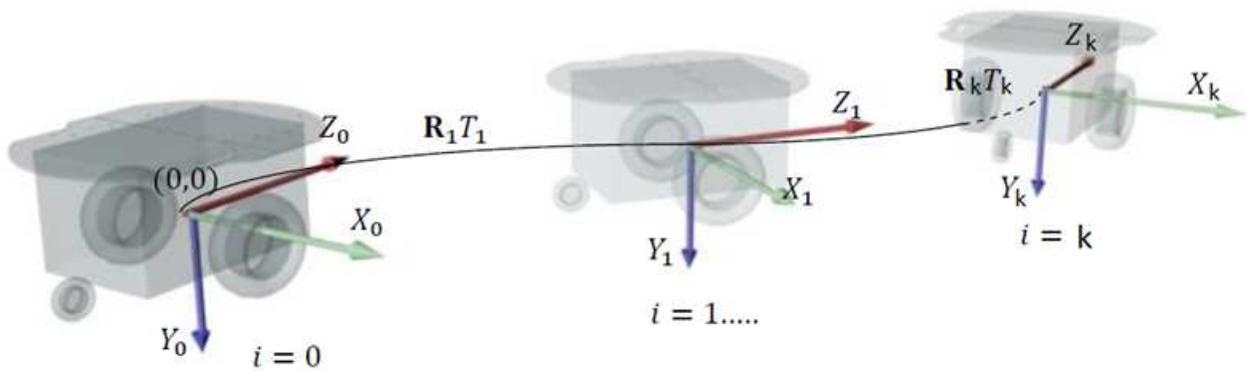


Figura 3.18: Orientação via filtro complementar (VARGAS, 2012).

4 Interface entre plataforma *Android* e Robô Móvel

Android é uma plataforma de código aberto para dispositivos móveis, que recentemente se tornou a plataforma mais utilizada em dispositivos *smartphones* (SHALA AND RODRIGUEZ, 2011). O Google disponibiliza o *Android SDK*(*Software Development Kit*) que habilita os desenvolvedores criarem aplicativos para a plataforma *Android*. O SDK contém as ferramentas, bibliotecas e API's necessários para criar diversas aplicações. A IDE(Integrated Development Environment) oficial para o desenvolvimento destes aplicativos é o ECLIPSE com o ADT(*Android Development Tools*), sendo possível construir códigos em Java e XML(*Extensible Markup Language*) para criar telas, ambientes gráficos, compilar, debugar e instalar os aplicativos desenvolvidos nos *smartphones*.

O dispositivo móvel baseado na plataforma *Android* é o componente central neste trabalho, pois realiza a tarefa de se conectar a dispositivos eletrônicos externos (e.g, *driver* de motores), enquanto faz a leitura e o processamento dos sensores inerciais embarcados. Foi desenvolvido um aplicativo em *Android* para realizar aquisição de dados dos sensores da UMI, a gravação na memória interna do dispositivo móvel destes dados e a interface entre o sinal de comando remoto(via rádio) e o acionamento do motores do robô, tal qual como realizar rotinas de reprodução de trajetórias pré-programadas.

4.1 O Dispositivo Móvel

Na aplicação realizada nesta pesquisa, o *smartphone* utilizado foi o Galaxy S3 Mini, da Samsung, visto na Figura 4.1, que foi selecionado pelo seu baixo custo de aquisição. Algumas características interessantes são:

- Baseado no sistema operacional *Android* 4.1 *Jelly Bean*.
- Processador Dual Core 1Ghz
- Memória RAM 1GB.
- Memória Interna de 16GB.

- Resolução de vídeo de *720 pixels*.
- Sensores Inerciais : Acelerômetro, Giroscópio e Magnetômetro.



Figura 4.1: Samsung *Galaxy S3 Mini* (MICHEL, 2014)

Os sensores inerciais(Acelerômetro e Giroscópio) embarcados são encapsulamentos do circuito integrado MPU-6050 da InvenSense Inc. A descrição de algumas características são apresentadas nas listas abaixo.

Acelerômetro:

- Medição de aceleração nos eixos x , y e z .
- Sensibilidade programável de 2g, 4g,8g e 16g.
- Operação normal de $500\mu A$

Giroscópio:

- Medição de velocidade angular nos eixos x , y e z .
- Faixa programável para 250, 500,1000 e $2000^\circ/s$.
- Corrente de operação de $3.6mA$.

4.2 Interface do dispositivo móvel com os periféricos externos

No trabalho desenvolvido, foi necessário encontrar uma solução para comunicar um *smartphone Android* com um hardware externo, com finalidade de enviar e receber sinais para sensores e atuadores e outros periféricos.

A solução adotada foi utilizar o IOIO-OTG, vista na Figura 4.2, uma placa microcontrolada desenvolvida pela *Sparkfun Electronics*, que pode agir como uma ponte entre o dispositivo *Android* e periféricos externos.

O fabricante disponibiliza uma biblioteca para ser instalada no ECLIPSE, habilitando o desenvolvedor a criar códigos em Java para a realização de funções I/O com o *smartphone*. A lista abaixo apresenta algumas características relevantes para o projeto:

- Baseada no microcontrolador PIC24F3256 com um *firmware* embarcado responsável por realizar a comunicação entre os dispositivos, sendo possível ser atualizado.
- Disponibiliza 46 pinos de I/O para propósitos gerais.
- 16 pinos de entradas analógicas.
- 9 pinos de saídas PWM.
- Pode ser conectado ao *smartphone* através de um cabo USB ou um receptor *Bluetooth*.

4.3 Plataforma Robótica

A base motora da cadeira de rodas elétrica, *Jazzy Select Elite* da Pride Mobility, foi utilizada como plataforma robótica neste projeto. Foi removido o banco, o suporte para os pés do usuário e a eletrônica de controle embarcada, utilizando apenas a sua base e seus motores.

A plataforma contém duas baterias de 12V que foram utilizadas isoladamente pelo circuito, uma alimentando a parte de potência, e a outra, alimentando a eletrônica de pequenos sinais (IOIO-OTG, condicionador de Sinais). A configuração da plataforma robótica é do tipo diferencial, com

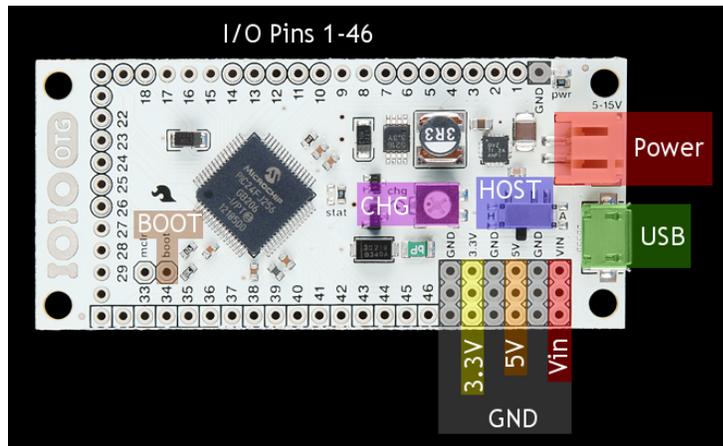


Figura 4.2: Placa IOIO-OTG. (BEN-TSVI, 2013)

dois motores de corrente contínua frontais e duas rodas bobas na parte traseira. A plataforma *Jazzy* no modelo comercial é mostrado na Figura 4.3.



Figura 4.3: Vista superior da plataforma robótica.

4.4 Diagrama do Sistema Robótico

A Figura 4.4 mostra a arquitetura do sistema robótico. O sinal de comando é enviado via rádio para o receptor conectado ao IOIO-OTG, que por sua vez, envia o sinal para o dispositivo móvel que interpreta e envia um sinal PWM proporcional ao sinal recebido para o circuito conversor, e assim, ser aplicado ao amplificador de potência *Sabertooth* (Explicado na sessão 5.2) para acionar os motores. É possível programar no aplicativo desenvolvido rotinas de comandos para serem realizados pelo robô sem precisar do comando externo via rádio.

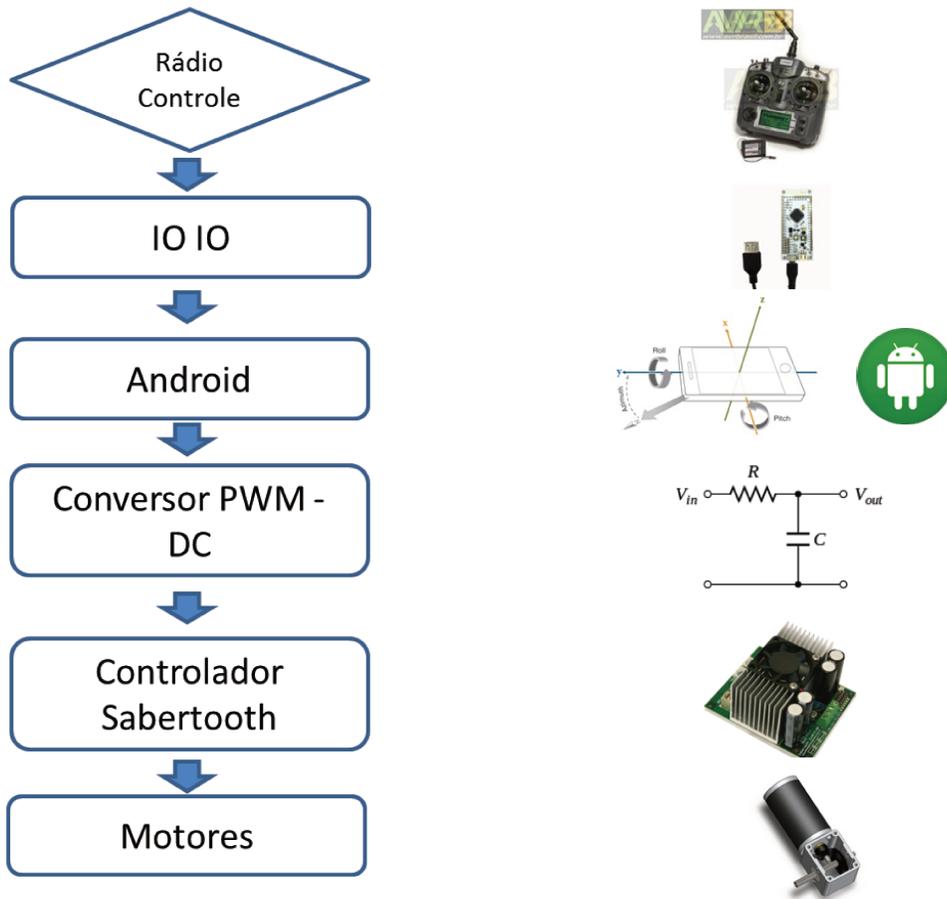


Figura 4.4: Diagrama do Sistema Robótico.

5 Sistemas de Controle de Movimento

Para que a aceleração linear do robô móvel tenha o perfil senoidal desejado, é necessário a aplicação correta do sinal de excitação nos atuadores do robô. As próximas seções explicam a realização desta tarefa.

5.1 *Driver de acionamento Sabertooth*

A placa IOIO-OTG faz a comunicação entre o dispositivo móvel e os periféricos, porém, para o acionamento de atuadores, é necessário uma etapa de amplificação de potência e condicionamento do sinal. Para isso, foi utilizado o *driver Sabertooth 2x60* da *Dimension Engineering* vista na Figura 5.1, responsável por aplicar a potência necessária aos motores do robô a partir de um sinal de baixa potência. A placa é capaz de controlar até dois motores de corrente contínua.



Figura 5.1: Driver de motor Sabertooth.

O *driver* pode operar em vários modos. O modo de trabalho utilizado foi o analógico/independente, podendo controlar cada motor separadamente a partir de dois sinais de entradas analógicas.. O sinal de entrada controla a direção e a velocidade do motor. A faixa de tensão varia de 0V a 5V. Uma tensão de 2.5V de entrada corresponde a velocidade nula, acima de 2.5V o motor rotaciona

para um sentido, e abaixo de 2.5V, inverte a rotação. Quanto maior ou menor for a tensão de entrada em relação a 2.5V(velocidade nula), maior é a velocidade angular do motor.

5.2 Condicionamento e Geração de Sinal

A placa IOIO-OTG é responsável por enviar os sinais de tensão para *driver Sabertooth*, de forma a controlar cada motor individualmente. O sinal de tensão de saída da placa IOIO-OTG, é aplicado em forma de PWM(*Pulse Width Modulation*), onde a tensão média resultante depende da largura de pulso do sinal com período constante. A Equação 5.1 que define a tensão média de um sinal PWM.

$$V_m = V \frac{T_{on}}{T_{on} + T_{off}} \quad (5.1)$$

Onde o seu ciclo ativo em T_{on} , depende de uma função senoidal programada no dispositivo móvel, com objetivo de aplicar a técnica de tensão senoidal modulada desejada no projeto. A tensão de saída do IOIO-OTG é descrita na Equação 5.2 abaixo.

$$V_m = \text{sen}(\omega t) \frac{V}{T_{on} + T_{off}} \quad (5.2)$$

Considerando-se que foi optado por usar o *Sabertooth* em modo analógico, é necessário a conversão do sinal pulsado do IOIO-OTG para um sinal analógico. Para isso, foi desenvolvido um circuito eletrônico que desempenha essa tarefa, fazendo o condicionamento do sinal do IOIO-OTG para o *Sabertooth*.

O circuito eletrônico é mostrado na Figura 5.2.

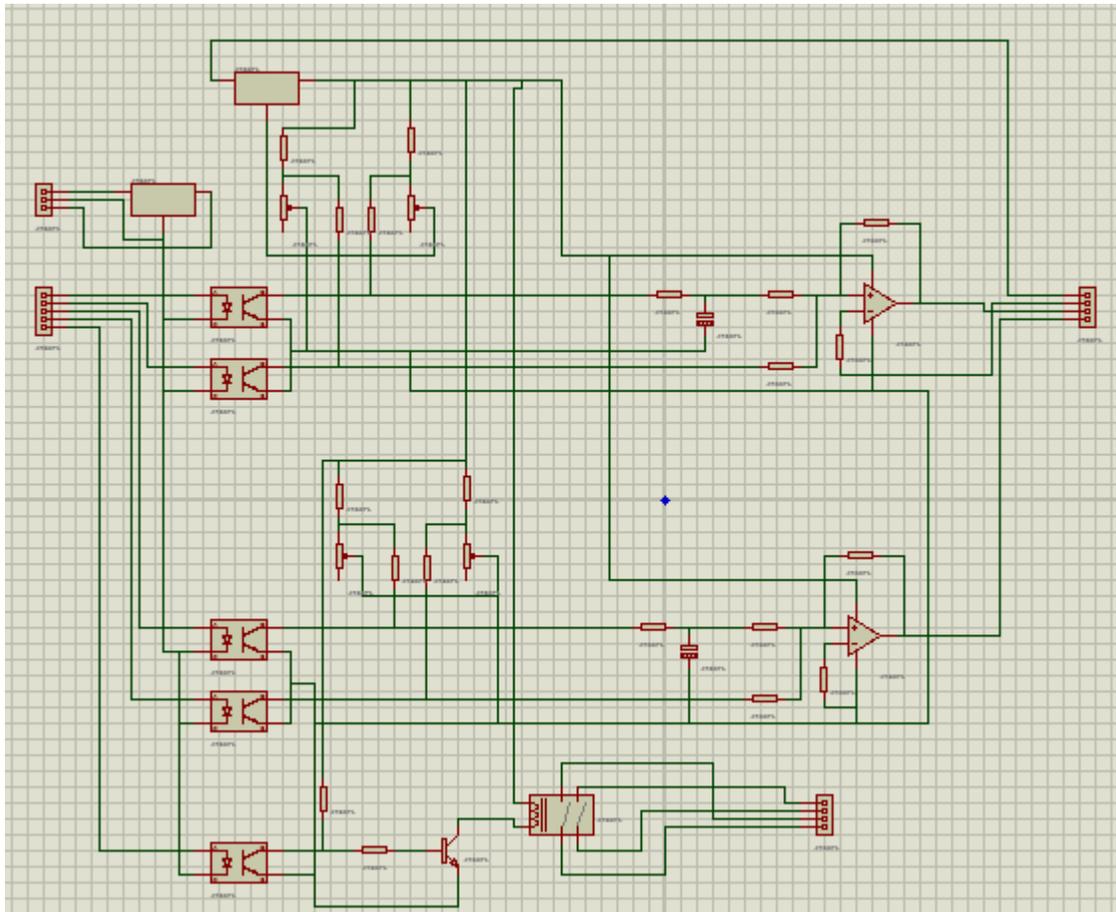


Figura 5.2: Circuito de condicionamento de sinal.

6 Resultados Experimentais

Nesta seção serão mostrados os testes experimentais realizados com o robô, aplicando a técnica de movimento modulado proposta para melhorar a correção do desvio de velocidade e melhorar a tarefa de odometria.

Para a coleta de dados, experimentos foram realizados em um ambiente interno de superfície considerada plana. Um algoritmo que mede as acelerações e estima as orientações do robô via fusão sensorial foi implementado em um aplicativo instalado em um *smartphone* Samsung Galaxy S3 Mini embarcado em um robô terrestre. A plataforma robótica com o celular e seus periféricos embarcados é visualizada na Figura 6.1.



Figura 6.1: Plataforma Robótica

O celular mede as acelerações e estima as orientações via fusão sensorial (filtro complementar) em tempo real. Periodicamente, os dados são gravados na memória do dispositivo para serem posteriormente processados *offline*. Foram realizados duas trajetórias, a primeira, um trajeto linear com curvas, e a segunda, um trajeto circular. Os dados inerciais gravados no dispositivo são lidos e processados no *MATLAB* para reproduzir as trajetórias realizadas. Para validação das trajetórias, foi utilizado o Kinovea, um software de análise biomecânica em vídeos. Os trajetos do robô foram filmados com uma câmera no teto do ambiente para posteriormente ter sua trajetória rastreada visualmente no Kinovea, obtendo-se uma referência do trajeto realizado pelo robô. A escala do ambiente também pode ser calibrada no próprio software.

6.1 Trajetória Linear com curvas

A trajetória realizada neste teste mostrado na Figura 6.2 é composta por 4 retas e 3 curvas.

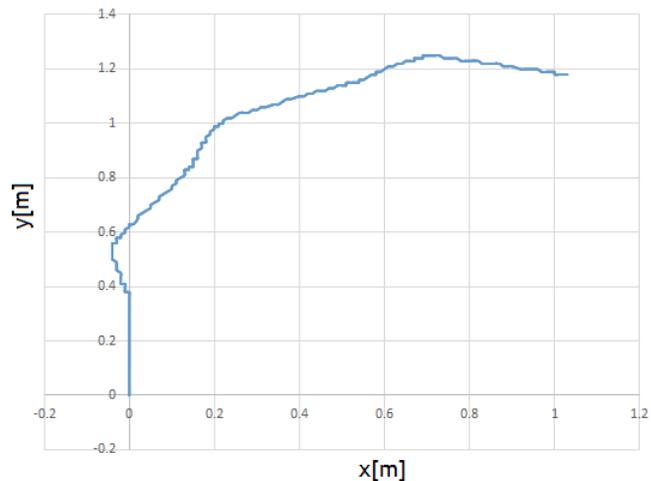


Figura 6.2: Trajetória real via Kinovea

Os gráficos de aceleração linear do robô em relação aos eixos x e y podem ser vistas nas Figuras 6.3 e 6.4 respectivamente, e a orientação, na Figura 6.5.

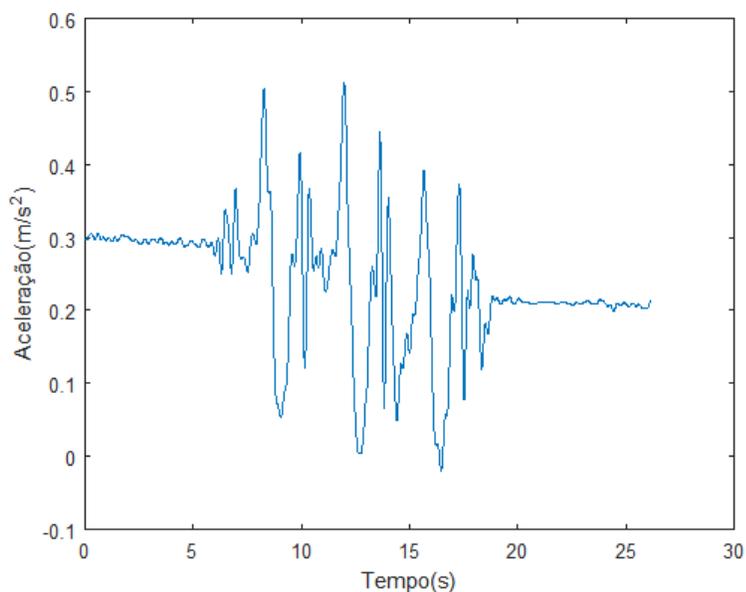


Figura 6.3: Aceleração em x

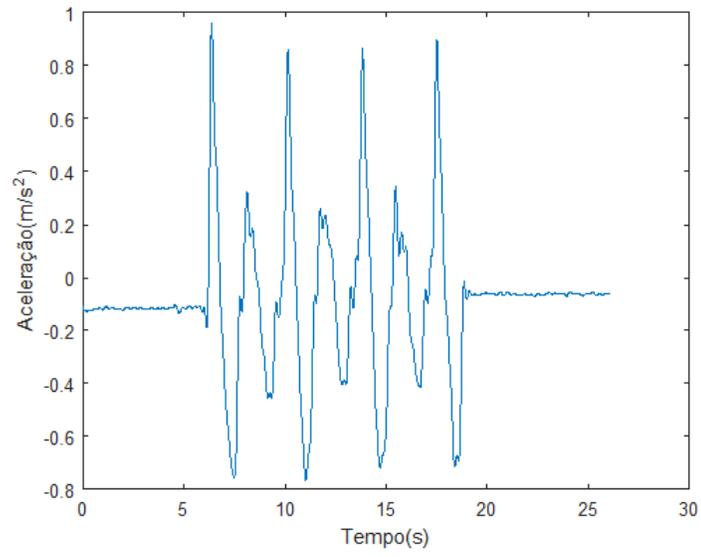


Figura 6.4: Aceleração em y

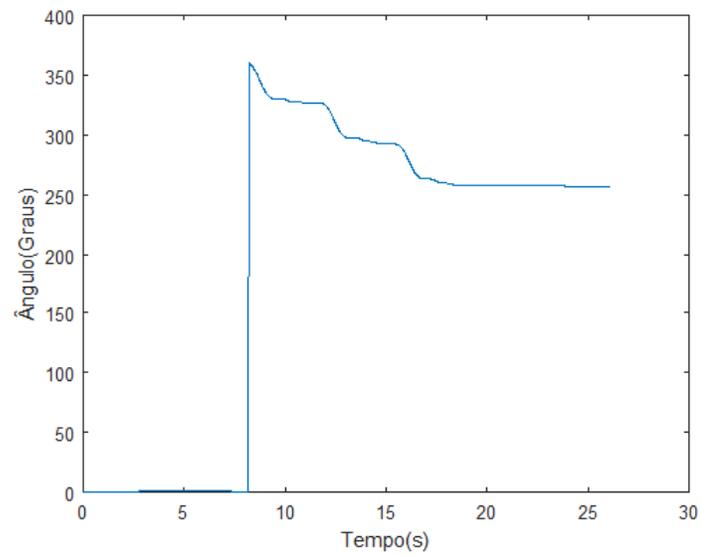


Figura 6.5: Orientação em torno de z

Primeiramente, considera-se o caso em que a estimação de velocidade e posição para cada eixo são obtidas através de integrais simples diretamente aplicadas nos sinais de aceleração. As velocidades em x e y , vistas nas Figuras 6.6 e 6.7, respectivamente, notavelmente sofrem de desvio ao longo do tempo devido a componente de corrente contínua presente nos sinais medidos pelo acelerômetro que acumulam erros no processo de integração. Essa situação causa em uma distorção da trajetória estimada no processo, visualizada na Figura 6.8.

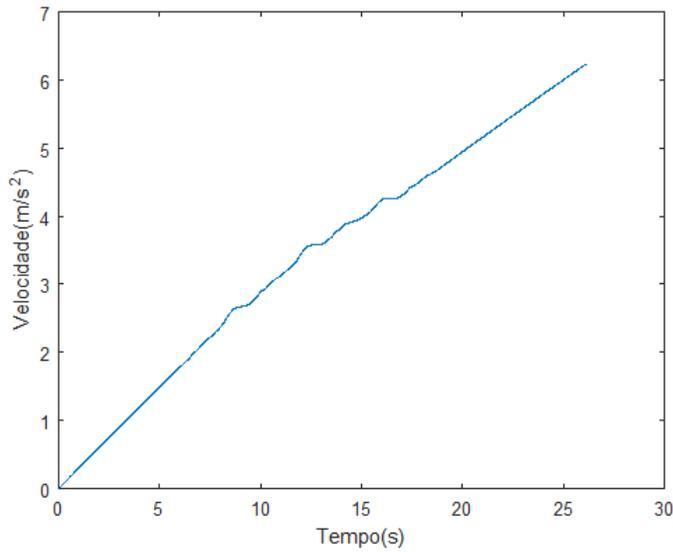


Figura 6.6: Velocidade em x com desvio

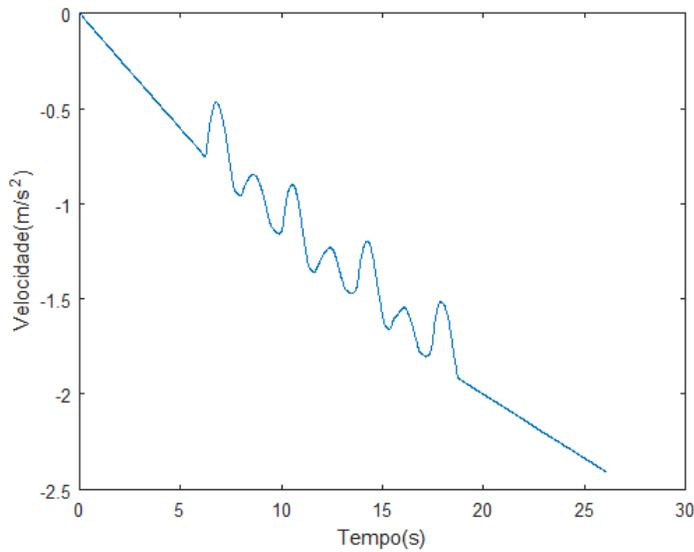


Figura 6.7: Velocidade em y com desvio

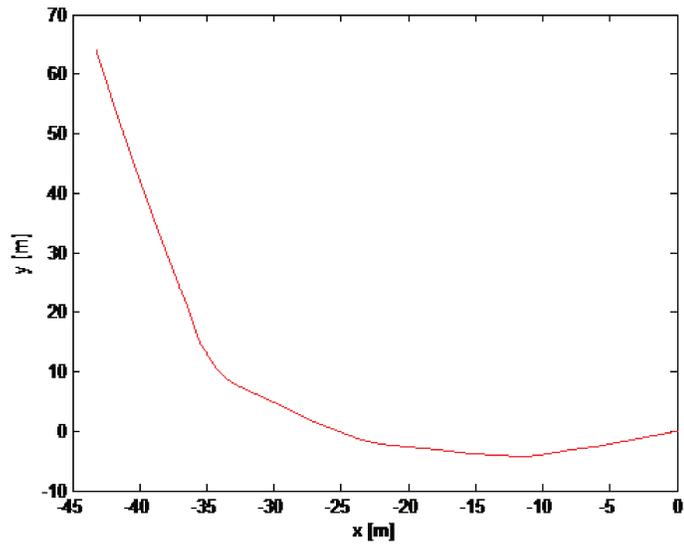


Figura 6.8: Trajetória distorcida

Porém, ao utilizar da vantagem do movimento modulado, que por sua vez, gera uma aceleração com perfil senoidal nos eixos x e y , pode-se aplicar o algoritmo descrito na seção 3.3 para melhorar a estimação de velocidade e deslocamento. Integrando as acelerações dentro dos intervalos identificados, é possível corrigir as velocidades distorcidas dentro destes intervalos, como mostrado na Figura 6.9, para a velocidade em x , e na Figura 6.10, para velocidade em y .

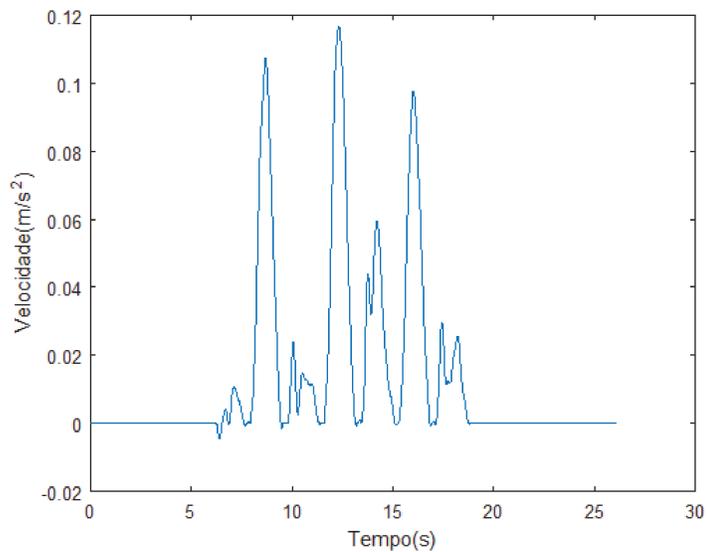


Figura 6.9: Velocidade corrigida em relação ao eixo x

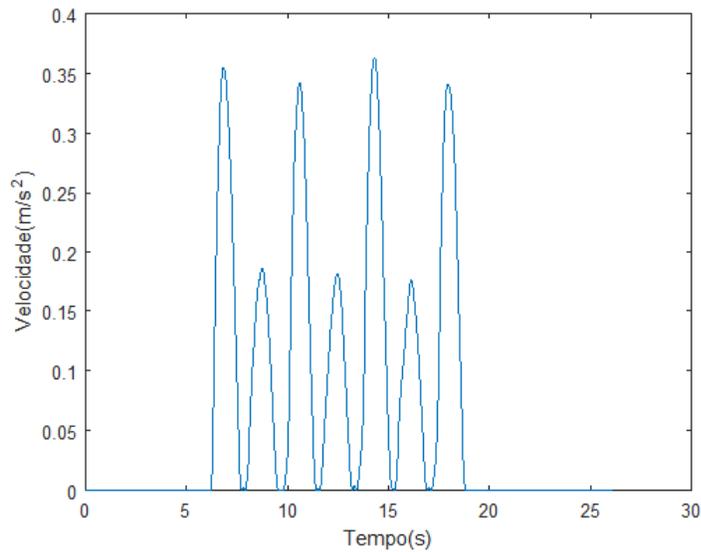


Figura 6.10: Velocidade corrigida em relação ao eixo y

Assim, aplicando novamente a integral nas velocidades em x e y corrigidas, pode ser observado na Figura 6.11, a trajetória estimada pelo robô em azul e a trajetória real em laranja sendo comparadas.

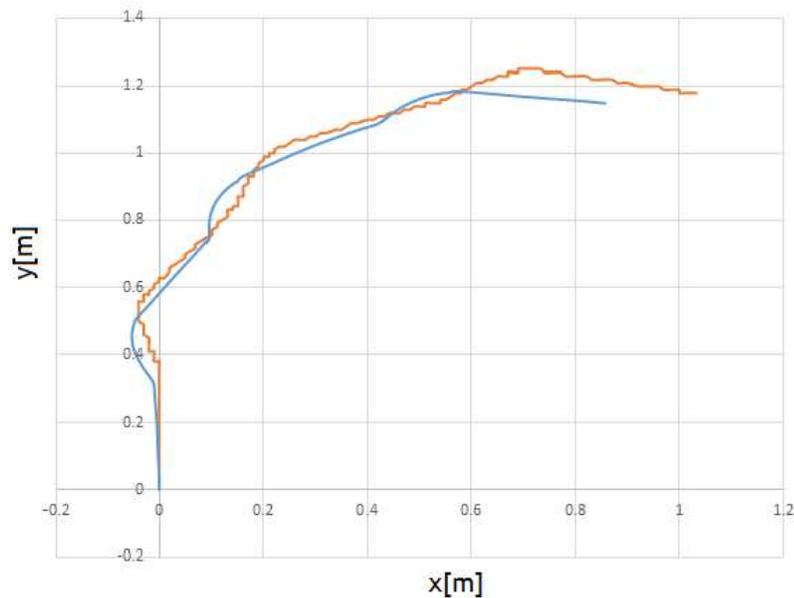


Figura 6.11: Trajetória real(via Kinovea) e estimada(movimento modulado)

É possível observar que a trajetória estimada é qualitativamente fiel a trajetória real, pois a suas formas são semelhantes com a diferenças apenas em escala e rotação. Os erros dos pontos

finais em x e y em percentuais é de 17.47% e 2.62%, respectivamente, melhorando consideravelmente a estimação de trajetória em relação ao método de integração dupla direta mostrado anteriormente.

6.2 Trajetória Circular

A trajetória circular foi realizada pelo robô, fazendo aproximadamente duas voltas, que pode ser visualizada na Figura abaixo.

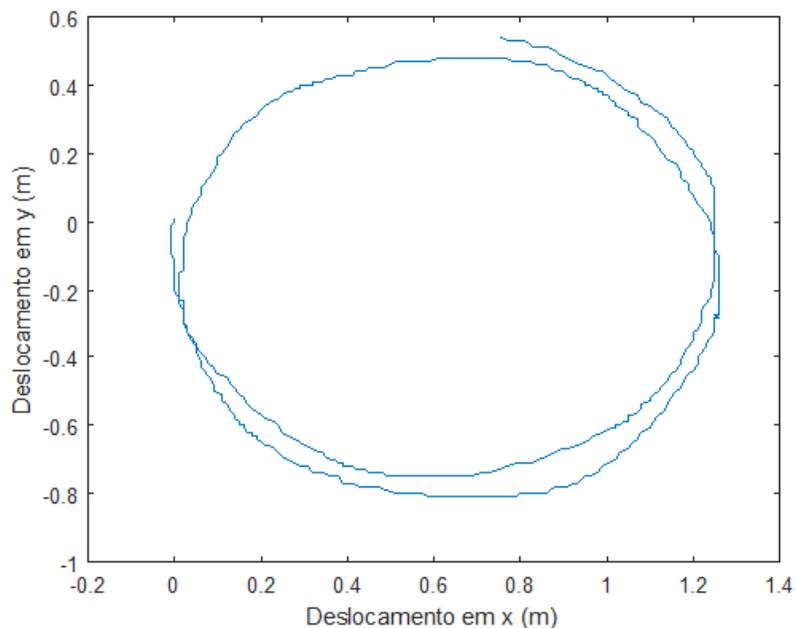


Figura 6.12: Trajetória real via Kinovea

As acelerações no eixo x,y e a orientação em torno do eixo z são mostrados nas Figura 6.13, 6.14 e 6.15 respectivamente.

Como no teste anterior, considera-se o caso em que a estimação de velocidade e posição para cada eixo são obtidas através de integrais simples diretamente aplicadas nos sinais de aceleração. As velocidades em x e y , vistas nas Figuras 6.16 e 6.17, respectivamente, notavelmente sofrem de desvio ao longo do tempo devido a componente de corrente contínua presente nos sinais medidos

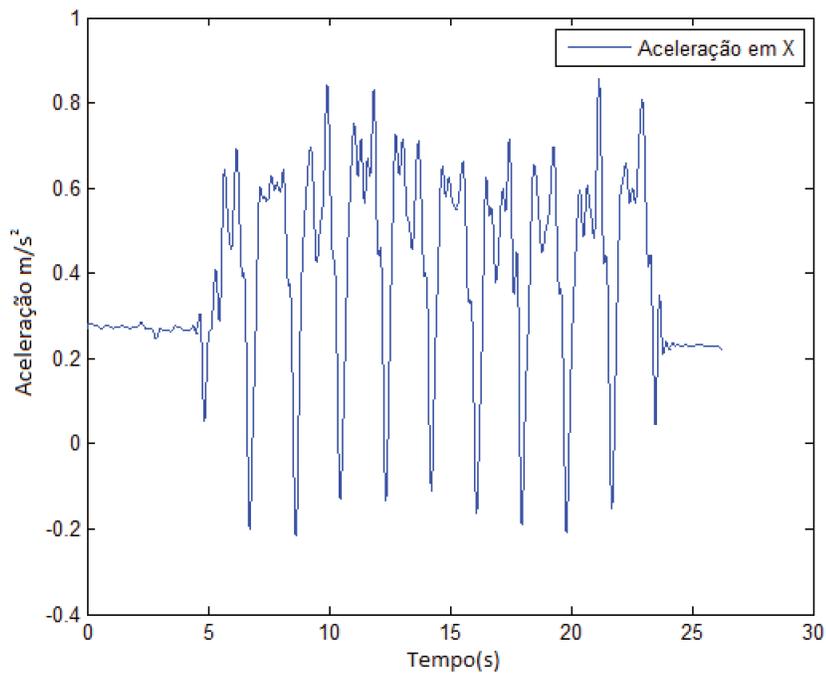


Figura 6.13: Aceleração em x .

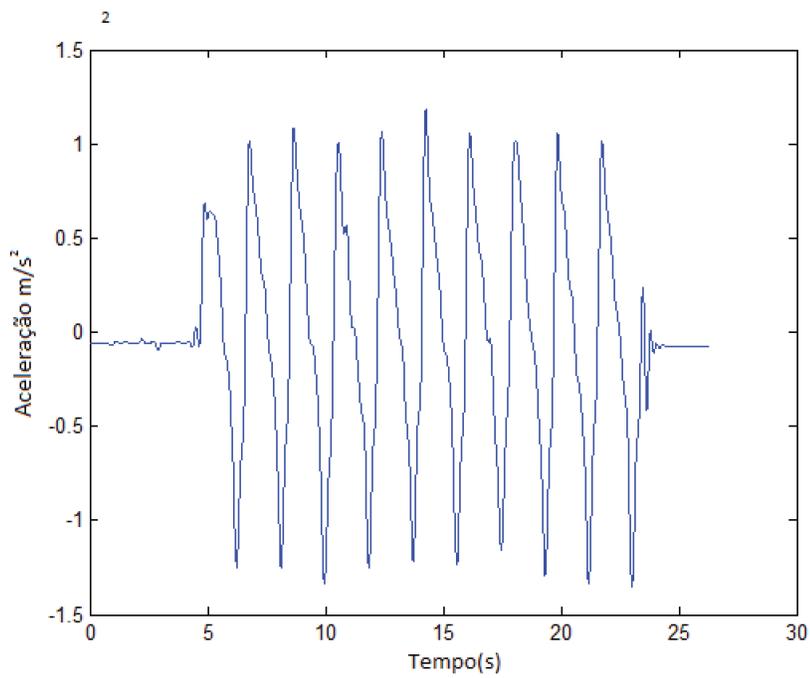


Figura 6.14: Aceleração em y .

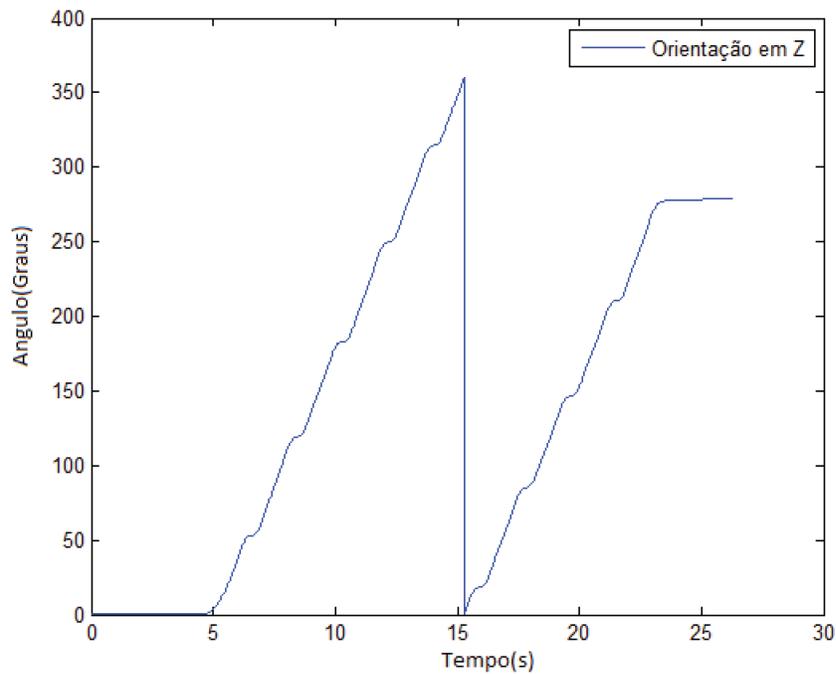


Figura 6.15: Orientação com relação ao eixo z.

pelos acelerômetros que acumulam erros no processo de integração. Essa situação causa em uma distorção da trajetória estimada no processo, visualizada na Figura 6.18.

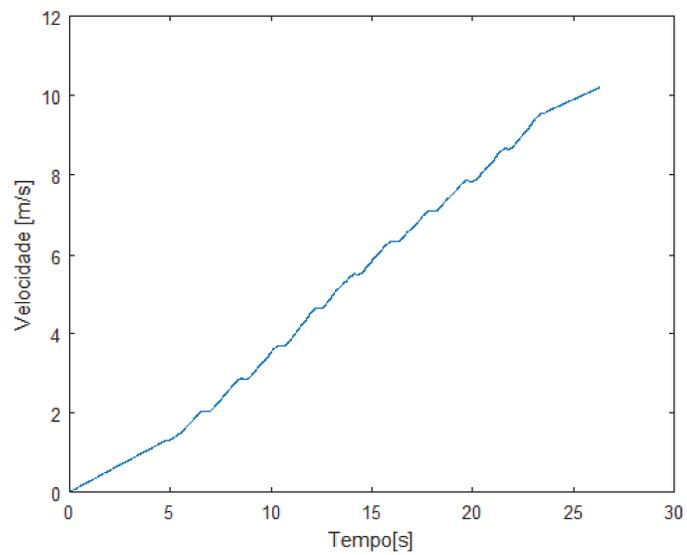


Figura 6.16: Aceleração em relação ao eixo x com intervalos de integração.

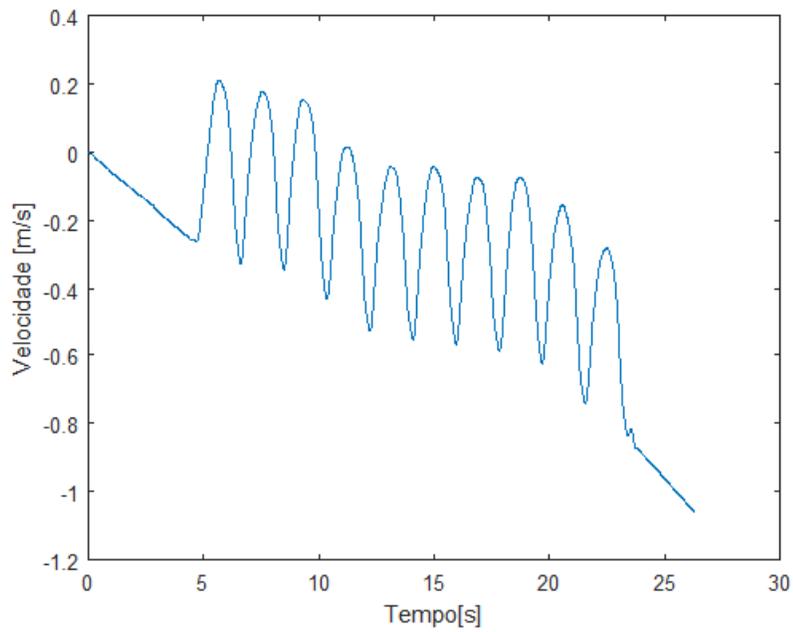


Figura 6.17: Aceleração em relação ao eixo y com intervalos de integração.

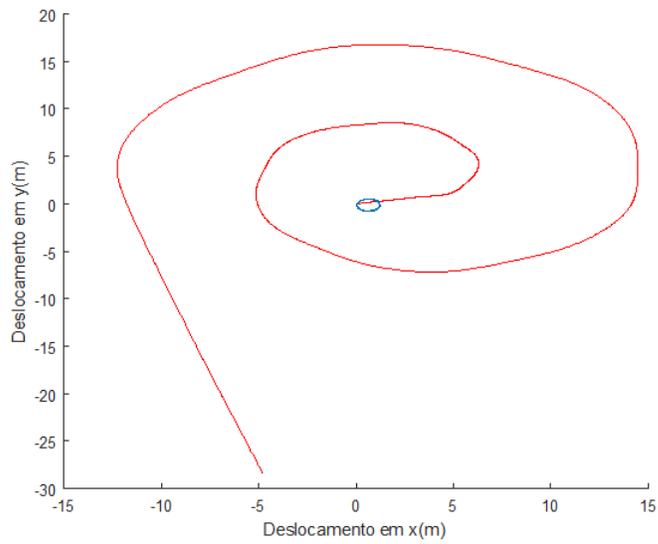


Figura 6.18: Comparação da trajetória distorcida com a trajetória real

Assim, pode-se novamente utilizar a vantagem do movimento modulado descrito na seção 3.3 para melhorar a estimação de velocidade e deslocamento. Integrando as acelerações dentro dos intervalos identificados, é possível corrigir as velocidades distorcidas dentro destes intervalos, como mostrado na Figura 6.19, para a velocidade em x , e na Figura 6.20, para velocidade em y .

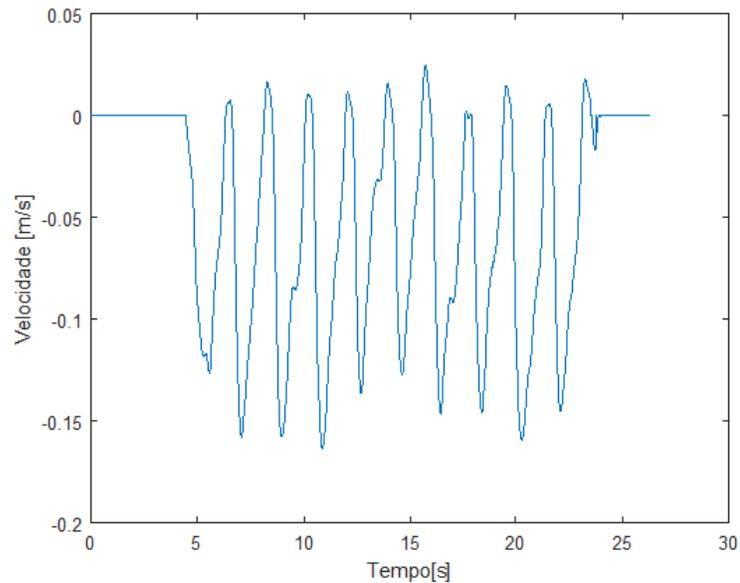


Figura 6.19: Velocidade corrigida relação ao eixo y

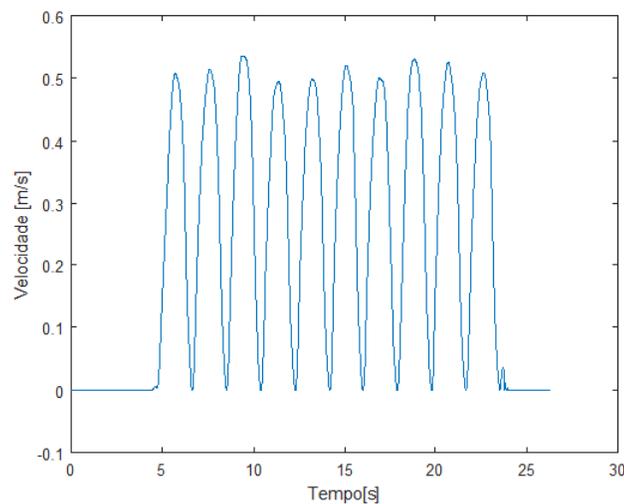


Figura 6.20: Velocidade corrigida relação ao eixo x

Assim, aplica-se novamente a integral nas velocidades em x e y corrigidas pode ser observado na Figura 6.21, a trajetória estimada pelo robô em vermelho e a trajetória real em azul sendo

comparadas.

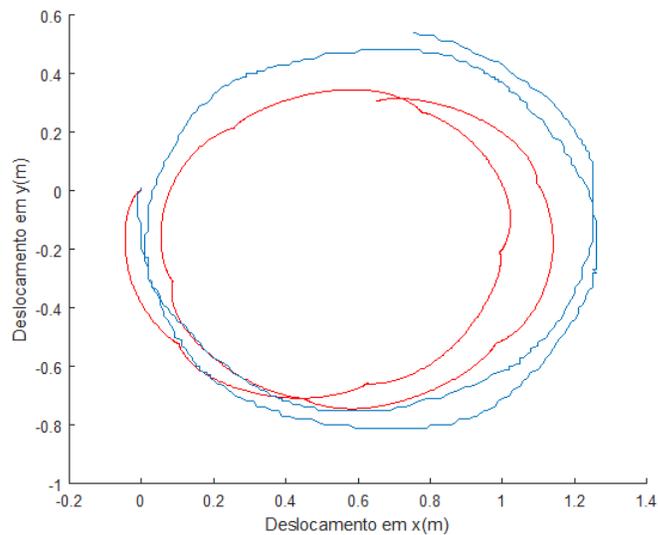


Figura 6.21: Comparação da trajetória estimada com a trajetória real

A trajetória estimada nesse caso tem uma qualidade menos fiel a trajetória real se comparado ao teste com a trajetória linear com curvas. Os erros dos pontos finais na posição x e y em percentuais são aproximadamente de 14,66% e 30,92%, respectivamente. Porém, a melhoria em relação a trajetória via integração das velocidades distorcidas, mostrado na Figura 6.18, é satisfatória.

7 Conclusão

A comunicação entre o dispositivo *Android (smartphone)* e a eletrônica de potência do robô foi bem sucedida ao utilizar o *IOIO-OTG*, que simplifica a comunicação entre os dispositivos. A placa desenvolvida para fazer a conversão entre o sinal *PWM* do *IOIO-OTG* em sinal analógico teve sua funcionalidade adequada para a necessidade do projeto, sendo bem sucedida em isolar o circuito de pequenos sinais (*IOIO-OTG, smartphone*) da placa de potência (*Sabertooth*) e os atuadores do robô. A implementação da fusão sensorial no dispositivo móvel aplicando o filtro complementar para estimação de orientação, teve um resultado satisfatório, usando o melhor de cada técnica de estimar a orientação, no caso, via acelerômetro/magnetômetro e a orientação via giroscópio, dando relevância para o giroscópio a curto prazo, e para o acelerômetro a longo prazo em relação a estimativa de orientação. A estratégia de movimento modulado melhorou significativamente a estimação de deslocamento por facilitar a correção do desvio do sinal de velocidade devido ao nível *CC (offset)* de aceleração, se comparado a integração dupla direta dos dados de aceleração. Experimentos com trajetórias lineares e com poucas curvas, observou-se um erro de trajetória estimada menor se comparado a trajetórias predominantemente circulares. Grande parte dos erros de trajetória presentes são devido a fatores como: amostragem, integração numérica, resolução dos sensores e sua baixa sensibilidade para pequenas acelerações.

Para dar continuidade ao trabalho, seria válido aplicar uma tecnologia de localização absoluta (como sinalizadores), a fim de minimizar os erros de deslocamento acumulados. Além disso, seria interessante implementar para projetos futuros um trabalho em conjunto de navegação inercial com odometria visual e odometria via *encoder*, utilizando ainda a plataforma *Android* para aprimorar a tarefa de estimação de trajetória, e ainda, utilizar visão computacional ou *laser* para diminuir as incertezas de posição e realizar um processo localização do robô utilizando Filtro de Kalman, aplicando a odometria inercial na fase de predição e a visão/*laser* na fase de atualização.

Referências

Ayub, S., Bahraminisaab, A., and & Honary, B. (2012). A sensor fusion method for smart phone. In *The 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting (PGNet2012)*. Liverpool: Liverpool John Moores University., volume 13.

Barshan, B. and Durrant-Whyte, H. F. (1995). Inertial navigation systems for mobile robots. *IEEE Transactions on Robotics and Automation*, 11(3):328–342.

Ben-Tsvi, Y. (2013). Getting to know the ioio otg board.

Bong-Su Cho, Woo-sung Moon, W.-J. S. K.-R. B. (2011). A dead reckoning localization system for mobile robots using inertial sensors wheel revolution encoding. *Journal of Mechanical Science and Technology* 25.

Cho, B.-S., Moon, W.-s., Seo, W.-J., Baek, ., and K.-R (2011). A dead reckoning localization system for mobile robots using inertial sensors. *Journal of Mechanical Science and Technology* 25, 25:2907–2917.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University.

Fourati, H. and Manamanni, N. (2013). Position estimation approach by complementary filter-aided imu. *12th biannual European Control Conference*, 1:4028–4213.

Goddard, R. H. (1914). Rocket apparatus. U.S. Patent 1102653.

Jarvis, R. (2010). Mobile robot localisation and terrain-aware path guidance for teleoperation in

virtual and real space. *International Journal on Advances in Intelligent Systems*, 3.

Lawitzki, P. (2014). Android sensor fusion tutorial. Disponível na Internet via www. URL: <http://www.codeproject.com/Articles/729759/Android-Sensor-Fusion-Tutorial>.

Lee, T., Shirr, J., and Cho, D. D. (2009). Position estimation for mobile robot using in-plane 3-axis imu and active beacon. *IEEE International Symposium on Industrial Electronics (ISIE 2009)*, 1:1956–1961.

Li, M. and Mourikis, A. I. (2013). 3-d motion estimation and online temporal calibration for camera-imu systems. *Robotics and Automation (ICRA)*.

Michel (2014). Vodafone's branded samsung galaxy s3 mini krijgt kleine update.

Nise, N. (2012). *Engenharia de Sistemas de Controle*. Wiley.

Shala, U. and Rodriguez, A. (2011). Indoor positioning using sensor-fusion in android devices. Master's thesis, School of Health and Society - Department Computer Science Kristianstad University.

Siegwart, R., Nourbakhsh, I. R., and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. Massachusetts Institute of Technology, 2 edition.

Smith, S. W. (1997). *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical.

Tesla, N. (1898). Method of and apparatus for controlling mechanism of moving vessels or vehicles. US Patent 613,809.

Vargas, J. A. D. (2012). Localização e navegação de robô autônomo através de odometria e visão estereoscópica. Master's thesis, Universidade Estadual de Campinas.

Walter T Higgins, J. (1975). A comparison of complementary and kalman filtering. *IEEE Aerospace and Electronic Systems Magazine*, Vol.AES-11.

Wooden, D., Malchano, M., Blankespoor, K., Howard, A., and Raibert, A. A. R. M. (2010). Autonomous navigation for bigdog. *Robotics and Automation (ICRA)*, 1:4736 – 4741.

Woodman, O. J. (2007). An introduction to inertial navigation. Technical report, Cambridge: University of Cambridge.

ANEXO A – Artigos Publicados

1) de Mello Silva, P.R.; Quiroz, C.H.C.; Kurka, P.R.G. **Localization of a Robot using Android Platform as Sensors** - 17th International Symposium on Dynamic Problems of Mechanics NATAL - RIO GRANDE DO NORTE - BRAZIL FEBRUARY, 25, 2015. DINAME 2015