

**INSTITUTO FEDERAL DO ESPÍRITO SANTO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E  
AUTOMAÇÃO**

**VINÍCIUS DA ROCHA MOTTA**

**AMBIENTE DE AVALIAÇÃO DE CONTROLADORES DE TRAJETÓRIA E  
ALGORITMOS DE FUSÃO SENSORIAL PARA UMA CADEIRA DE RODAS  
ROBÓTICA**

Serra  
2017

VINÍCIUS DA ROCHA MOTTA

**AMBIENTE DE AVALIAÇÃO DE CONTROLADORES DE TRAJETÓRIA E  
ALGORITMOS DE FUSÃO SENSORIAL PARA UMA CADEIRA DE RODAS  
ROBÓTICA**

Dissertação de Mestrado apresentada à  
Coordenadoria do Curso de Mestrado Profissional  
em Engenharia de Controle e Automação do  
Instituto Federal do Espírito Santo como requisito  
parcial para a obtenção do título de Mestre em  
Engenharia de Controle e Automação.

Orientadores: Prof. Dr. Marco Antonio de Souza  
Leite Cuadros e Prof. Dr. Gustavo Maia de  
Almeida.

Serra

2017

### Dados Internacionais de Catalogação na Publicação (CIP)

---

M921a Motta, Vinícius da Rocha  
2017 Ambiente de avaliação de controladores de trajetória e algoritmos de fusão sensorial para uma cadeira de rodas robótica / Vinícius da Rocha Motta. - 2017.  
109 f.; il.; 30 cm

Orientadores: Prof. Dr. Marco Antonio de Souza Leite Cuadros e Prof. Dr. Gustavo Maia de Almeida

Dissertação (mestrado) - Instituto Federal do Espírito Santo, Programa de Pós-graduação em Engenharia de Controle de Automação, 2017.

1. Robótica. 2. Robôs móveis. 3. Sistemas de controle inteligente. 4. Cadeiras de roda. I. Cuadros, Marco Antonio de Souza Leite. II. Almeida, Gustavo Maia de. III. Instituto Federal do Espírito Santo. IV. Título.

CDD 629.892

---

Bibliotecária Rogeria Gomes Belchior - CRB6/ES 417



**MINISTÉRIO DA EDUCAÇÃO**

INSTITUTO FEDERAL DO ESPÍRITO SANTO

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

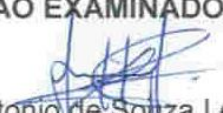
**VINICIUS DA ROCHA MOTTA**

**AMBIENTE PARA AVALIAÇÃO DE CONTROLADORES DE TRAJETÓRIA E  
ALGORÍTMOS DE FUSÃO SENSORIAL PARA UMA CADEIRA DE RODAS ROBÓTICA**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Controle e Automação do Instituto Federal do Espírito Santo, como requisito parcial para obtenção de título de Mestre em Engenharia de Controle e Automação.

Aprovado em 27 de novembro de 2017

**COMISSÃO EXAMINADORA**

  
Prof. Dr. Marco Antonio de Souza Leite Cuadros

Instituto Federal do Espírito Santo

Orientador

  
Prof. Dr. Gustavo Maia de Almeida


Instituto Federal do Espírito Santo

Orientador

  
Prof. Dr. Saul Munareto

Instituto Federal do Espírito Santo

Membro Interno

  
Prof. Dr. Teodiano Freire Bastos Filho  
Universidade Federal do Espírito Santo

Membro externo

## **AGRADECIMENTOS**

Agradeço, primeiramente, a Deus por ter me dado saúde e força para superar as dificuldades ao longo da minha vida. Aos professores orientadores Marco Antonio e Gustavo Maia, por me apresentarem à área de robótica móvel, e também, pelos diversos ensinamentos e orientações, sendo esses, de grandiosa importância para a conclusão deste trabalho e que me ajudam, a cada dia, em melhorar como ser humano. A todos os membros do grupo de estudos de Automação Industrial - GAI, que me proporcionaram crescimento intelectual e humano. Aos amigos estudantes do programa, pois sem os quais não chegaria a esta etapa. Aos meus pais, pelos ensinamentos basilares que forjaram meu caráter, me tornando a pessoa que sou, e que me fortalecem na busca dos objetivos que desejo alcançar. A minha esposa, pela paciência, compreensão e companheirismo nas horas mais difíceis.

Agradeço também a todos os professores por me proporcionarem o conhecimento, não apenas racional, mas a manifestação do caráter e afetividade da educação no processo de formação profissional, pelo tanto que se dedicaram a mim, não somente por terem me ensinado, mas por terem me feito aprender.

E a todos os demais amigos, colegas e familiares que, direta ou indiretamente, contribuíram para a realização deste trabalho.

*“Escolhe um trabalho de que gostes, e não terás que trabalhar nem um dia na tua vida”.*

*(Confúcio)*

## RESUMO

Um dos maiores problemas da navegação de robôs móveis em ambientes internos é a obtenção correta da sua pose em seu espaço de trabalho. A utilização de um método odométrico (*Dead Reckoning*) baseada na leitura de *encoders* é uma técnica frequentemente utilizada, entretanto, ela acumula erros durante a navegação. Por outro lado, o GPS pode determinar a posição através da triangulação de sinais de satélite, entretanto, a recepção deste tipo de sinal se torna inviável em ambientes fechados, além disso, este sensor informa, usualmente, apenas a posição e não a orientação. Embora existam outros métodos, nenhum por si só, é suficiente para garantir uma navegação correta. Nesta Dissertação de Mestrado, objetiva-se encontrar o melhor controlador de trajetória e implementar algoritmos de fusão sensorial numa cadeira de rodas elétrica com o intuito de melhorar a informação da estimativa da sua pose. Essa informação é obtida através do método odométrico *Dead Reckoning* e do uso de sensores. Um gerador de trajetória foi usado para ser referência do controlador de trajetória. O controlador de trajetória tem a função de minimizar o erro da pose em relação a referência. Os três controladores de trajetória utilizados para a comparação foram escolhidos com base na literatura, pois tem melhor desempenho para este tipo de aplicação. Foram construídas e comparadas duas fusões sensoriais. Na primeira fusão foi utilizado o *Dead Reckoning* juntamente com os dados do giroscópio para corrigir a orientação e realimentar a odometria de modo que ela seja mais precisa. Na segunda fusão foi utilizado o *Dead Reckoning* juntamente com os dados do giroscópio e o magnetômetro. A fusão foi realizada com o filtro de Kalman estendido (*Extended Kalman Filter* - EKF). Os resultados foram validados através de informações advindas de uma câmera posicionada no teto (calibrada e testada) com a informação considerada como pose real. Assim, foram utilizadas técnicas como o *Dead Reckoning*, sensores inerciais e magnéticos, visão computacional e o EKF. Os resultados obtidos foram significativos em relação ao esperado, uma vez que se melhorou a estimativa da pose do robô. Portanto, as implicações dos estudos realizados podem servir como base prática para uma aplicação em uma cadeira de rodas elétrica.

Palavras-chave: Fusão sensorial. Filtro de Kalman estendido. Navegação de robôs móveis. Cadeira de rodas robótica.

## ABSTRACT

One of the biggest problems for indoor mobile robot navigation is a correct acquisition of its pose in its workspace. The use of Dead Reckoning method, based on pulse count through encoders is a technique frequently used, however, it is accumulating error during navigation. On the other hand, GPS is used on outdoors because it determines a location through the triangulation of satellite signals, however, this signal reception becomes invalid indoors, moreover, this sensor usually reports the position, but not the orientation. Although there exist other methods, none of them is enough to ensure a correct navigation. In this Master's Dissertation, we aim to find the best of trajectory controller and implement algorithms of sensorial fusion in an electric wheelchair to determine the estimation of its pose. This information is obtained through Dead Reckoning and the use of other sensors. A trajectory generator was used to refer to the trajectory controller. The trajectory controller has the function of minimizing the pose error in relation to the reference. The three path controllers used for the comparison were chosen based on the literature, since it has better performance for this type of application. Two sensory fusion techniques were used and compared. In the first fusion, Dead Reckoning was used along with gyro data to correct the orientation and to feed the odometry, in the order to be more accurate. In the second fusion, Dead Reckoning was used along with data of the gyro and magnetometer. The fusion was performed with the Extended Kalman Filter (EKF). The results were validated through information from a camera positioned on the ceiling (calibrated and tested), with information considered as real pose. Thus, techniques such as Dead Reckoning, inertial and magnetic sensors, computational vision and EKF were used. The results obtained were significant in relation to the expected one, once the robot's pose estimation was improved. Therefore, the implications of this study can serve as a practical basis for an application in an electric wheelchair.

Key-words: Sensorial Fusion. Extended Kalman Filter. Mobile robot navigation. Robotic wheelchair.



## LISTA DE ILUSTRAÇÕES

Figura 1 - Algoritmo do KF. ....	23
Figura 2 - Algoritmo do EKF. ....	27
Figura 3 - Erros do controlador Fuzzy. ....	29
Figura 4 - Função de pertinência da entrada $\gamma$ . ....	30
Figura 5 - Função de pertinência da entrada $D$ . ....	31
Figura 6 - Função de pertinência da entrada $v$ . ....	31
Figura 7 - Função de pertinência da entrada $w$ . ....	31
Figura 8 - Curvas de entrada-saída globais, a) $v$ e b) $w$ . ....	33
Figura 9 - Disposição dos erros de trajetória. ....	34
Figura 10 - Modelo cinemático não holonômico de acessibilidade melhorada. ....	35
Figura 11 - Referência global do robô. ....	39
Figura 12 - Modelo de arquitetura da fusão sensorial do EKF e saída da estimação. ....	43
Figura 13 - Leitura realizada pelo <i>encoder</i> . ....	44
Figura 14 - Exemplo de giroscópio mecânico. ....	45
.Figura 15 - Linhas de campo magnético da Terra. ....	46
Figura 16 - Declinação e inclinação magnética. ....	47
Figura 17 - Representação gráfica do <i>Starter kit 2.0</i> . ....	49
Figura 18 - Representação gráfica da cadeira elétrica motorizada <i>Styles SX20</i> . ....	50
Figura 19 - Representação gráfica dos movimentos aceitos pelo robô móvel não-holonômico. ....	51
Figura 20 - Simulador do Starterkit 2.0 - Escolha do projeto para Robotics. ....	53
Figura 21 - Simulador do <i>Starter kit 2.0</i> - Escolha do projeto para simulação. ....	54
Figura 22 - Simulador do Starterkit 2.0 - Configuração da simulação. ....	54
Figura 23 - Simulador do Starterkit 2.0 - Projeto criado. ....	55
Figura 24 - Simulador do Starterkit 2.0 - Execução do ambiente de simulação. ....	55
Figura 25 - DaNI 2.0 e seus principais componentes. ....	56
Figura 26 - Placa 9632 NI <i>Single-Board RIO</i> <sup>TM</sup> com processador de tempo real, FPGA e I/Os (digital e analógicas). ....	57
Figura 27 - Ambiente de testes. ....	59
Figura 28 - Placa com círculos usada como referência para obter a pose do robô...59	
Figura 29 - <i>Grid</i> construído para facilitação da calibração. ....	61

Figura 30 - Posicionamento da placa de localização posicionada na cadeira de rodas robótica.....	62
Figura 31 - Ambiente de testes utilizado para se obter os resultados.....	63
Figura 32 - Leitura da posição do robô com câmera.....	63
Figura 33 - Diagrama de blocos da cadeira de rodas robótica.....	66
Figura 34 - Cadeira motorizada Styles SX20.....	66
Figura 35 - Módulo <i>driver connect</i> .....	67
Figura 36 - Joystick da cadeira motorizada <i>Styles SX20</i> .....	67
Figura 37 - Instalação do encoder: a) Visão lado externo e b) Visão lado interno. ...	69
Figura 38 - Descrição do <i>joystick</i> .....	70
Figura 39 - Placa eletrônica do módulo do painel <i>connect</i> .....	71
Figura 40 - Placa eletrônica do módulo do <i>drive connect</i> .....	72
Figura 41 - Procedimento de calibração do giroscópio.....	74
Figura 42 - Procedimento de calibração do magnetômetro.....	75
Figura 43 - Gráfico de representação da velocidade angular com diferentes referências.....	78
Figura 44 - Gráfico de representação da velocidade linear com diferentes referências.....	79
Figura 45 - Conectores do dispositivo embarcado NI MYRIO™.....	79
Figura 46 - Conexões realizadas no NI MYRIO™.....	80
Figura 47 - Bloco <i>analog output</i> do Labview™.....	80
Figura 48 - Respostas ao degrau em malha aberta da velocidade angular da cadeira de rodas.....	82
Figura 49 - Respostas ao degrau em malha aberta da velocidade linear da cadeira de rodas.....	82
Figura 50 - Respostas ao degrau em malha aberta da velocidade linear da cadeira de rodas.....	84
Figura 51 - Respostas ao degrau em malha aberta da velocidade angular da cadeira de rodas.....	84
Figura 52 - Origem do modelo cinemático do robô móvel utilizado.....	89
Figura 53 - Gráfico do experimento em simulação da trajetória.....	92
Figura 54 - Gráfico de comparação de erros de controladores em simulação.....	93
Figura 55 - Gráfico do experimento prático da trajetória.....	94
Figura 56 - Gráfico do erro absoluto do experimento prático.....	95
Figura 57 - Esquema completo do controle da cadeira de rodas robótica.....	97
Figura 58 - Esquema de fusão sensorial implementado com odometria e giroscópio.....	98

Figura 59 - Esquema de fusão sensorial implementado com odometria, giroscópio e magnetômetro. ....	98
Figura 60 - Sintonia do Backstepping com a cadeira suspensa. ....	99
Figura 61 - Comparação da pose da câmera, da odometria e das duas fusões de dados.....	100
Figura 62 - Cálculo do Erro Absoluto no tempo para as duas fusões. ....	101
Figura 63 - Comparação dos erros dos <i>thetas</i> da pose da câmera, da odometria e da fusão de dados.....	101

## LISTA DE TABELAS

Tabela 1 - Notações comuns para dimensões de <i>Array</i> .....	26
Tabela 2 - Variáveis <i>Fuzzy</i> .....	30
Tabela 3 - Regras <i>Fuzzy</i> .....	32
Tabela 4 - Regras <i>Fuzzy</i> detalhadas.....	32
Tabela 5 - Dados das plataformas utilizados no <i>Dead Reckoning</i> . ....	39
Tabela 6 - Velocidades máximas aplicadas as plataformas robóticas. ....	51
Tabela 7 - Dados das variáveis utilizados em ambos os programas de simulação. ....	65
Tabela 8 - Características da cadeira motorizada. ....	68
Tabela 9 - Descrição das funções do painel <i>connect</i> . ....	70
Tabela 10 - Legenda de cores dos fios. ....	72
Tabela 11 - Dados de velocidades versus referência da velocidade angular.....	77
Tabela 12 - Dados de velocidades versus referência da velocidade linear.....	78
Tabela 13 - Ganhos dos PID's das velocidades.....	85
Tabela 14 - Valores de IAE de cada controlador na simulação.....	93
Tabela 15 - Valores de IAE de cada controlador nos experimentos práticos. ....	95
Tabela 16 - Ganhos do controlador de trajetória <i>Backstepping</i> .....	100
Tabela 17 - IAE de comparação dos dados obtidos pela odometria e pelas fusões de dados com os dados obtidos pela câmera. ....	102
Tabela 18 - Valores da variância do erro da matriz <i>R</i> . ....	102

## LISTA DE ABREVIATURAS, SIGLAS

CAN - *Controller Area Network*

EKF - *Extended Kalman Filter*

GAIn - Grupo de Estudo de Automação Industrial do Campus Serra do Instituto Federal do Espírito Santo.

GPS - *Global Positioning System*

IMU - *Inertial Measurement Unit*

KF - *Kalman Filter*

MIT - *Massachusetts Institute of Technology*

NI - *National Instruments*

subVI - *Subroutine Virtual Instruments*

UKF - *Unscented Kalman Filter*

VANT - Veículo Aéreo Não Tripulado

VI - *Virtual Instruments*

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	20
1.1	OBJETIVO GERAL.....	17
1.2	METODOLOGIA.....	18
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b> .....	20
2.1	TEORIA DO FILTRO DE KALMAN .....	20
2.1.1	<b>Aplicações do filtro de Kalman</b> .....	21
2.1.2	<b>Filtro de Kalman estendido</b> .....	27
2.2	CONTROLADOR DE SEGUIMENTO DE TRAJETÓRIA .....	27
2.2.1	<b>Controlador Fuzzy</b> .....	29
2.2.2	<b>Controlador <i>Backstepping</i></b> .....	33
2.2.3	<b>Controlador de ganhos fixos</b> .....	35
2.3	<i>DEAD RECKONING</i> .....	38
2.4	ESQUEMA DE FUSÃO DE DADOS COM O FILTRO DE KALMAN .....	41
2.4.1	<b>Encoders</b> .....	43
2.4.2	<b>Giroscópio</b> .....	45
2.4.3	<b>Magnetômetro</b> .....	46
<b>3</b>	<b>MATERIAIS E MÉTODOS</b> .....	48
3.1	DESCRIÇÃO DA PLATAFORMA ROBÓTICA .....	48
3.1.1	<b>Plataformas robóticas usadas</b> .....	49
3.1.2	<b>Ambiente de simulação do Starter KIT 2.0</b> .....	52
3.1.3	<b>Starter KIT 2.0</b> .....	56
3.1.4	<b>Estimativa de localização por processamento de imagens</b> .....	58
3.1.5	<b>Cadeira de rodas Robótica</b> .....	65
3.2	CALIBRAÇÃO DOS SENSORES.....	73
3.2.1	<b>Giroscópio</b> .....	73
3.2.2	<b>Magnetômetro</b> .....	74
3.3	ANÁLISE E COMPARAÇÃO DOS DADOS COM <i>INTEGRAL ABSOLUTE ERROR</i> (IAE).....	76
3.4	INSTRUMENTAÇÃO, MODELAGEM E <i>TUNNING</i> DOS GANHOS DOS CONTROLADORES DE VELOCIDADES UMA DA CADEIRA DE RODAS ROBÓTICA .....	76
3.5	ERRO DE POSIÇÃO.....	85

3.6	FILTRO DE KALMAN ESTENDIDO .....	85
<b>4</b>	<b>IMPLEMENTAÇÃO .....</b>	<b>90</b>
4.1	COMPARAÇÃO DE CONTROLADORES DE TRAJETÓRIA.....	90
<b>4.1.1</b>	<b>Resultados dos controladores de trajetória .....</b>	<b>90</b>
<b>4.1.2</b>	<b>Dados da simulação.....</b>	<b>91</b>
<b>4.1.3</b>	<b>Dados do experimento prático.....</b>	<b>93</b>
4.2	IMPLEMENTAÇÃO DO CONTROLADOR DE TRAJETÓRIA NA CADEIRA DE RODAS ROBÓTICA .....	96
4.3	RESULTADOS OBTIDOS .....	99
<b>5</b>	<b>CONCLUSÕES .....</b>	<b>104</b>
	<b>REFERÊNCIAS.....</b>	<b>104</b>

## 1 INTRODUÇÃO

A navegação é uma atividade essencial para que os robôs móveis realizem sua atividade fim. No entanto, segundo Heinen (2002), existem determinados fatores que causam problemas à navegação, os quais são associados às características típicas deste tipo de plataforma, que são as limitações dos sensores, pois eles invariavelmente não conseguem medir diretamente a informação de interesse. Por exemplo, câmeras de vídeo percebem a cor, o brilho e a saturação da luz, entretanto na navegação de um robô é essencial encontrar respostas para as seguintes questões: “existe uma porta em frente ao robô?”. Por outro lado, a faixa de medida dos sensores (por exemplo: sensores de distância ultrassônicos) é restrita a uma região no entorno do robô. Além disso, o ruído sensorial é um outro fator que interfere nas medidas, pois as informações obtidas pelos sensores estão normalmente contaminadas com ruídos, ou possuem distorções nas medidas, leituras incorretas, etc.

A informação da pose de um robô móvel é de essencial importância para a sua navegação. Para diminuir a incerteza da informação referente a sua pose (posição e orientação), pode-se utilizar diversos métodos. Nesta Dissertação de Mestrado, se propõe a realização de uma fusão de dados de sensores com o método odométrico *Dead Reckoning*, utilizando-se o filtro de Kalman estendido (*Extended Kalman Filter* - EKF). Com isso, deseja-se melhorar os valores de  $\theta$ , ou seja, melhorar a informação da orientação do robô tendo em vista sua grande influência na estimativa de localização realizada pela odometria.

Existem muitas versões do filtro de Kalman, as quais são amplamente estudadas na literatura, tais como filtro de Kalman (*Kalman Filter* - KF) (GREWAL; ANDREWS, 2008), filtro de Kalman Estendido (*Extended Kalman Filter* - EKF) (GREWAL; ANDREWS, 2008) e filtro de Kalman *Unscented* (*Unscented Kalman Filter* - UKF) (JULIER, UHLMANN, DURRANT-WHYTE, 2000). Essas variações do filtro de Kalman se dão devido aos vários tipos de aplicação, mas todos se baseiam na operação no algoritmo fundamental originalmente proposto por Kalman (1960), cuja forma padrão foi desenvolvido para sistemas lineares.



O EKF é a versão estendida do KF, sendo utilizado para sistemas não lineares e invariantes no tempo. O UKF também é utilizado para sistemas não lineares invariantes no tempo, sendo a sua diferença em relação ao EKF o fato de usar a transformada *Unscented* (*Unscented Transform* - UT), a qual utiliza um conjunto de pontos para representar mais precisamente a média e a variância da pose do robô (supondo que se tenha uma distribuição Gaussiana), em lugar de estimar a covariância do erro. Com isso, a ideia é propagar esses pontos através de um sistema não linear, e a covariância média posterior é obtida com precisão, o que dá um resultado melhor quando comparado com a abordagem de EKF (JULIER, UHLMANN, DURRANT-WHYTE, 2000). Assim, o filtro de Kalman possui a característica necessária para a fusão sensorial, ou seja, ele proporciona uma maneira de realizar a integração dos dados dos sensores, mesmo com medidas contaminadas por ruído, e conseguir retirar uma informação melhor através da fusão (PANIAGUA, 2012).

Existe também a imprecisão associada ao movimento do robô, devido a força de inércia sobre ele, somado a complexidade e dinâmica do espaço de trabalho ou ambiente no qual o robô está inserido, que geralmente não é estruturado. Além disso, é difícil a obtenção de um modelo válido para o ambiente completo, de forma a prevenir eventos de maneira precisa, já que para isto o modelo do ambiente deveria ser bem detalhado, e ao mesmo tempo simples e de fácil acesso, pois as ações sobre ele necessitam ser em tempo real. Portanto, diversas pesquisas foram desenvolvidas para tentar resolver estes problemas e tornar a navegação de um robô móvel mais robusta e capaz de atuar de forma satisfatória em ambientes complexos (HEINEN, 2002).

Para efeitos de controle do robô, a informação da pose é de suma importância, uma vez que ela indica a sua localização e a sua orientação em seu espaço de trabalho. A falta ou imprecisão da informação da pose do robô faz com que não consiga navegar em seu entorno, seguir uma trajetória ou seguir para um determinado destino ou mesmo voltar ao ponto inicial. Sempre existe um erro sobre a informação da pose do robô, o qual é associado à leitura de sensores, sendo este erro causado pela não linearidade dos sensores ou por contaminações ocasionadas mediante

ruídos, causando uma grande incerteza na medição da pose do robô (SIEGWART, NOURBAKHS, SCARAMUZZA, 2011).

A pose do robô móvel pode ser estimada de várias formas. Uma forma de estimar a pose do robô é a odometria, ou seja, o método odométrico *Dead Reckoning*, o qual admite que a pose inicial do robô é conhecida. A partir daí a pose pode ser estimada a qualquer instante através da informação relativa ao movimento (MICHEL, ROHRER, HEINIGER, 2014). A vantagem deste método é que o tempo de resposta é pequeno, a desvantagem é que o erro entre a pose real e a estimada se acumula ao longo do tempo resultando em estimação de sua pose muito diferente da pose real (PANIAGUA, 2012).

Para realizar a navegação dos robôs móveis é necessária a utilização de informações adquiridas de sensores. As informações adquiridas de sensores são contaminadas com ruído. Técnicas odométricas, como o *Dead Reckoning*, acumulam erros que aumentam com a distância percorrida devido ao uso do sensor *encoder* para aquisição da informação causando incerteza na pose do robô. Assim, é conveniente incorporar medidas de outros sensores, tais como giroscópio, bússola ou acelerômetro, com o objetivo de aumentar a precisão da medição da pose do robô. Entretanto, surge o problema de como integrar informações de diferentes sensores, considerando seus próprios ruídos e precisões, numa única medição. Uma técnica comumente usada para este fim na academia é o Filtro de Kalman (GREWAL; ANDREWS, 2008).

Foram implementadas duas fusões de sensores utilizando-se o Filtro de Kalman Estendido (EKF) para melhorar a informação da pose da cadeira de rodas robótica. Para realizar a fusão de dados é utilizado o estado  $\theta$  relacionado à orientação. A primeira fusão sensorial utiliza os dados obtidos por um giroscópio e pela odometria, já a segunda fusão sensorial utiliza os dados obtidos por um giroscópio, pela odometria e por um magnetômetro. O objetivo das fusões sensoriais é a de melhorar a informação da pose de uma cadeira de rodas robótica.

Os resultados foram validados num ambiente controlado, composto por uma câmera posicionada no teto, e utilizando um sistema de localização por imagens. Com isso, a pose obtida pela câmera é utilizada como base de comparação para os métodos

testados, além de ser utilizada como realimentação para um controlador de trajetória.

Ao final, são apresentados resultados práticos para mostrar os benefícios da fusão de dados, a qual melhora a estimativa de pose de uma cadeira de rodas robótica.

## 1.1 OBJETIVO GERAL

A informação da pose de um robô móvel é de essencial importância para que ele consiga realizar sua atividade fim, a navegação. A odometria é o método mais utilizado na literatura para a estimação da pose devido a vantagem deste método que o tempo de resposta pequeno, no entanto, tem como desvantagem o erro entre a pose real e a estimada que se acumula ao longo do tempo resultando em estimação de sua pose muito diferente da pose real (PANIAGUA, 2012). Para realizar a navegação dos robôs móveis é necessária a utilização de informações adquiridas de sensores. As informações adquiridas de sensores são contaminadas com ruído, que também estão presentes no modelo do cinemático do robô móvel usado.

O problema encontrado na navegação é a incerteza na informação da pose do robô móvel. Como os sensores, o método de odometria e o modelo do cinemático do robô móvel usados estão contaminados com ruído é necessário utilizar um método que consiga realiza a fusão dessas informações para a obtenção de uma melhor estimativa de pose, que é a função do EKF.

O objetivo final é estimar a pose (posição e orientação) de uma cadeira de rodas robótica através de fusão de sinais de odometria, giroscópio e magnetômetro utilizando Filtro de Kalman estendido. Em seguida, validar os resultados com a pose obtida por uma câmera.

## 1.2 METODOLOGIA

A metodologia empregada durante o desenvolvimento deste trabalho, a fim de alcançar o objetivo proposto, está detalhada a seguir:

- Estudo e simulação da plataforma robótica *Starter kit 2.0*, da *National Instruments* (NI) com a implementação de programas simples para conhecer seu funcionamento básico. Parte do conhecimento foi repassado por membros do Grupo de Pesquisa de Automação Industrial do Instituto Federal do Espírito Santo - GAin (Grupo de Estudo de Automação Industrial), que pesquisaram seu funcionamento.
- Implementação dos controladores de trajetória *Fuzzy*, *Backstepping* e ganhos fixos na simulação da plataforma robótica *Starterkit 2.0*.
- Criação do ambiente de testes para realizar os experimentos práticos, no qual é colocado uma lona emborrachada para evitar escorregamentos do robô, de modo a minimizar os erros causados pelas forças físicas envolvidas no sistema, e uma câmera é posicionada no teto para obter a pose do robô através de processamento de imagem. E também fixada uma placa com dois círculos de cores, que se destacam do restante do ambiente, e com diferentes tamanhos sobre o robô, para que a pose do robô seja calculada.
- Implementação dos controladores na plataforma robótica *Starter kit 2.0* real, com os ganhos obtidos para os controladores pelo método de tentativa e erro.
- Estudo do Filtro de Kalman com o intuito de aprender o seu funcionamento e realizar sua implementação.
- Estudo e simulação do Filtro de Kalman (EKF) para a fusão de dados dos sensores.
- Implementação da odometria e duas fusões sensoriais utilizando-se o EKF. A primeira fusão utiliza a odometria e o giroscópio. A segunda fusão utiliza a

odometria, o giroscópio e o magnetômetro. A odometria e as duas fusões são comparadas e os resultados são analisados.

- Esta Dissertação de Mestrado está organizada da seguinte forma: o no Capítulo 2 é apresentada a fundamentação teórica dos assuntos abordados para o desenvolvimento do trabalho. No Capítulo 3 é apresentada a descrição das plataformas robóticas utilizadas. No Capítulo 4 são explicadas as implementações realizadas para a geração do resultado final e no Capítulo 5 são apresentados os resultados alcançados e os trabalhos futuros.

## 2 FUNDAMENTAÇÃO TEÓRICA

A área da robótica vem sofrendo grandes avanços, além disso, a sua área de aplicação vem aumentando. São exemplos de aplicações comuns os robôs industriais (realizam atividades de pintura, soldagem, transporte de cargas), robôs cirúrgicos, robôs de acompanhamento de idosos e etc. Os robôs também podem operar em ambientes inhóspitos para seres humanos, como por exemplo, para procurar e desativar minas e bombas, procurar sobreviventes em zonas de desastres naturais e aplicar veneno em plantações. Portanto, a robótica se instalou no nosso entorno, trazendo vários benefícios (SECCHI, 2008).

### 2.1 TEORIA DO FILTRO DE KALMAN

O Filtro de Kalman (KF), teoricamente, é um estimador para um problema quadrático linear que consiste na estimação do estado instantâneo de um sistema linear dinâmico perturbado por ruído branco. Seu objetivo é utilizar medições de grandezas feitas ao longo do tempo (contaminadas com ruído e outras incertezas) e produzir resultados que visem se aproximar dos valores reais das grandezas medidas e valores associados. Portanto, o KF é uma ferramenta matemática que propõe encontrar valores que se aproximam das grandezas medidas a partir de informações do modelo e da medição, as quais estão contaminadas com ruídos e outras incertezas (GREWAL; ANDREWS, 2008).

O KF é uma caracterização estatística consistente de um problema de estimação. Porém, ele não se limita apenas à estimação, porque ele propaga o conhecimento do estado corrente do sistema dinâmico, incluindo a influência de dinâmicas aleatórias e os efeitos de todas as medições anteriores, ou seja, é uma ferramenta recursiva (PANIAGUA, 2012). Essas propriedades são extremamente úteis para a análise estatística de sistemas preditivos baseados em sensores, como é o caso de aplicação desta Dissertação de Mestrado.

### 2.1.1 Aplicações do filtro de Kalman

O KF pode ser aplicado para resolver as mais diversas gamas de problemas em áreas diversificadas como, por exemplo, análise em medicina, processos químicos, engenharia, entre outras. O problema de estimação pode envolver medição de um processo através de um sensor, ou de múltiplos processos, através de vários sensores, sendo que, esta estimação minimiza a média quadrática do erro do estado estimado (GELB, 1974).

Na utilização do KF, considera-se o sistema a ser estimado como linear e invariante no tempo, discreto, e com ruído. A equação de transição de estados é mostrada em (1) e a de observação é mostrada em (2),

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1}, \quad (1)$$

$$z_k = Hx_k + v_k, \quad (2)$$

onde  $x_k \in \mathbb{R}^n$  (representação de números reais em  $n$  dimensões) é o vetor de estados,  $u_k \in \mathbb{R}^u$  (representação de números reais em  $u$  dimensões) é o vetor de entrada, e  $z_k \in \mathbb{R}^m$  (representação de números reais em  $m$  dimensões) é o vetor de medições. Todos os vetores têm uma distribuição de probabilidade independente, branca e normal com média zero (Gaussiana). A matriz  $A$  relaciona o estado anterior com o estado atual e a matriz  $H$  relaciona o estado atual com a medição. Os ruídos do processo são representados por  $w_k$ , e as medições representadas por  $v_k$  no instante  $k$ . Para que o KF funcione de forma ótima não há necessidade que estes ruídos tenham distribuição gaussiana (SIMON, 2006). No KF, supõe-se que os valores das matrizes de covariância do erro do processo, representado por  $Q_k$ , e da matriz de covariância do erro das medições, representados por  $R_k$ , são conhecidos, no entanto, os valores  $w_k$  e  $v_k$  são geralmente desconhecidos (PANIAGUA, 2012). Dessa forma, estes valores devem ser descobertos. O valor inicial para o ruído dos sensores é, geralmente, definido como o desvio padrão do erro definido em seu *datasheet*. Depois do ajuste referente à covariância do ruído dos sensores é

necessária a realização do ajuste referente à covariância do ruído do processo. Para isso, também foi utilizado o método de tentativa e erro. O uso do KF minimiza os efeitos de  $w_k$  e  $v_k$  na estimação do estado.

O KF tem por objetivo realizar a estimativa do estado  $x_k$  baseado nas relações dinâmicas de um sistema linear utilizado, das perturbações inerentes deste sistema e das medições  $z_k$ , considerando que elas contêm ruído branco, e também, minimizar os ruídos do processo  $w_k$  e das medições  $v_k$  na estimação do estado. Com isso, o KF define  $\hat{x}_k^- \in \mathbb{R}^n$ , que é a estimação do estado a priori no instante  $k$ , o qual é obtido a partir do conhecimento do processo antes do instante de tempo  $k$  e  $\hat{x}_k \in \mathbb{R}^n$ , que é a estimação do estado a posteriori, no instante  $k$ , adquirida da medida  $z_k$ , na qual os erros estimados a priori e a posteriori são indicados em (3) e (4).

$$e_k^- = x_k - \hat{x}_k^- \quad (3)$$

$$e_k = x_k - \hat{x}_k \quad (4)$$

A estimação a priori e a posteriori da covariância do erro é demonstrada em (5) e (6), onde  $E$  representa o erro esperado,

$$P_k^- = E[e_k^- e_k^{-T}], \quad (5)$$

$$P_k = E[e_k e_k^T]. \quad (6)$$

O cálculo de  $\hat{x}_k \in \mathbb{R}^n$  do estado a posteriori é feita através de uma combinação linear da estimação a priori  $\hat{x}_k^- \in \mathbb{R}^n$ . O cálculo do erro estimado é a diferença entre a medida atual  $z_k$  e a predição da medida  $H\hat{x}_k^-$ , sendo que esta parcela  $z_k - H\hat{x}_k^-$  é denominada inovação ou resíduo. A inovação ou resíduo espelha a discordância entre medições e previsões para realizar a atualização da estimativa do estado a priori.  $K$  é uma matriz que representa o ganho de Kalman, e seu valor é encontrado

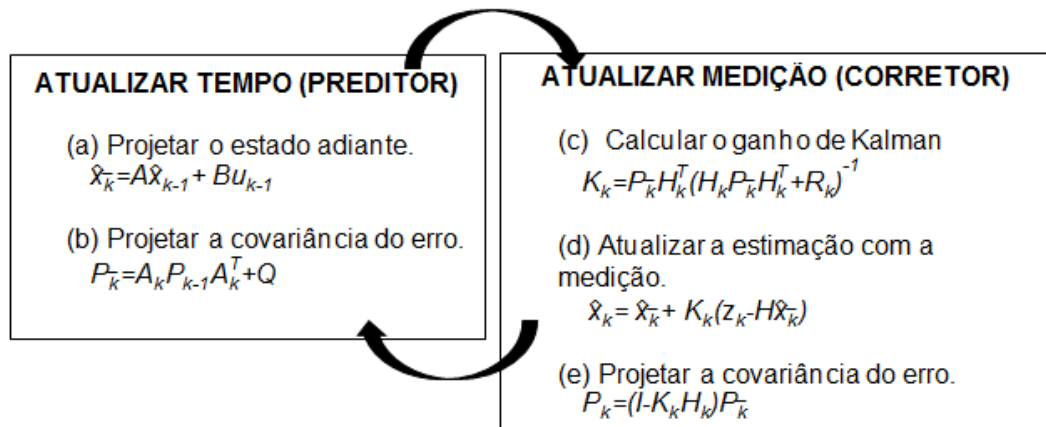


para minimizar a covariância do erro a posteriori  $P_k$ , a qual é determinada por uma abordagem probabilística usando as covariâncias de erro e ruído.

O resultado da substituição da equação mostrada na figura 1(a) em (3) e (4), mais especificamente  $e_k$ , depois de obtido, é utilizado para a substituição em  $P_k$ , conforme mostrado em (5) e (6). Minimizando (5) e (6) com respeito a  $K$  (diferenciando com relação a  $K$  e igualando a zero para retirar  $K$ ) se obtêm o ganho ótimo  $K$ , como mostrado na equação da figura 1(c). Na equação mostrada na figura 1(b), se necessita do valor da covariância a priori  $P_k^-$ , que é obtido a partir de (5) e (6), as quais, necessitam do conhecimento da variável  $x_k$ . Assim, chega-se na covariância mostrada na figura 1(b), sendo que o valor da covariância a posteriori  $P_k$  é obtida através de (5) e (6), como pode-se observar na figura 1(e). A partir da equação mostrada na figura 1(a), até a equação mostrada na figura 1(e), pode-se perceber que o KF pode ser implementado na forma de algoritmo, sendo dividido em duas etapas: predição e correção (WELCH; BISHOP, 2008):

- Predição: realiza o cálculo do estado a priori com o valor da estimativa anterior do estado e o valor de entrada corrente. Com isso, é realizado o cálculo da covariância a priori.
- Atualização: usa a medida atual para refinar o resultado dado pela etapa anterior para obter uma melhora na estimativa a posteriori.

Figura 1 - Algoritmo do KF



Fonte: Welch; Bishop (2008) adaptado.

### 2.1.2 Filtro de Kalman estendido

O KF é utilizado para sistemas lineares, sendo que no caso de sistemas não lineares é utilizado o filtro de Kalman estendido (*Extended Kalman Filter* - EKF), ou seja, este filtro é projetado para resolver o problema de processos e medições não lineares. Considera-se o sistema não linear invariante no tempo, discreto e com ruído mostrado em (7) e (8) (PANIAGUA, 2012). Nestas equações, podem-se verificar mudanças do EKF em relação ao KF. Em (7) o estado é definido por uma função de transição de estado, já em (1), ao invés disso, existe uma matriz  $A$  cujo papel é realizar a transição de estados. Caso a função de transição de estado seja não linear, é necessário que ela seja submetida a um processo de linearização e o mesmo acontece com (8) e (2). A equação (8), que é a função equivalente à observação (modelo de leitura dos sensores), deve ser linearizada caso seja não linear, sendo que em (2) a matriz  $H$  representa o modelo dos sensores,

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}), \quad (7)$$

$$z_k = h(x_k, v_k). \quad (8)$$

Da mesma forma que o KF, o EKF possui as variâncias de erro  $w_k$  e  $v_k$ , que representam ruídos do processo e medições no instante  $k$ . Os termos  $x_k \in R^n$  é o vetor de estado do processo,  $u_k \in R^u$  é o vetor de entrada,  $z_k \in R^m$  é o vetor de medidas, todos com uma distribuição de probabilidade independente, branca e normal com média zero (gaussiana). Como no KF, os valores exatos de  $w_k$  e  $v_k$  no instante  $k$  geralmente são desconhecidos, mas assume-se que se tem um certo conhecimento de suas matrizes de covariância  $Q_k$  e  $R_k$ , que são pertencentes ao processo e às medições, respectivamente. Formalmente, o processo e as medições são variantes no tempo, mas, neste trabalho, será considerado como invariante no tempo. Pode ser que se conheçam os valores das matrizes de covariância  $Q_k$  e  $R_k$ , no entanto, é mais comum o conhecimento do valor para  $R_k$ , apesar de se conseguir

encontrar o valor de  $Q_k$  através do método de tentativa e erro com experimentação de valores e observação do comportamento do KF.

A representação  $x_k^- \in R^n$  é o estado a priori no instante  $k$  obtido do conhecimento do processo antes do instante de tempo  $k$ , e  $\hat{x}_k \in R^n$  é a estimativa do estado a posteriori no instante  $k$  usando a medida  $z_k$ . O cálculo dos erros estimados a priori é realizado na etapa de predição e depende apenas do modelo. Por outro lado, o cálculo dos erros estimados a posteriori é realizado na etapa de atualização e são ajustados com a obtenção da medição definindo (3) e (4). Essas equações são as mesmas utilizadas no KF (PANIAGUA, 2012), sendo que a estimativa a priori e a posteriori da covariância do erro define (5) e (6), que são similares ao KF, onde  $E$  é o valor esperado.

A função não linear  $h(x_k, v_k)$  relaciona o estado  $x_k$  com as medições  $z_k$ . O estado e as medições no instante inicial podem ser aproximados considerando a resposta do sistema ante a entrada anterior sem a presença do ruído, como mostrado em (9) e (10) (PANIAGUA, 2012),

$$\hat{x}_k = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (9)$$

$$\hat{z}_k = h(\hat{x}_k, 0) \quad (10)$$

sendo  $\hat{x}_k$  a estimativa do estado a posteriori (após a realização da medição) no instante de tempo anterior  $k-1$ . Para se utilizar as equações do filtro pertencente ao filtro de Kalman linear, linearizam-se (3) e (4), obtendo-se (11) e (12) (PANIAGUA, 2012),

$$x_k \approx \hat{x}_k^- + A(x_{k-1} - \hat{x}_{k-1}) + w_{k-1} \quad (11)$$

$$z_k \approx \hat{z}_k^- + H(x_k - \hat{x}_k^-) + v_k \quad (12)$$

onde  $\hat{x}_k$  e  $\hat{z}_k$  são as estimações a partir de (7) e (8). O cálculo das matrizes  $A$  e  $H$  é representada por funções mostradas em (13) e (14) no EKF e, geralmente, são não lineares e estão contidas em (11) e (12). O cálculo dos ruídos do processo e da medição são mostrados em (15) e (16) (PANIAGUA, 2012),

$$A_k = \frac{\partial f}{\partial x} \Big|_{\hat{x}_{k-1} + u_{k-1}, 0} \quad (13)$$

$$H_k = \frac{\partial h}{\partial x} \Big|_{\hat{x}_{k-1}, 0} \quad (14)$$

$$W_k = \frac{\partial f}{\partial w} \Big|_{\hat{x}_{k-1} + u_{k-1}, 0} \quad (15)$$

$$V_k = \frac{\partial h}{\partial v} \Big|_{\hat{x}_k, 0} \quad (16)$$

Na tabela 1 são mostrados os símbolos, que geralmente são matrizes, utilizados nas equações do EKF juntamente com seu significado e dimensões (tamanho de suas linhas e de suas colunas) das matrizes relacionadas a cada um desses símbolos.

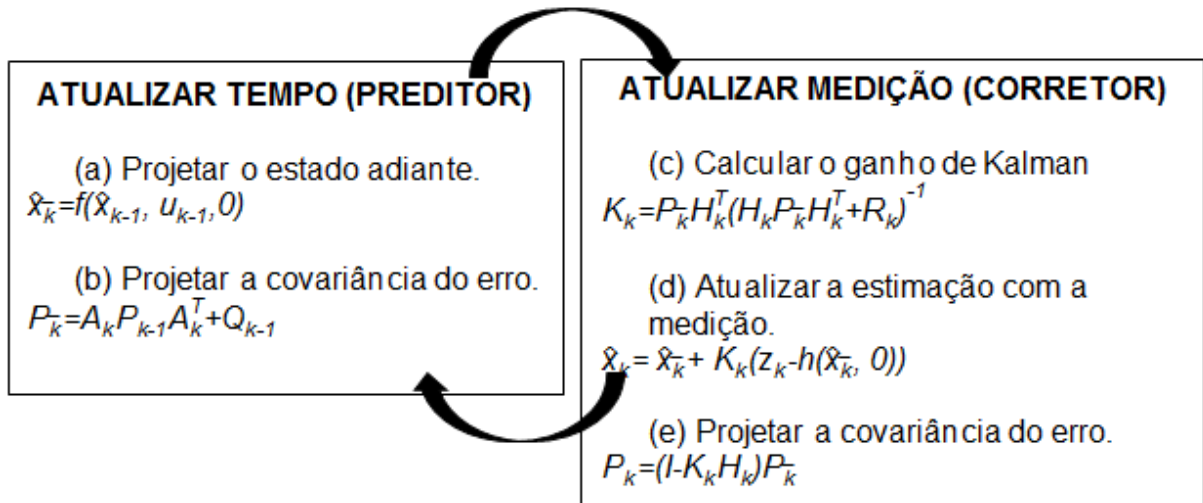
Tabela 1 - Notações comuns para dimensões de Array.

Símbolo	Nome do vetor	Dimensões	Símbolo	Nome da Matriz	Linha	Coluna
$x$	Estado do sistema	$N$	$A$	Transição de estado	$n$	$n$
$w$	Ruído do processo	$R$	$W$	Ruído do processo	$n$	$r$
$u$	Sinal de controle	$R$	$Q$	Covariância do ruído do	$r$	$r$
$\hat{x}_k$	Medição	$L$	$H$	Medição	$l$	$n$
$r$	Ruído da medição	$L$	$R$	Covariância do ruído da medição	$l$	$l$

Fonte: Grewal; Andrews (2008) adaptado.

O EKF pode ser representado como um algoritmo, conforme mostra a figura 2.

Figura 2 - Algoritmo do EKF



Fonte: Welch; Bishop (2008) adaptado.

## 2.2 CONTROLADOR DE SEGUIMENTO DE TRAJETRIA

O controlador de trajetria tem por funo aplicar sinais de controle sobre os atuadores do rob para que siga uma determinada trajetria. Em Tommasi et al. (2015) foi realizada uma pesquisa a fim de comparar o desempenho de trs controladores de trajetria para robs de trao diferencial, utilizando apenas odometria como medida da estimativa da pose do rob com o objetivo de encontrar o melhor controlador. Nesta Dissertao de Mestrado, foi realizada a mesma comparao realizada por Tommasi et al. (2015), utilizando-se os mesmos trs controladores de trajetria, com a diferena de se incluir um mtodo de estimativa da pose do rob, utilizando o processamento de imagens ao invs de odometria. Com isso, pde-se verificar uma melhoria na preciso devido ao processamento de imagem, que consegue obter a pose do rob mvel com um erro muito menor em comparao com a odometria.

Os controladores estudados nas comparações são: controlador cinemático de seguimento de trajetória de ganhos fixos (MARTINS, 2009) e (RESENDE et al., 2011), controlador *Backstepping* (FIERRO; LEWIS, 1998) e (YE, 2008), e controlador Fuzzy do tipo Mandani (TOMMASI et al., 2015). A propósito, a escolha destes controladores foi realizada por Tommasi et al. (2015). A descrição de cada controlador será realizada nas próximas seções.

Para realizar o teste de seguimento de trajetória é necessário que seja projetado um gerador de trajetórias que tenha como função enviar valores de referência ao controlador. Na literatura, a forma mais utilizada para gerar uma trajetória a ser seguida pelo robô é o *lemniscate* (curva plana em forma do símbolo de infinito) devido a ela testar toda a dinâmica do robô (TOMMASI et al., 2015). O *lemniscate* é uma boa abordagem de teste para os controladores de trajetória, pois espera-se a ocorrência de erros devido às forças físicas envolvidas na dinâmica do robô que devem ser compensadas pelo controlador. Portanto, o gerador de trajetória é quem define a referência em  $x$ ,  $y$  no tempo, e esses serão os valores de referência para o controlador de trajetória.

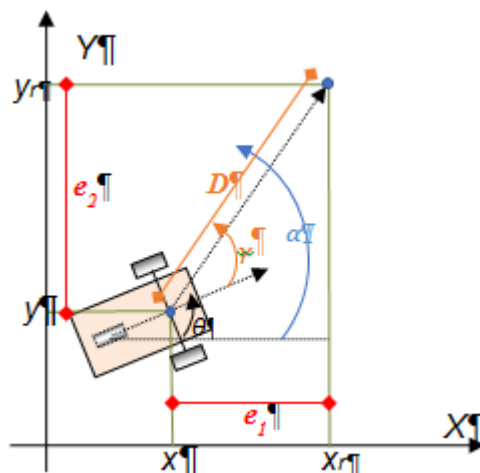
A técnica odométrica utilizada por Tommasi et al. (2015) foi o *Dead Reckoning*, com sua função de estimar a pose do robô. Essa é uma técnica que usa a integração de medições dos deslocamentos incrementais das rodas do robô, no entanto, com a integração das medições, também se integram os erros (VALGAS FILHO, 2002). A integração dos erros compromete os resultados obtidos, devido às curvas realizadas pelo robô, as quais acarretam deslizamentos nas rodas provocados pelas forças físicas envolvidas (YU et al., 2010). Portanto, para que não houvesse acúmulo de erros foi utilizado o processamento de imagens para se obter a localização e, assim, comparar os controladores, nesta dissertação.

A plataforma de desenvolvimento utilizada para encontrar o melhor controlador de trajetória foi o *Start Kit 2.0* (DaNI). Esta plataforma robótica conta com um ambiente de simulação de robôs contido no *software* de desenvolvimento Labview<sup>TM</sup>. Portanto, o objetivo buscado neste experimento foi a repetição dos resultados conseguidos no DaNI para uma cadeira de rodas elétrica, validando o *Backstepping* como melhor controlador.

### 2.2.1 Controlador Fuzzy

O controlador de trajetória *Fuzzy* foi projetado com as seguintes etapas: definição das variáveis de entrada e saída, criação das funções de pertinência, criação das regras do controlador, definição do método de “defuzzificação”, e ajuste de ganhos das variáveis de saída. Na **Erro! Fonte de referência não encontrada.** é mostrado esquema de erros, os quais tendem a serem corrigidos pelo controlador Fuzzy. As regras do controlador *Fuzzy* foram definidas a partir de duas entradas: o ângulo  $\gamma$  (erro de orientação) e a distância  $D$  (erro de posição) (FARIA et al., 2016).

Figura 3 - Erros do controlador Fuzzy



Fonte: Tommasi et al. (2015).

As regras Fuzzy são apresentadas com duas entradas: o ângulo  $\gamma$  e a distância  $D$ , mostrados na figura 3. O erro de orientação  $\gamma$  é calculado conforme (17),

$$\gamma = \alpha - \theta, \quad (17)$$

onde  $\alpha$  é o ângulo do vetor  $D$ , e  $\theta$  o ângulo de orientação do robô. Já o erro de posição  $D$  é obtido através de (18),

$$D = \sqrt{(x_r-x)^2+(y_r-y)^2}, \quad (18)$$

onde  $e_1 = x_r - x$ , e  $e_2 = y_r - y$  são os erros em  $x$  e  $y$ , respectivamente.

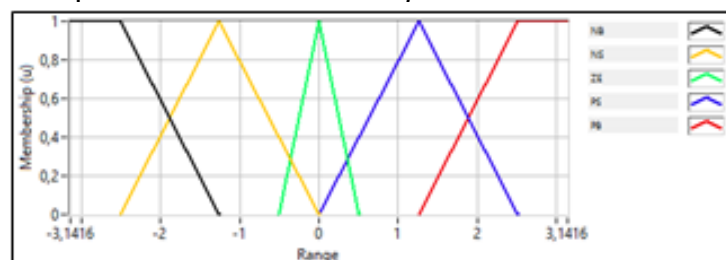
As duas saídas do controlador *Fuzzy* foram definidas como sendo as velocidades linear  $v$  e angular  $w$  de referência para o robô. A velocidade linear de referência será utilizada para minimizar o erro  $D$ , assim como a velocidade angular de referência será utilizada para reduzir o erro  $\gamma$ . As funções membros descritas nas figuras 4, 5, 6 e 7 possuem a forma de triângulos e trapezoides divididas em cinco partições, as quais são mostradas na tabela 2.

Tabela 2 - Variáveis *Fuzzy*.

Variável	Símbolo	Descrição
$\gamma$	NB	Negative Big
	NS	Negative Small
	ZE	Zero
	PS	Positive Small
	PB	Positive Big
$D e v$	VS	Very Small
	S	Small
	M	Medium
$w$	B	Big
	VB	Very Big
	LF	Left Fast
	LS	Left Slow
	Z	Zero
	RS	Right Slow
	RF	Right Fast

Fonte: Faria et al. (2016).

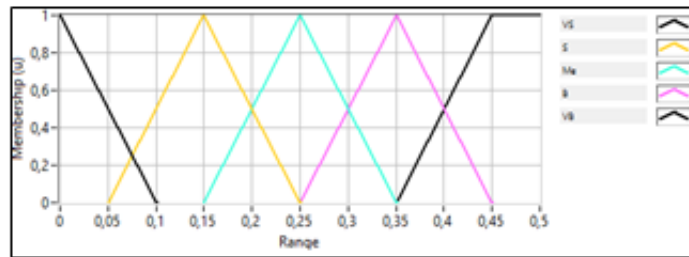
Figura 4 - Função de pertinência da entrada  $\gamma$ .



Fonte: Faria et al. (2016).

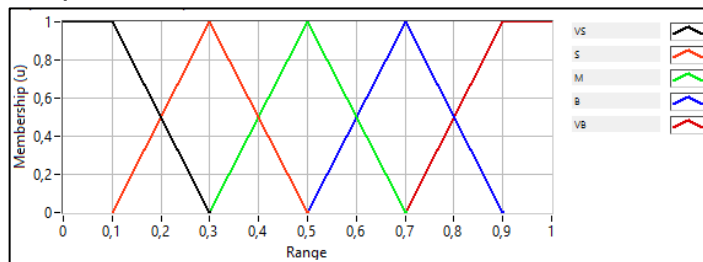


Figura 5 - Função de pertinência da entrada  $D$ .



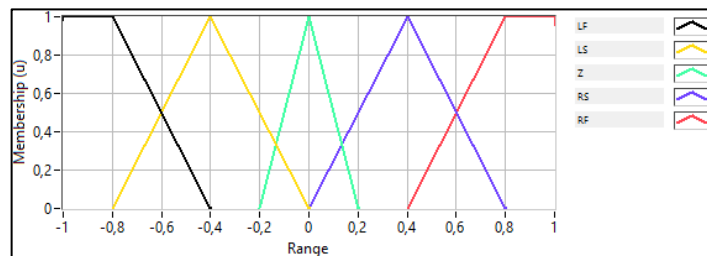
Fonte: Faria et al. (2016).

Figura 6 - Função de pertinência da entrada  $v$ .



Fonte: Faria et al. (2016).

Figura 7 - Função de pertinência da entrada  $w$ .



Fonte: Faria et al. (2016).

As regras *Fuzzy* são mostradas na tabela 3. Os elementos na primeira coluna e linha são as variáveis linguísticas de entrada ( $\gamma$  e  $D$ ), enquanto as outras correspondem às variáveis linguísticas de saída ( $v$  e  $w$ ).

Tabela 3 - Regras Fuzzy.

		$\gamma$				
		<b>NB</b>	<b>NS</b>	<b>ZE</b>	<b>PS</b>	<b>PB</b>
<b>D</b>	<b>VS</b>	VS (RF)	VS (RS)	VS (Z)	VS (LS)	VS (LF)
	<b>S</b>	S (RF)	S (RS)	S (Z)	S (LS)	S (LF)
	<b>M</b>	Me (RF)	Me (RS)	Me (Z)	Me (LS)	Me (LF)
	<b>B</b>	B (RF)	B (RS)	B (Z)	B (LS)	B (LF)
	<b>VB</b>	VB (RF)	VB (RS)	VB (Z)	VB (LS)	VB (LF)

Fonte: Faria et al. (2016).

Existem vinte e cinco regras aplicadas ao controlador conforme tabela 4.

Tabela 4 - Regras Fuzzy detalhadas.

1: If $D$ is VS and $\gamma$ is NB then $v$ is VS and $w$ is LF
2: If $D$ is S and $\gamma$ is NB then $v$ is S and $w$ is LF
3: If $D$ is M and $\gamma$ is NB then $v$ is Me and $w$ is LF
4: If $D$ is B and $\gamma$ is NB then $v$ is B and $w$ is LF
5: If $D$ is VB and $\gamma$ is NB then $v$ is VB and $w$ is LF
6: If $D$ is VS and $\gamma$ is NS then $v$ is VS and $w$ is LS
7: If $D$ is S and $\gamma$ is NS then $v$ is S and $w$ is LS
8: If $D$ is M and $\gamma$ is NS then $v$ is Me and $w$ is LS
9: If $D$ is B and $\gamma$ is NS then $v$ is B and $w$ is LS
10: If $D$ is VB and $\gamma$ is NS then $v$ is VF and $w$ is LS
11: If $D$ is VS and $\gamma$ is M then $v$ is VB and $w$ is Z
12: If $D$ is S and $\gamma$ is M then $v$ is S and $w$ is Z
13: If $D$ is M and $\gamma$ is M then $v$ is Me and $w$ is Z
14: If $D$ is B and $\gamma$ is M then $v$ is B and $w$ is Z
15: If $D$ is VG and $\gamma$ is M then $v$ is VB and $w$ is Z
16: If $D$ is VS and $\gamma$ is PS then $v$ is VS and $w$ is RS
17: If $D$ is S and $\gamma$ is PS then $v$ is S and $w$ is RS
18: If $D$ is M and $\gamma$ is PS then $v$ is Me and $w$ is RS
19: If $D$ is B and $\gamma$ is PS then $v$ is B and $w$ is RS
20: If $D$ is VB and $\gamma$ is PS then $v$ is VB and $w$ is RS
21: If $D$ is VS and $\gamma$ is PB then $v$ is VS and $w$ is RF
22: If $D$ is S and $\gamma$ is PB then $v$ is S and $w$ is RF
23: If $D$ is M and $\gamma$ is PB then $v$ is Me and $w$ is RF
24: If $D$ is V and $\gamma$ is PB then $v$ is B and $w$ is RF
25: If $D$ is VB and $\gamma$ is PB then $v$ is VB and $w$ is RF

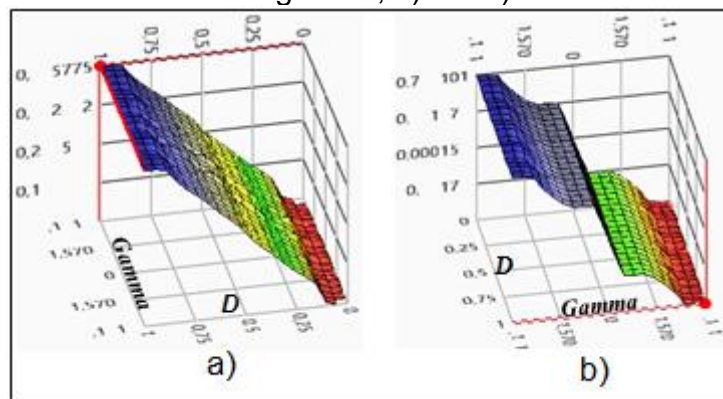
Fonte: Faria et al. (2016)

Como método de “defuzzificação” é utilizado o centro de área (*Center of Area - CoA*), também chamado de método de centro de gravidade (*Center of Gravity - CoG*). O controlador de lógica *Fuzzy* usa o cálculo do centro geométrico desta área como mostrado em (19),

$$CoA = \frac{\int_{x_{min}}^{x_{max}} f(x) \cdot x \, dx}{\int_{x_{min}}^{x_{max}} f(x) \, dx}, \quad (19)$$

onde, CoA é o centro da área,  $x$  é o valor da variável linguística,  $x_{min}$  e  $x_{max}$  representam o intervalo da variável linguística. A figura 8a mostra a relação das entradas  $\gamma$  e  $D$  com a saída  $v$ . A figura 8b) mostra a relação das entradas  $\gamma$  e  $D$  com a saída  $w$ .

Figura 8 - Curvas de entrada-saída globais, a)  $v$  e b)  $w$ .



Fonte: Faria et al. (2016)

Na aplicação, cada saída ( $v$  e  $w$ ) do controlador é multiplicada por um fator. As variáveis  $K_v$  e  $K_w$  representam o ganho de velocidade linear e angular, respectivamente. Esses fatores devem ser sintonizados para obter um resultado melhor.

### 2.2.2 Controlador *Backstepping*

O segundo controlador testado foi o *Backstepping*, cuja ideia é utilizar variáveis de estado  $q=[x \ y \ \theta]^T$ , as quais representam as poses que podem ser assumidas pelo robô, como “controles virtuais”. O desenvolvimento foi baseado nos trabalhos de (FIERRO; LEWIS, 1998) e (YE, 2008). Em (30) é apresentada a lei de controle do controlador *Backstepping* utilizado em Kanayama et al. (1990) e também utilizada e recomendada nos trabalhos de Fierro; Lewis (1998), Jiang; Nijmeijer (1997) e Fukao; Nakagawa e Adachi (2000).

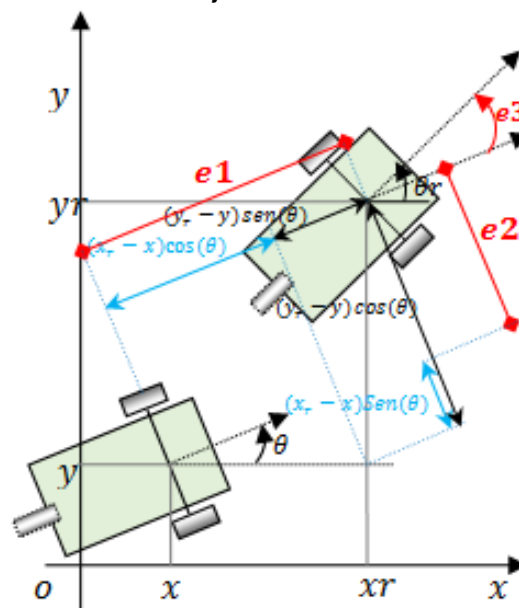
Em (20) são mostrados os cálculos dos erros usados no desenvolvimento desse controlador, os quais estão organizados na forma matricial,

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_r - x \\ y_r - y \\ \theta_r - \theta \end{bmatrix} \quad (20)$$

onde,  $e_1$  é correspondente ao erro no eixo das abscissas ( $x$ ),  $e_2$  é correspondente ao erro no eixo das ordenadas ( $y$ ), e  $e_3$  corresponde ao erro de orientação ( $\theta$ ).

A figura 9 apresenta os erros considerados no desenvolvimento desse controlador.

Figura 9 - Disposição dos erros de trajetória.



Fonte: Tommasi et al. (2015).

Em (21) é apresentada a lei de controle do controlador *Backstepping* utilizada nesta Dissertação de Mestrado,

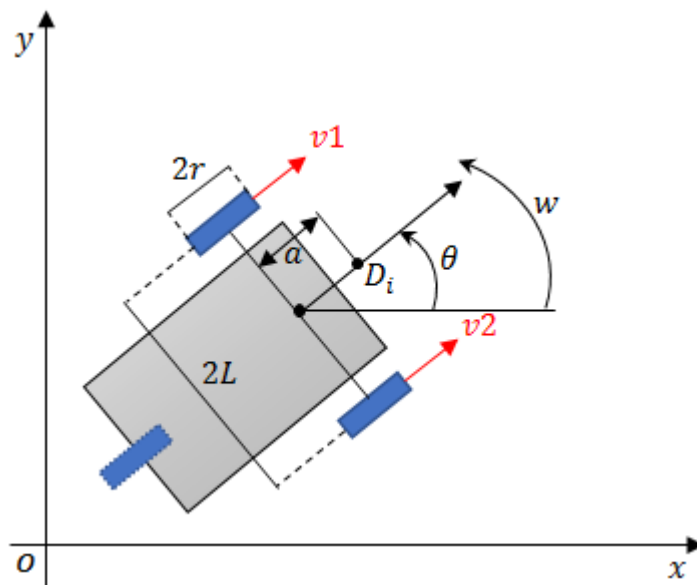
$$\begin{bmatrix} v_c \\ w_c \end{bmatrix} = \begin{bmatrix} v_r \cos e_3 + k_1 e_1 \\ w_r + k_2 v_r e_2 + k_3 v_r \sin e_3 \end{bmatrix} \quad (21)$$

onde  $k_1$ ,  $k_2$  e  $k_3$  são parâmetros de sintonia do controlador,  $v_c$  e  $w_c$  são as saídas do controlador. Os ganhos  $k_1$ ,  $k_2$  e  $k_3$  devem ser tais que o erro de posição do robô seja minimizado.

### 2.2.3 Controlador de ganhos fixos

Em Tommasi et al. (2015) foi demonstrado o modelo cinemático não holonômico de acessibilidade melhorada utilizado no desenvolvimento do controlador cinemático de ganhos fixos. Conforme esse modelo, a pose do robô é definida pelo ponto de interesse  $D_i$ , conforme a figura 10.

Figura 10 - Modelo cinemático não holonômico de acessibilidade melhorada



Fonte: Tommasi et al. (2015).

A partir do modelo mostrado na figura 10 chega-se a (22). Essa equação representa o modelo de acessibilidade melhorada utilizado no desenvolvimento do controlador cinemático de trajetória de ganhos fixos,

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & -a \cdot \sin\theta \\ -\sin\theta & a \cdot \cos\theta \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \quad (22)$$

Para cálculos posteriores é definida a matriz  $A$ , definida em (23), a partir de (22),

$$A^{-1} = \begin{bmatrix} \cos\theta & -a \cdot \sin\theta \\ -\sin\theta & a \cdot \cos\theta \end{bmatrix} \quad (23)$$

Pode-se obter, também, a matriz inversa de  $A$ , descrita em (24),

$$A^{-1} = \begin{bmatrix} \cos\theta & \sin\theta \\ \frac{1}{a} \sin\theta & \frac{1}{a} \cdot \cos\theta \end{bmatrix} \quad (24)$$

A matriz inversa  $A^{-1}$ , mostrada em (24), é usada para a obtenção da cinemática inversa do robô, mostrada em (25),

$$\begin{bmatrix} v(t) \\ w(t) \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ \frac{1}{a} \sin\theta & \frac{1}{a} \cdot \cos\theta \end{bmatrix} \begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} \quad (25)$$

A partir das equações demonstradas é possível utilizar a seguinte lei de controle, mostrada em (26),

$$\begin{bmatrix} Vr \\ wr \end{bmatrix} = A^{-1} \begin{bmatrix} \dot{x}_d + l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}(t)\right) \\ \dot{y}_d + l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}(t)\right) \end{bmatrix}, \quad (26)$$

onde  $Vr$  e  $wr$  são as saídas do controlador cinemático,  $\dot{x}_d$  e  $\dot{y}_d$  são as velocidades das posições desejadas em  $x$  e  $y$ ,  $\tilde{x} = x_d - x$  e  $\tilde{y} = y_d - y$  são os erros de posição nos eixos

$x$  e  $y$ ,  $k_x > 0$  e  $k_y > 0$  são os ganhos do controlador,  $l_x > 0$  e  $l_y > 0$  são constantes de saturação, e  $(x, y)$  e  $(x_d, y_d)$  são as coordenadas atual e desejada do ponto de interesse do robô. Para o sistema em malha fechada, considerando-se o seguimento perfeito de velocidade, ou seja,  $v \equiv v_r$  e  $\omega \equiv \omega_r$ , podem-se determinar os valores de velocidade do ponto de interesse do robô nas direções  $x$  e  $y$ , substituindo-se (26) em (22), e obtendo-se (27),

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_d + l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}(t)\right) \\ \dot{y}_d + l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}(t)\right) \end{bmatrix}. \quad (27)$$

Para erros de posição elevados (27) pode ser aproximada como em (28),

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}_d(t) + l_x \operatorname{sign}(\tilde{x}) \\ \dot{y}_d(t) + l_y \operatorname{sign}(\tilde{y}) \end{bmatrix}, \quad (28)$$

uma vez que para  $|\tilde{x}| \gg 0$ ,  $l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) \approx l_x \operatorname{sign}(\tilde{x})$ , e para  $|\tilde{y}| \gg 0$ ,  $l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \approx l_y \operatorname{sign}(\tilde{y})$ , onde  $\operatorname{sign}(\cdot)$  representa o sinal do argumento. Assim, os valores das constantes de saturação  $l_x$  e  $l_y$  mostradas em (27) podem ser determinadas a partir das seguintes considerações de velocidade máxima:

- Limites de velocidade para a trajetória planejada:  $|\dot{x}_d|, |\dot{y}_d| \leq T_{V \text{ máx}} \left[ \frac{m}{s} \right]$ ;
- Limites de velocidade para o ponto de interesse do robô nas direções  $x$  e  $y$ :  $|\dot{x}|, |\dot{y}| \leq R_{V \text{ máx}} \left[ \frac{m}{s} \right]$ ;

Considerando que a velocidade da trajetória planejada tenha o valor máximo permitido, ou seja,  $|\dot{x}_d|, |\dot{y}_d| \leq T_{V \text{ máx}} \left[ \frac{m}{s} \right]$ , para que os limites de velocidade para o ponto de interesse do robô sejam cumpridos, tem-se:  $|T_{V \text{ máx}} + l_x \operatorname{sign}(\tilde{x})| \leq R_{V \text{ máx}}$  e  $|T_{V \text{ máx}} + l_y \operatorname{sign}(\tilde{y})| \leq R_{V \text{ máx}}$ , concluindo-se que a adequação dos valores para as constantes de saturação é dada por:  $l_x = l_y = R_{V \text{ máx}} - T_{V \text{ máx}}$ .

Quanto maior os valores de  $l_x$  e  $l_y$ , mais rápido o robô irá alcançar a trajetória de referência. Já os valores adequados dos ganhos  $k_x$  e  $k_y$  são determinados de forma experimental, que consiste em executar o algoritmo de controle em tempo real no robô e ajustar os ganhos de forma que o robô siga a trajetória de forma mais adequada.

### 2.3 DEAD RECKONING

Métodos de navegação de robôs móveis requerem medições da pose do robô em seu espaço de trabalho, sendo comumente utilizada em odometria (GANGANATH; LEUNG, 2012). O *Dead Reckoning* é um método baseado em odometria que permite estimar a pose do robô através da integração das medições dos deslocamentos das rodas em certo intervalo de tempo (*time step*) por meio de sensores chamados *encoders* (codificadores ópticos). Os *encoders* geram pulsos no giro do eixo ao qual estão acoplados. No entanto, este método está sujeito a erros que se acumulam de acordo com o espaço percorrido pelo robô, devido aos deslizamentos das rodas, assimetrias do robô, dentre outros. Estes erros inviabilizam o uso único do *Dead Reckoning* para a determinação da pose de um robô (BEZERRA, 2004).

As informações de revolução de pulsos por volta do *Starter Kit 2.0* foram retiradas do seu *Datasheet* feito por Ibarra-Manzano et al. (2012). Já para a cadeira de rodas robótica os dados de revolução de pulsos por volta dos *encoders* foram obtidos experimentalmente. Os *encoders* foram fixados de forma a manterem contato com a correia de cada roda, com isso, percebe-se um número muito alto de pulsos por volta, conforme mostra a tabela 5.



Tabela 5 - Dados das plataformas utilizados no *Dead Reckoning*.

Plataformas	raio (cm)	Eixo entre rodas ( $2L$ ) (cm)	Pulsos por volta (Roda esquerda)	Pulsos por volta (Roda direita)
<i>Starter kit 2.0</i>	5,09	36,50	400	400
Modelo da cadeira de rodas (Freedom Ltda)	24,00	58,00	50880	56350

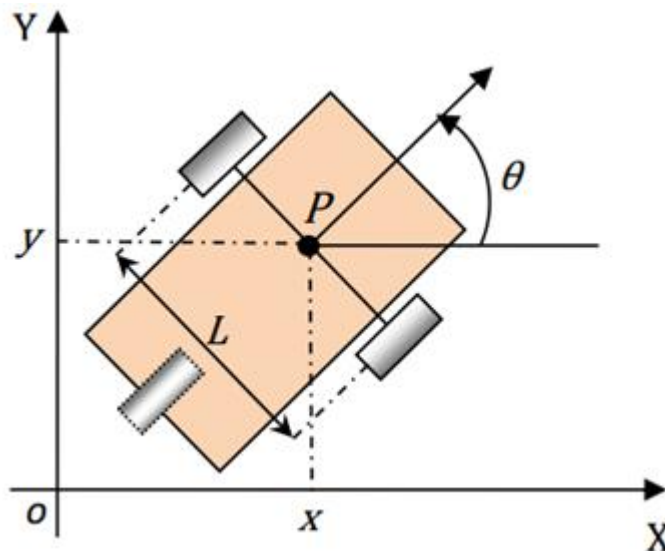
Fonte: Próprio autor (2017).

Para efetuar o cálculo da posição do robô (ponto  $P$  - ponto médio entre as rodas que é usado como ponto de referência do robô), conforme mostra a figura 11, é necessário obter a distância de cada pulso do *encoder*, mostrado em (29),

$$DistPulsos = \frac{2\pi R}{NP}, \quad (29)$$

onde,  $DistPulsos$  é a distância de cada pulso,  $R$  é o raio da roda e  $NP$  é a quantidade de pulsos por volta.

Figura 11- Referência global do robô



Fonte: Tommasi et al. (2015) adaptado.

Em seguida, é calculada a distância percorrida por cada roda. Em (30) e (31) encontra-se a distância percorrida pela roda direita e esquerda do robô, respectivamente,

$$\Delta S_d = \text{DistPulsos} * P_d, \quad (30)$$

$$\Delta S_e = \text{DistPulsos} * P_e, \quad (31)$$

onde,  $\Delta S_d$  e  $\Delta S_e$  são distâncias percorridas pelas rodas direita e esquerda, respectivamente, no intervalo  $\Delta t$ ;  $P_d$  e  $P_e$  são as quantidades de pulsos dos *encoders* das rodas direita e esquerda, respectivamente. A distância percorrida pelo ponto médio, referência do robô ( $P$ ), é dada por (32),

$$\Delta S = (\Delta S_d + \Delta S_e) / 2 \quad (32)$$

Com intuito de se obter a variação do ângulo  $\theta$ , é necessário usar (33),

$$\Delta \theta = (\Delta S_d - \Delta S_e) / L \quad (33)$$

onde  $L$  é a distância entre as rodas. Em seguida, são calculados os deslocamentos no eixo  $x$  e  $y$  no intervalo de tempo  $\Delta t$  mostrados em (34) e (35),

$$\Delta x = \Delta S * \cos(\theta + \Delta \theta / 2) \quad (34)$$

$$\Delta y = \Delta S * \sin(\theta + \Delta \theta / 2). \quad (35)$$

Com as variações de deslocamento  $\Delta \theta$  e de  $\theta$ , pode-se calcular a posição atualizada ( $P'$ ), como mostrado em (36),

$$P' = \begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = P' + \begin{bmatrix} \Delta x = \Delta S \cdot \cos(\theta + \Delta\theta/2) \\ \Delta y = \Delta S \cdot \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} \Delta x = \Delta S \cdot \cos(\theta + \Delta\theta/2) \\ \Delta y = \Delta S \cdot \sin(\theta + \Delta\theta/2) \\ \Delta\theta \end{bmatrix} = \begin{bmatrix} x + \Delta x \\ y + \Delta y \\ \theta + \Delta\theta \end{bmatrix}. \quad (36)$$

## 2.4 ESQUEMA DE FUSÃO DE DADOS COM O FILTRO DE KALMAN

A navegação dos robôs móveis tem como principal dificuldade a obtenção precisas das suas poses (THRUN, 2002). Por isso, é importante obter a pose com a máxima precisão possível, através de vários sensores, uma vez que as informações de cada sensor, separadamente, são imperfeitas. Desse modo, a implementação de um algoritmo para obter a pose de um robô deve usar o ponto mais forte de cada sensor e diminuir os seus pontos fracos. Um meio de garantir a qualidade na informação da posição e orientação providos é a combinação de dados de sensores através de fusão sensorial.

A determinação da pose baseado na informação de vários sensores é uma técnica relativamente antiga. Pode-se definir fusão sensorial como o processo que lida com a associação, correlação e combinação de dados e informações de uma ou mais fontes sensoriais a fim de obter o posicionamento refinado e estimativas de identidade (LIGGINS; HALL; LLINAS, 2008).

Se os dados dos sensores têm entre si medidas comuns, é interessante que sejam combinados de forma direta, podendo-se utilizar métodos clássicos de estimação como o Filtro de Kalman (ANTONY, 1995).

A fusão sensorial foi feita segundo Levy (2017), realizando-se as devidas alterações, mas ao invés de utilizar o KF se utilizou o EKF, dadas as características do modelo utilizado. Em (37) é mostrada a equação do modelo, em (38) é mostrada a equação da observação, em (39) é mostrada a equação da estimação do modelo a priori, em (40) é mostrada a equação da probabilidade do erro a priori para o momento da predição, em (41) é mostrada a equação do ganho de Kalman, em (42) é mostrada a

equação da estimação do modelo a posteriori, e em (43) é mostrada a equação da probabilidade do erro a posteriori para o momento da atualização.

$$x_k = f(x_{k-1}, u_{k-1}, 0) \quad (37)$$

$$z_k = h(x_k, 0) \quad (38)$$

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \quad (39)$$

$$P_k^- = A_k P_{k-1} A_k^T + Q_{k-1} \quad (40)$$

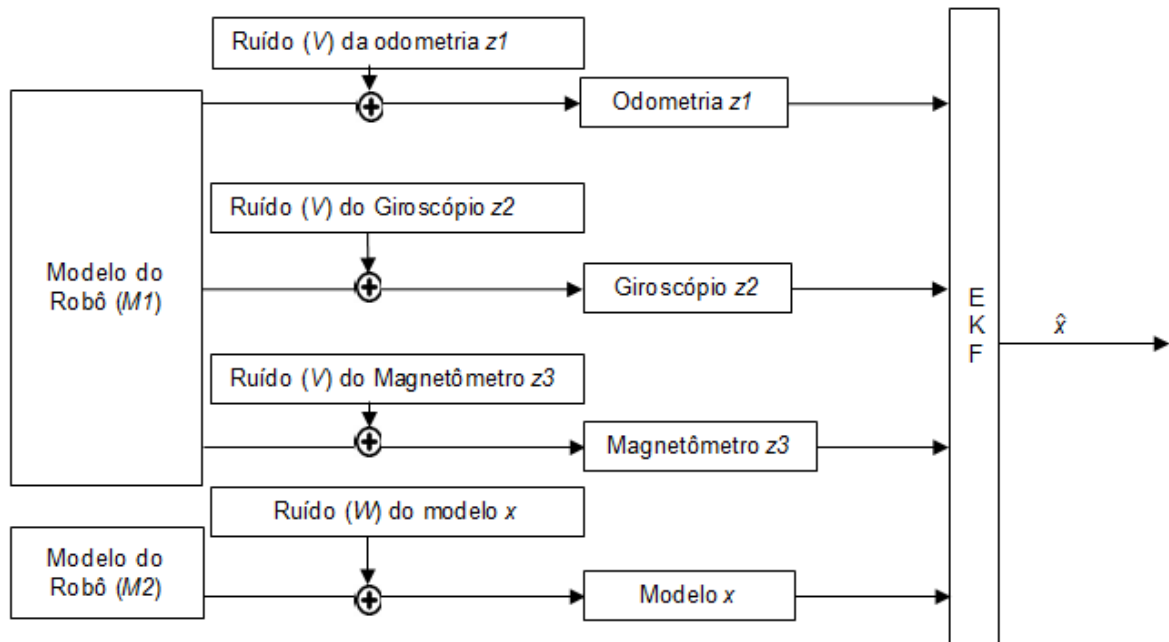
$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \quad (41)$$

$$\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-, 0)) \quad (42)$$

$$P_k = (I - K_k H_k) P_k^- \quad (43)$$

Dadas as equações, é necessária a realização da montagem da arquitetura do EKF, onde se darão três medições: uma proveniente da odometria e as outras duas do giroscópio e do magnetômetro, como mostra a figura 12. O modelo cinemático do robô móvel não-holonômico é utilizado tanto para o modelo, quanto para a medição. Para a realização das medições é utilizada uma instância do modelo do robô (*M1*). Os sensores utilizados para se obter a orientação da cadeira de rodas são o giroscópio (*Z2*) e o magnetômetro (*Z3*), os quais estão embutidos em uma IMU (*Inertial Measurement Unit*), que é um dispositivo que contém diversos sensores, sendo que a odometria (*Z1*) faz a estimativa do estado da pose da cadeira de rodas robótica. Uma outra instancia do modelo do robô será utilizada para o modelo (*M2*). As saídas da odometria, dos sensores e do modelo são as entradas do EKF. A saída do EKF será a estimativa ótima do estado da pose do robô, conforme é mostrado na figura 12. As dimensões da matriz de estado do EKF são iguais à matriz de estados do robô  $x_{3 \times 1}$  que é representada por  $q = [x \ y \ \theta]^T$ .

Figura 12 - Modelo de arquitetura da fusão sensorial do EKF e saída da estimativa.



Fonte: Próprio autor (2017).

Os sensores têm a função de extrair informações do robô e do ambiente no qual ele atua, permitindo que o robô atue de forma satisfatória sobre o seu espaço de trabalho, e fazendo com que estime seu deslocamento e faça evasão de obstáculos, por exemplo. Os sensores são compostos de transdutores, os quais transformam a energia associada a uma medida em outra forma de energia (ROMERO et al. 2014).

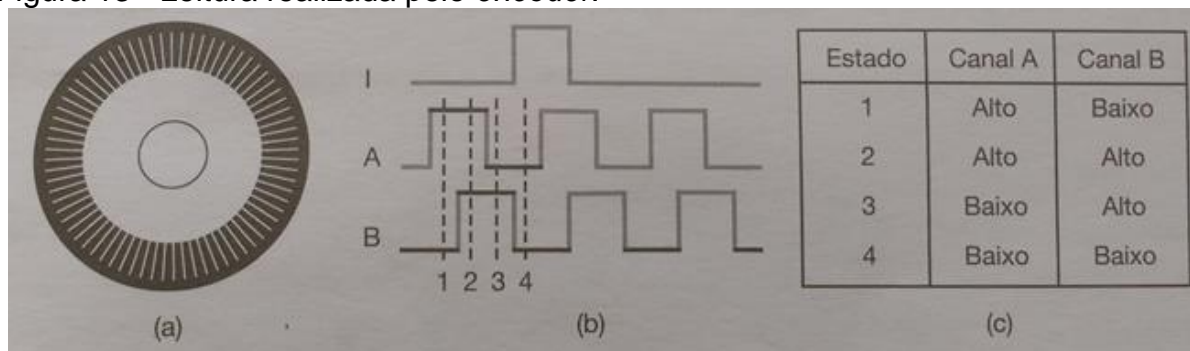
### 2.4.1 Encoders

Os *encoders*, também denominados codificadores ou odômetros óticos, são extremamente aplicados na área de robótica móvel. Eles servem para realizar a leitura de pulsos no giro das rodas do robô, a qual é muito útil para o controle da posição e velocidade das rodas. Na medida em que o robô móvel usa somente a odometria para determinar sua localização no ambiente, diz-se que ele usa reconhecimento passivo (*Dead reckoning*). O termo *Dead reckoning* surgiu da aviação, onde, para determinar a localização da aeronave, o piloto usava uma

bússola, a velocidade da aeronave e o tempo de voo. Esta técnica é bastante útil em pequenos deslocamentos, porém, ela fica comprometida em longos deslocamentos, devido a vários fatores como derrapagens, modelagem errada e etc. (ROMERO et. al., 2014).

O *encoder*, geralmente, é acoplado ao eixo da roda do robô para a contagem do seu giro. Ele é constituído de componentes como fotoemissor e fotorreceptor, e um disco com padrões de perfuração, permitindo a contagem dos pulsos de emissão de luz periodicamente pelos furos contidos no disco, sendo que a luz transmitida através do disco é transformada em uma onda quadrada associada à revolução do eixo, conforme mostra a figura 13 (ROMERO et. al., 2014).

Figura 13 - Leitura realizada pelo *encoder*.



Fonte: Romero et. al. (2014).

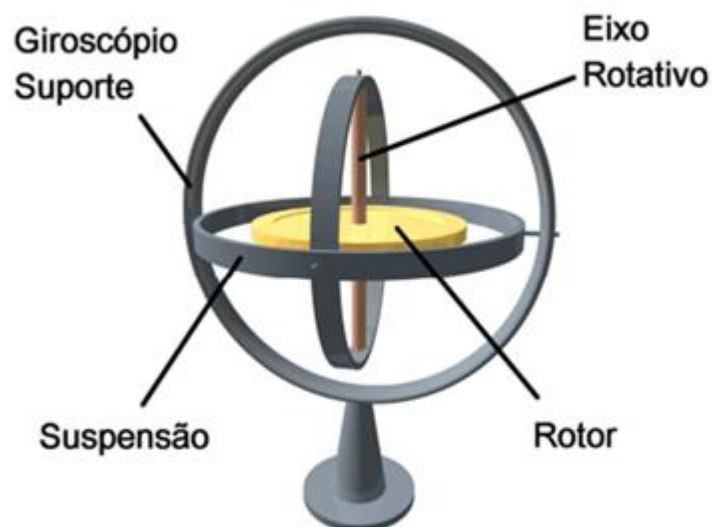
Como foram utilizadas duas plataformas robóticas diferentes (ver seção 3.3 e 3.5), também foram utilizados modelos de *encoders* diferentes. No *StarterKit* 2.0 foi utilizado o *encoder* incremental de quadratura, com 400 pulsos por revolução, e com alimentação de 5 V (IBARRA-MANZANO et al., 2012). Já a cadeira elétrica SX20 que foi instrumentada para o desenvolvimento desse trabalho, o *encoder* utilizado foi o *encoder* série 75R da marca Hohner<sup>TM</sup>. O *datasheet* (localizado em [www.hohner.com.br](http://www.hohner.com.br)) desse *encoder* informa que este fornece 10000 pulsos por revolução, é de quadratura e sua alimentação pode ser de 5 a 28 V.

## 2.4.2 Giroscópio

Através da informação proporcionada pelo *encoder* e com a utilização da odometria o robô consegue estimar sua localização ao longo do tempo, uma vez que sua localização inicial é conhecida. No entanto, a odometria é sensível a erros, principalmente aos erros cumulativos. Devido a isso, é comum a utilização de outros sensores que ajudam a amenizar alguns problemas encontrados na odometria. Assim, como a odometria causa erro de pose ilimitado, o eixo de orientação do giroscópio, onde a integração da velocidade angular fornece o ângulo percorrido em *theta* (ISHLINSKII, 1976). Contudo, o giroscópio apresenta *bias*, causando erros de leitura.

O giroscópio utilizado foi o que está presente na IMU (imu01b serie 0j3865 da marca Pololu™), o qual contém um *gyro* de 3 eixos L3G4200D. Uma interface feita em I<sup>2</sup>C acessa nove medições independentes de rotação, aceleração e magnéticas que podem ser usadas para calcular a orientação absoluta do sensor (POLOLU, 2017). Um exemplo de giroscópio mecânico é mostrado na figura 14.

Figura 14 - Exemplo de giroscópio mecânico.



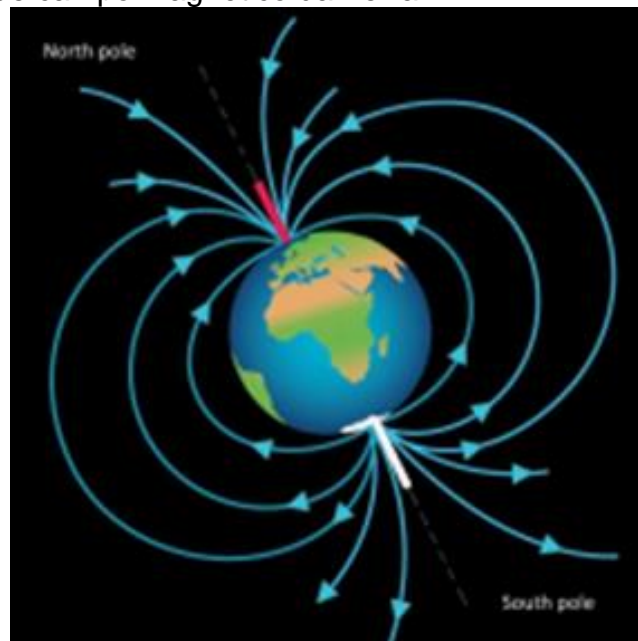
Fonte: Clark et al. (1865) adaptado.

### 2.4.3 Magnetômetro

Os magnetômetros são empregados em inúmeras aplicações de navegação. O seu emprego habitual é para determinação da orientação relativamente ao norte magnético. Contudo, campos magnéticos inseridos no mesmo ambiente destes sensores ocasionam perturbações que se somam ao vetor do campo magnético terrestre, fornecendo grandes erros nas leituras em relação aos valores reais (GOIS, 2015).

O magnetômetro utilizado possui 3 eixos (LSM303DLH), e está integrado em uma pequena placa de 0,9 "x 0,6". Utilizando uma interface I<sup>2</sup>C, pode-se realizar leituras magnéticas que podem ser usadas para calcular a orientação absoluta do sensor (POLOLU, 2017). O magnetômetro realiza a leitura do campo magnético do planeta Terra, o qual pode ser visto como um magneto gigante, com linhas de campo sobre a superfície terrestre, saído do polo sul até o polo norte, conforme mostra a figura 15.

Figura 15 - Linhas de campo magnético da Terra

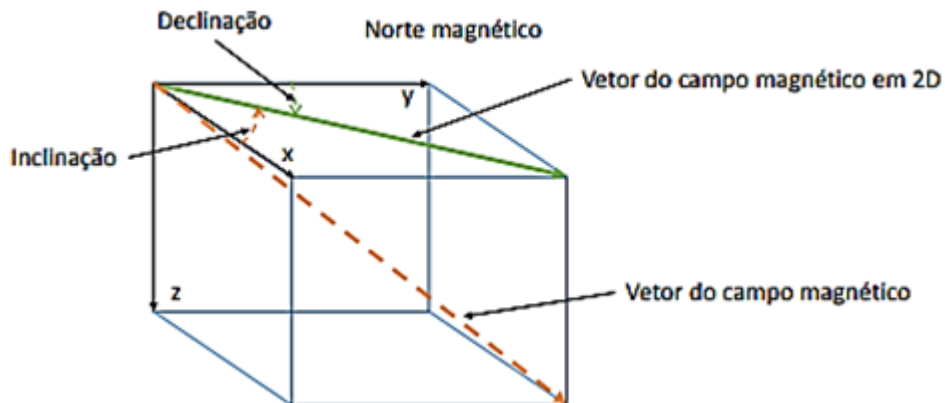


Fonte: Gois (2015).



Os polos geográficos da Terra não são paralelos aos polos magnéticos, na verdade, existe um ângulo (*offset*) entre o eixo de rotação da terra e o eixo formado pelos dois polos. Este ângulo designa-se por declinação, conforme mostra a figura 16.

Figura 16 - Declinação e inclinação magnética



Fonte: Gois (2015).

O ângulo de declinação não possui valor fixo, (apesar de isso ser considerado), já que, na verdade, o ângulo muda vagarosamente (GOIS, 2015). Além da declinação, existe o ângulo que o vetor do campo magnético forma com a superfície da terra. Este ângulo designa-se inclinação, que também é não linear (KONVALIN, 2008).

### 3 MATERIAIS E MÉTODOS

A seção Material e Métodos é um plano detalhado sobre tudo o que será realizado na sua pesquisa. Com isso, esta seção deve conter um nível de detalhamento, de forma tal que se o projeto for apresentado a outro pesquisador, esse pesquisador deva conseguir executar a pesquisa exatamente da mesma forma que o autor da pesquisa executaria (VIANNA, 2001).

Nesta seção serão mostradas as plataformas robóticas utilizadas para realização dos experimentos e as configurações utilizadas, o método de processamento de imagem utilizado para validação da localização, os métodos utilizados para a calibração dos sensores utilizados e a montagem do EKF aplicado ao robô. Com os conhecimentos mostrados espera-se que outro pesquisador que siga os passos demonstrados consiga repetir os resultados.

#### 3.1 DESCRIÇÃO DA PLATAFORMA ROBÓTICA

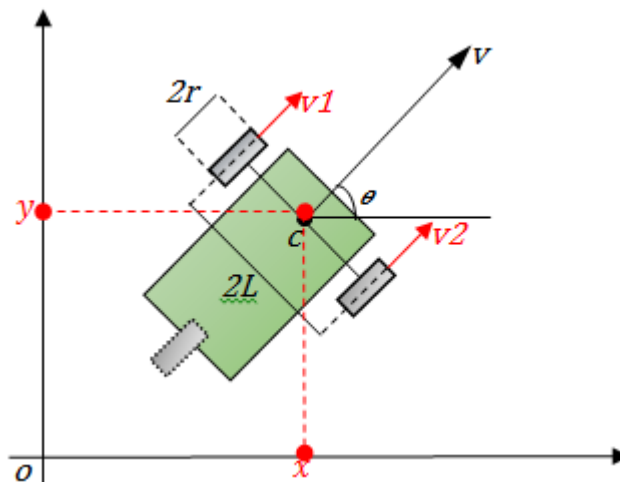
Nesta seção são descritas as características das plataformas robóticas utilizadas que formam o robô *Starter kit 2.0* e a cadeira de rodas elétrica (*Styles SX20*), a qual foi instrumentada, conforme a seção 3.5. Os robôs móveis utilizados para a realização deste trabalho são de tração diferencial e não-holonômicos.

Primeiramente, o *Starter kit 2.0* foi utilizado por ser mais didático, sendo mais vantajoso no aprendizado dos métodos a serem utilizados. Após a implementação dos métodos, a cadeira de rodas robótica começou a ser utilizada, pois ela pertence a um projeto futuro do Gain/Ifes, sendo esta Dissertação de Mestrado uma parte do projeto final de uma cadeira de rodas autônoma.

### 3.1.1 Plataformas robóticas usadas

O *Starter kit 2.0* possui duas rodas paralelas de tração independente uma da outra posicionadas uma em cada lateral do robô, além de uma roda livre (roda boba ou *caster*), posicionada na parte traseira do robô. As duas rodas motoras são semelhantes, com raio  $r=5,09$  cm, e distância  $2L=36,5$  cm do eixo que liga uma roda à outra. O *Starter kit 2.0* possui a configuração mostrada na figura 17.

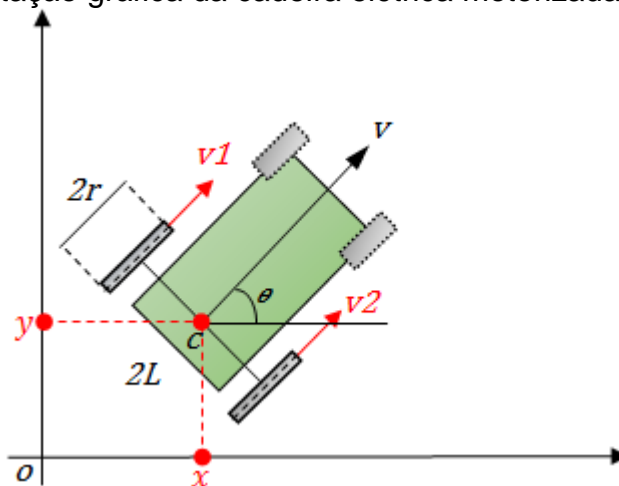
Figura 17 - Representação gráfica do *Starter kit 2.0*



Fonte: Tommasi et al. (2015) adaptado.

A cadeira de rodas elétrica *Styler SX20* possui duas rodas motoras paralelas de tração separadas uma da outra posicionadas uma em cada lateral e duas rodas livres (roda boba ou *caster*) posicionadas na parte dianteira. As duas rodas motoras são semelhantes, com raio  $r=24$  cm, separadas por uma distância igual a  $2L=58$  cm. A cadeira de rodas elétrica possui a configuração mostrada na figura 18.

Figura 1 - Representação gráfica da cadeira elétrica motorizada *Styles SX20*

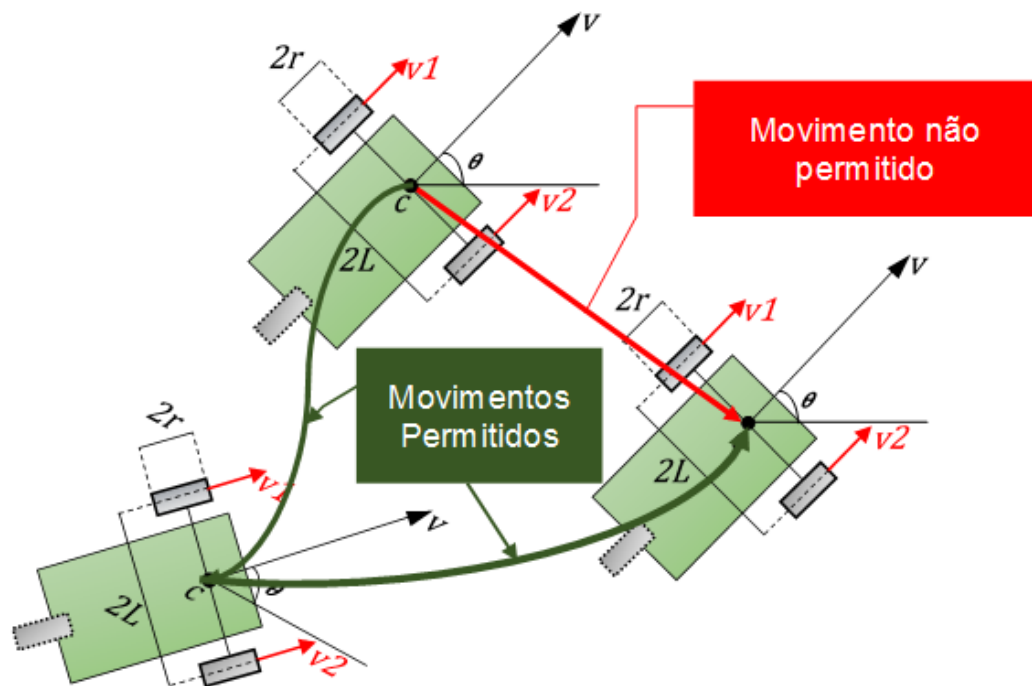


Fonte: Próprio autor (2017).

As poses das plataformas citadas são definidas pelas coordenadas  $q=[x, y, \theta]^T$  cujos parâmetros  $x$  e  $y$  demarcam a sua posição no eixo cartesiano, e  $\theta$  determina o ângulo de orientação, conforme mostrado nas figuras 17 e 18. O centro de massa está localizado no ponto  $C$ , que fica no centro do eixo das rodas. O ponto  $C$  será utilizado como ponto de referência, ou ponto de interesse, para o controle de cada uma das plataformas robóticas.

Caso as velocidades das duas rodas sejam idênticas ( $v1=v2$ ), o deslocamento será em linha reta; se a velocidade da roda direita ( $v2$ ) for menor do que a roda esquerda ( $v1$ ), o movimento ocorrerá de maneira circular para a direita; caso a velocidade da roda esquerda  $v1$  seja menor, o movimento ocorrerá de maneira circular para a esquerda. Apesar disso, as plataformas robóticas expostas não podem efetuar cursos transversais, por isso é preciso executar uma sequência de manobras para efetuar movimentos mais complexos. Este tipo de robô, com limitações de deslocamento lateral é chamado de não holonômico. Na figura 19 é mostrado um exemplo de restrições holonômicas relacionadas ao movimento do robô móvel, onde ele não consegue realizar apenas um movimento para realizar uma mudança de localização (movimento de translação), com isso, realizam-se vários movimentos (de translação e de rotação) para atingir o objetivo final.

Figura 19 - Representação gráfica dos movimentos aceitos pelo robô móvel não-holonômico



Fonte: Próprio autor (2017).

Na tabela 6 são mostradas as medidas dos raios das rodas, o tamanho do eixo entre as rodas e as velocidades máximas das plataformas robóticas utilizadas. Como as velocidades angulares máximas aplicadas pelo motor a cada roda é igual, ou seja,  $w1_{max}=w2_{max}$ , é possível inferir que as velocidades lineares máximas de cada roda também são iguais, ou seja,  $v1_{max}=v2_{max}$ . Para efeito dos experimentos práticos as assimetrias das plataformas são desconsideradas.

Tabela 6 - Velocidades máximas aplicadas as plataformas robóticas.

Plataformas	raio (cm)	Eixo entre rodas (cm)	$w1_{max}=w2_{max}$ (m/s)	$v1_{max}=v2_{max}$ (rad/s)
<b>Starter kit 2.0</b>	5,09	36,50	0,80	15,70
<b>Styles SX20</b>	24,00	58,00	1,94	8,10

Fonte: Próprio autor (2017).

Em (44) é mostrado o modelo cinemático do robô móvel. Esse modelo é responsável por realizar as sequências de estados (poses assumidas pelo robô no tempo),

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 \\ \sin\theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ w(t) \end{bmatrix} \quad (44)$$

onde,  $v$  e  $w$  são as velocidades linear e angular do robô, respectivamente. Destaca-se que em tal modelo considera-se como a localização do robô o ponto de interesse C, conforme as figuras 17 e 18.

Na literatura podem ser encontradas outras formas de representar este modelo, como em (45), (46), (47) e (48), as mesmas relacionadas ao movimento do robô em função da a velocidade angular e velocidade linear do robô móvel,

$$v_1 = w_1 r, \quad (45)$$

$$v_2 = w_2 r, \quad (46)$$

$$v = \frac{v_1 + v_2}{2}, \quad (47)$$

$$w = \frac{v_2 - v_1}{L}, \quad (48)$$

onde,  $w_1$  e  $w_2$  são as velocidades angulares das rodas direita e esquerda, respectivamente, e  $v_1$  e  $v_2$  são as velocidades lineares das rodas direita e esquerda, respectivamente.

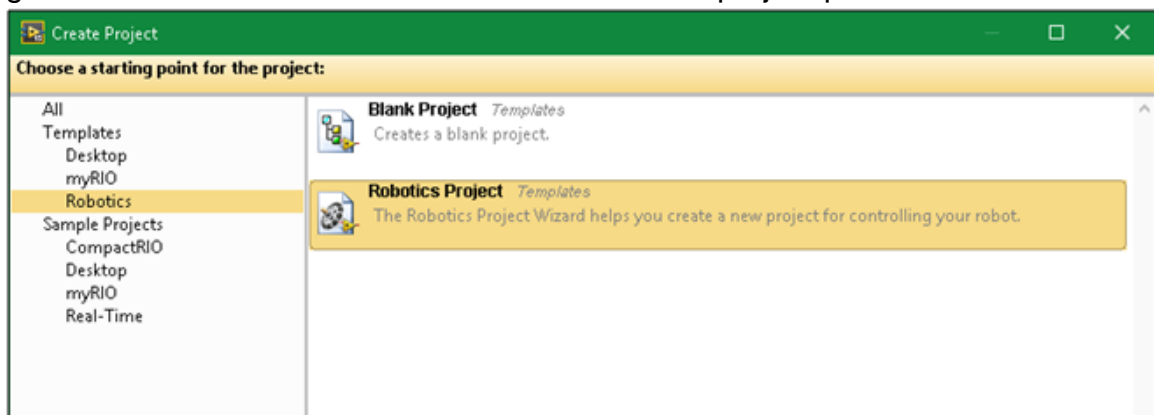
### 3.1.2 Ambiente de simulação do Starter KIT 2.0

A simulação de robôs móveis é uma necessidade comum para o desenvolvimento de aplicações. A simulação permite que os desenvolvedores das aplicações

realizem testes no código muito mais rápido do que se usassem hardware real. O módulo LabVIEW™ *Robotics* é um pacote de software que possibilita o desenvolvimento e simulação de aplicações robóticas. Este módulo da *National Instruments* permite que se possam realizar simulações, validações e testes de algoritmos como mapeamento, navegação e planejamento de trajeto e, em seguida, transferir o mesmo código para o *Starter Kit 2.0* (NATIONAL INSTRUMENTS, 2017).

Na figura 20 é mostrada a tela que cria o programa LabVIEW™, o qual permite escolher o tipo de projeto desejado. No caso da criação do projeto de simulação, foi escolhida a opção *Robotics*.

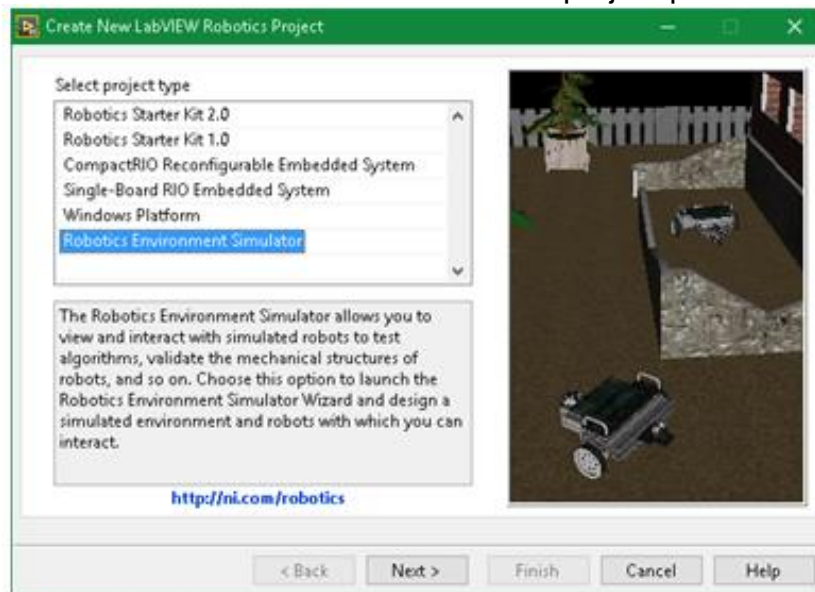
Figura 20 - Simulador do Starterkit 2.0 - Escolha do projeto para Robotics.



Fonte: Próprio autor (2017).

Após criar o projeto *Robotics*, é necessário a escolha da plataforma robótica que será utilizada no *Robotics Environment Simulator*, como mostra a figura 21.

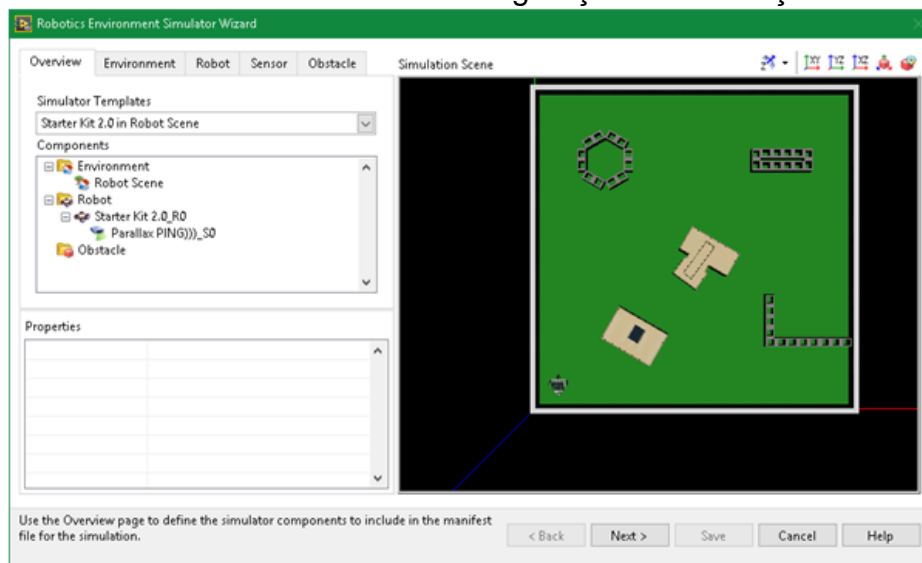
Figura 21 - Simulador do Starter kit 2.0 - Escolha do projeto para simulação



Fonte: Próprio autor (2017).

Após a criação da simulação, é necessário realizar a configuração em relação a: ambiente, robô usado, sensores a serem utilizados e os obstáculos. Neste momento, deve-se criar o ambiente de simulação o mais fiel possível ao ambiente real, de forma que a transição da aplicação deste ambiente para a etapa real seja com o mínimo de esforço possível, conforme mostra a figura 22.

Figura 22 - Simulador do Starterkit 2.0 - Configuração da simulação

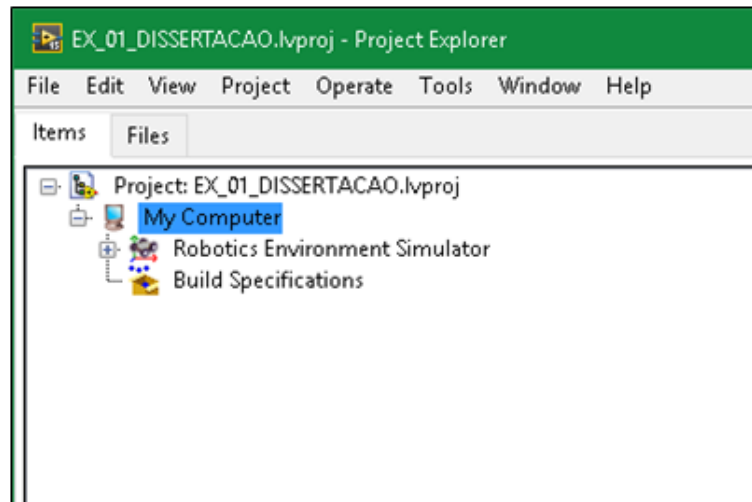


Fonte: Próprio autor (2017).



O resultado final é a criação do projeto, conforme mostra a figura 23.

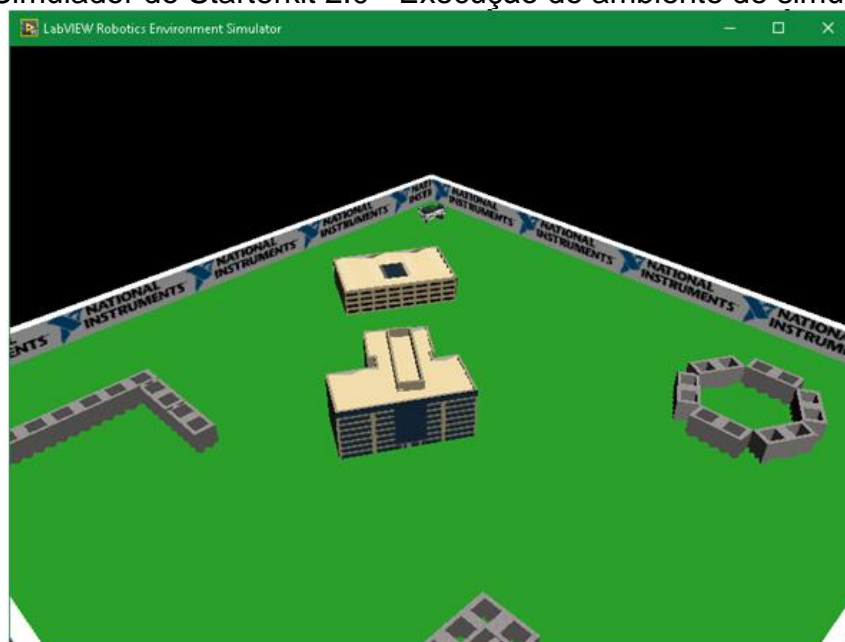
Figura 23 - Simulador do Starterkit 2.0 - Projeto criado.



Fonte: Próprio autor (2017).

Ao executar o projeto, pode-se verificar o ambiente virtual de simulação, conforme mostrado na figura 24.

Figura 24 - Simulador do Starterkit 2.0 - Execução do ambiente de simulação



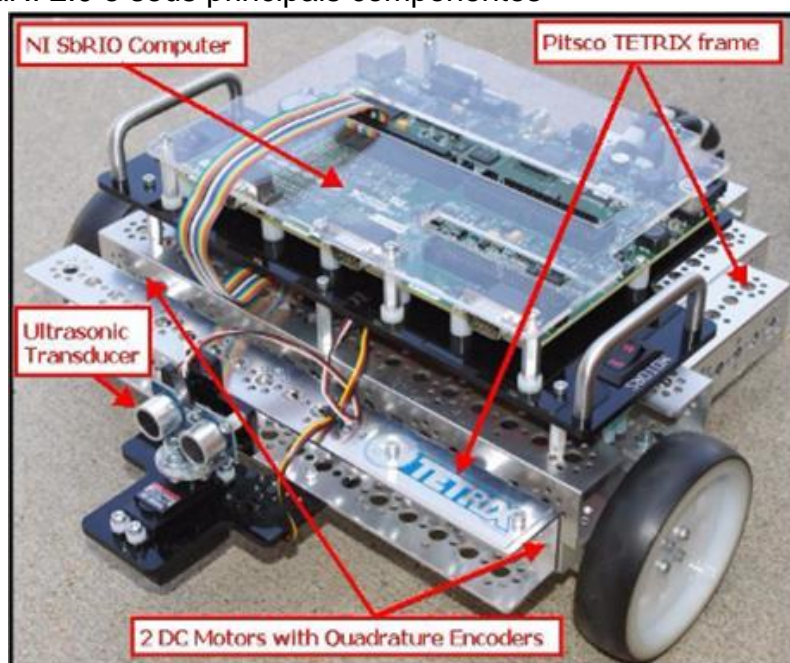
Fonte: Próprio autor (2017).

### 3.1.3 Starter KIT 2.0

A plataforma robótica NI LabVIEW *Robotics Starter Kit*<sup>TM</sup> na versão 2.0 (DaNi) é uma plataforma comercializada pela *National Instruments*<sup>TM</sup> para ensino, pesquisa e prototipagem. A plataforma é um *Kit* base de robótica móvel totalmente montado com sensor, codificadores, motores, bateria, carregador e sensores de ultrassom incluídos. O seu controlador é baseado em NI Single-Board RIO, tem tomada de decisão em tempo real e processamento de I/O baseado em FPGA. Vem com os módulos do LabVIEW *Robotics*<sup>TM</sup>, LabVIEW *Real-Time*<sup>TM</sup> e *software* do módulo LabVIEW FPGA<sup>TM</sup>. O robô permite se conectar facilmente com uma variedade de sensores e atuadores.

O DaNi, como é popularmente chamado, é um robô móvel, que contém sensores, motores e um componente NI 9632 *Single-Board Reconfigurable I/O*<sup>TM</sup> (sbRIO) montado sobre uma estrutura Pitsco TETRIX<sup>TM</sup>. O kit é produzido por PITSCO *Education*<sup>TM</sup>, que produz *kits*, guias de ensino e ferramentas educativas, como mostra a figura 25.

Figura 25 - DaNi 2.0 e seus principais componentes

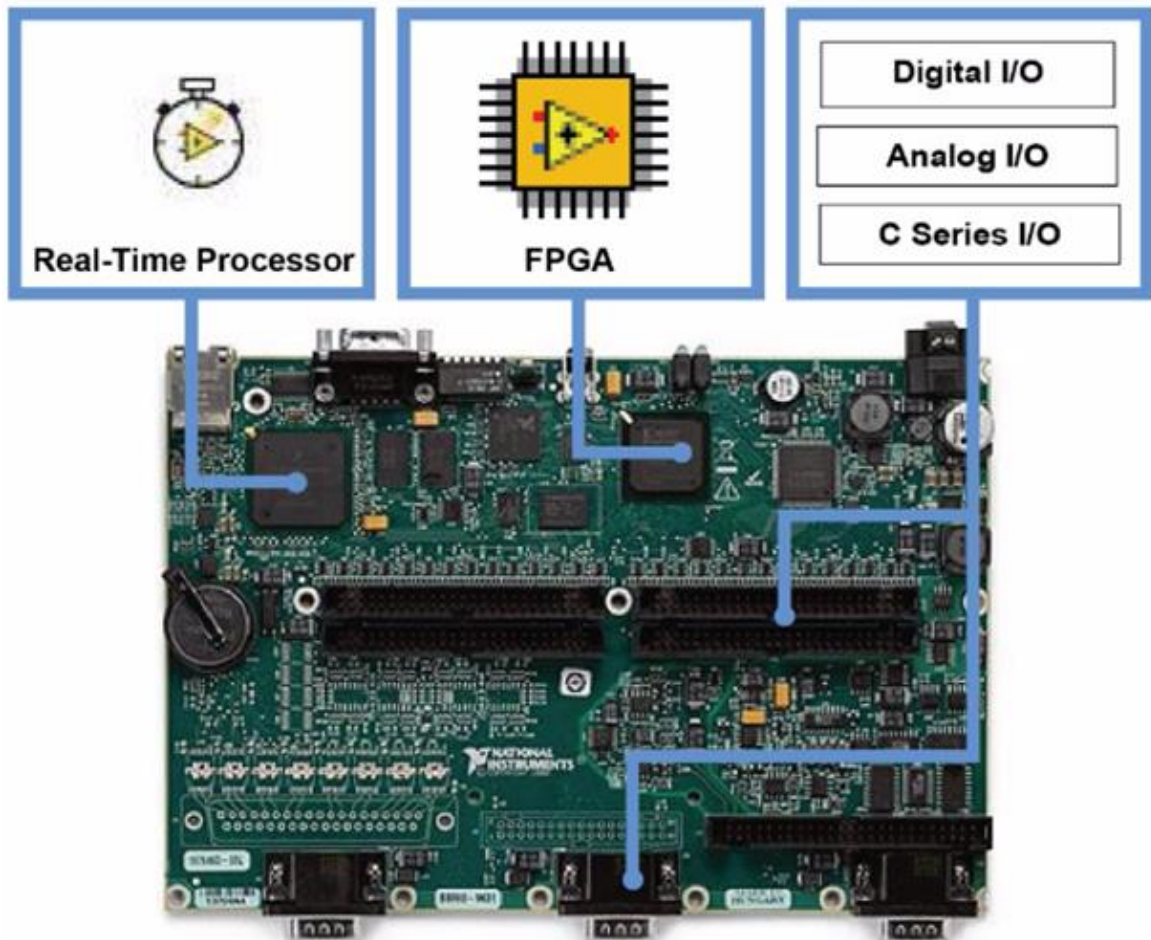


Fonte: King (2011).

O NI *Single-Board RIO*<sup>™</sup>, mostrado na figura 26, é um desenvolvimento de uma plataforma integrada com processador em tempo real FPGA (*reconfigurable field-programmable gate array*) e entradas e saídas analógicas e digitais. Esta placa é programada com o LabVIEW *Real-Time*<sup>™</sup>, LabVIEW *FPGA*<sup>™</sup> e LabVIEW *Robotics*<sup>™</sup>.

O *starter kit* 2.0 inclui sensor de ultrassom e *encoders*, contudo, a plataforma também aceita expansão para uma variedade grande de sensores, tais como: LIDAR, radar, infravermelho, bússola, giroscópio, IMU (*Inertial Measurement Unit*), GPS (*Global Position System*), câmera (CCD e CMOS), etc (KING, 2011).

Figura 26 - Placa 9632 NI *Single-Board RIO*<sup>™</sup> com processador de tempo real, FPGA e I/Os (digital e analógicas)



Fonte: King (2011).

### 3.1.4 Estimativa de localização por processamento de imagens

O processamento de imagem é utilizado para se obter a estimativa de pose a mais próxima do real quanto possível para realizar o comparativo de três controladores de trajetória. Com a utilização exclusiva da odometria feita por Tommasi et al. (2015), se conclui que apesar da odometria sempre mostrar a pose do robô, ela não considera os erros acumulados, fazendo com que a trajetória real seguida pelo robô não corresponde, com a que foi estimada, pois é visível que existem deslocamentos da trajetória tanto de forma rotacional, quanto translacional.

Para obter a pose do robô a validação da posição foi necessário utilizar um método no qual o erro não se acumulasse com a distância percorrida e que fosse mais preciso que a odometria (*Dead Reckoning*). Com isso, para se obter menores erros na estimativa de pose do robô, e utilizar essas informações para a validação, foi usado o processamento de imagens para realizar esta medição (MOTTA et al., 2017).

Foi instalada uma câmera no teto do laboratório de testes, a qual é ligada diretamente ao computador para realizar a estimativa de pose. Além disso, uma placa contendo dois círculos de diferentes tamanhos foi colocada sobre o robô. Pode-se ver o laboratório montado para a execução dos experimentos na figura 27, o qual é constituído de uma lona emborrachada e uma câmera ligada diretamente ao computador e posicionada no teto. A lona emborrachada serve para evitar o deslizamento das rodas do robô.

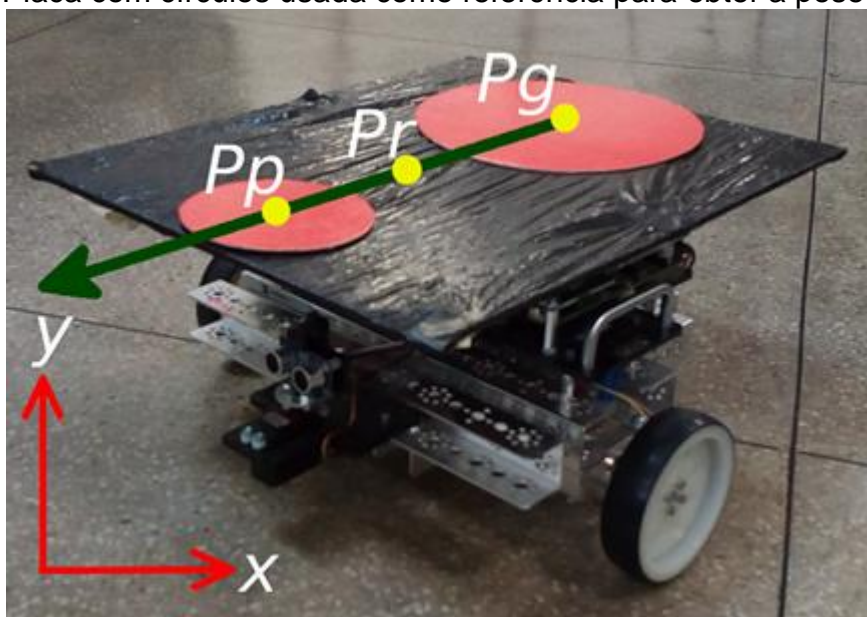
Figura 27 - Ambiente de testes



Fonte: Próprio autor (2017).

A centroide do círculo menor é representada por  $P_p$ , a centroide do círculo maior é representado por  $P_g$ , e a centroide do robô é representada pelo ponto  $P_r$ . A pose do robô é representada por uma flecha verde, conforme mostra a figura 28.

Figura 28 - Placa com círculos usada como referência para obter a pose do robô



Fonte: Próprio autor (2017).

As imagens adquiridas são obtidas pela câmera, configurada de forma a melhor capturar a cor dos círculos localizados na placa posicionada acima do robô, para realizar a etapa de processamento de imagem. O processamento de imagem tem a finalidade adquirir os dados de interesse, como a área e pontos referentes ao centro de massa dos discos. Através dos dados adquiridos é possível estimar o erro de localização do robô em relação à trajetória de referência.

A placa com dois círculos foi utilizada para efeito de referência de postura do robô. Ela foi posicionada acima do robô de forma que o círculo menor ficasse localizado na parte da frente, e o círculo maior localizado na traseira. A referência dos pontos  $x$  e  $y$  nos dois círculos é tomada do ponto de seu centroide. Com isso, as variáveis  $x_p$  e  $y_p$  são referentes ao ponto centroide do círculo pequeno, e  $x_g$  e  $y_g$  são referentes ao ponto centroide do círculo grande. O ponto centroide do robô é representado pelas variáveis  $x_r$  e  $y_r$ , as quais são calculadas como o ponto médio do segmento de reta dos pontos definidos por  $(x_p, y_p)$  e  $(x_g, y_g)$ , conforme (49) e (50),

$$x_r = (x_g + x_p)/2, \quad (49)$$

$$y_r = (y_g + y_p)/2. \quad (50)$$

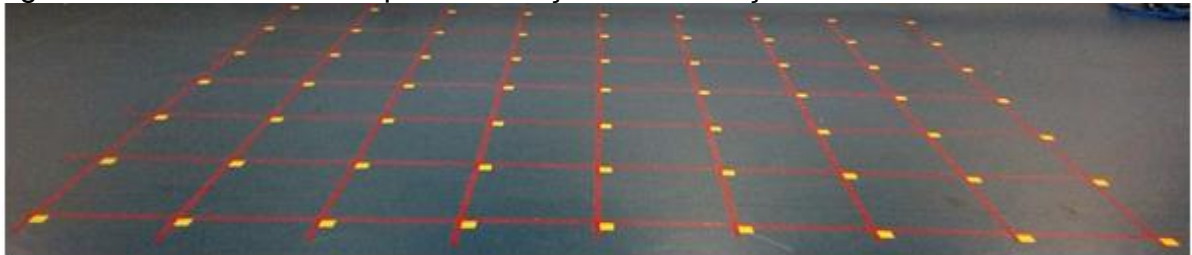
Para o cálculo da orientação do robô ( $\theta$ ) foi utilizada a função inversa do arco tangente, que recebe o valor da tangente e retorna o ângulo, em radianos, utilizado na medida, tendo como entrada a diferença das variáveis  $x$  e  $y$ , ou seja,  $\Delta x$  e  $\Delta y$ .

O processamento de imagem foi implementado com o intuito de estimar a pose do robô e, com isso, comparar a pose conseguida com a câmera com a odometria e as duas fusões. A ferramenta utilizada para o tratamento da imagem foi o módulo *Vision Development*, que é um módulo do programa LabView<sup>TM</sup> com uso próprio para tratamento de imagens.

Para conseguir estimar a pose do robô através de processamento de imagens, é necessário vincular as dimensões das imagens adquiridas com as dimensões do mundo real, sendo essa etapa chamada de calibração. A calibração tem como

objetivo extrair e identificar as características específicas na imagem. Para que fosse mais fácil este processo, foi construído um *grid*, mostrado na figura 29, que é um retângulo que mede 4 m por 3 m, e foram demarcadas as distâncias de cada um dos quadrados do *grid*. Os quadrados colocados no chão foram espaçados de 50 cm entre si. O ponto central do *grid* também foi demarcado. O método utilizado para a calibração é conhecido como calibração automática em *grid*. Utilizando essa calibração é possível fazer a correspondência entre as coordenadas localizadas na imagem e as coordenadas do mundo real, ou seja, mapear os *pixels* da imagem em metros no mundo real. (NASCIMENTO, 2014).

Figura 29 - Grid construído para facilitação da calibração



Fonte: Próprio autor (2017).

Para o reconhecimento dos círculos foi utilizado o método de limiar de cor, o qual é usado para remover as cores da imagem que não seja de interesse. Este método pode ser utilizado para detectar objetos com valor de cor, destacando-se os círculos de interesse ao processamento de imagem. Para realizar este procedimento foi utilizado um padrão de cor chamado HSL, o qual utiliza os conceitos de matiz (*hue* - H), pureza de cor (*saturation* - S) e luminosidade (*Lightness* - L), sendo esses parâmetros são ajustados para encontrar a cor desejada. Este sistema oferece uma descrição muito precisa das cores, dando suporte à sua difusão (PORTELA, 2014).

Para melhorar a forma dos círculos, foram realizadas transformações morfológicas que modificam a forma dos objetos nas imagens binárias. Esses procedimentos fazem a expansão, preenchimento de “buracos” e a retirada de ruídos. O filtro de partículas extrai as partículas restantes que sejam indesejadas. Ao final, os dados das partículas são adquiridos, tais como área, centroide, entre outros. Estes dados

podem ser retornados tanto como valores das imagens (*pixels*) ou como medidas do mundo real (centímetros) para dentro do programa.

A placa com dois círculos foi posicionada sobre a cadeira robótica, com o círculo menor localizado à frente. Com isso, foi necessária a desmontagem do encosto da cadeira de rodas, conforme mostra a figura 30.

Figura 30 - Posicionamento da placa de localização posicionada na cadeira de rodas robótica

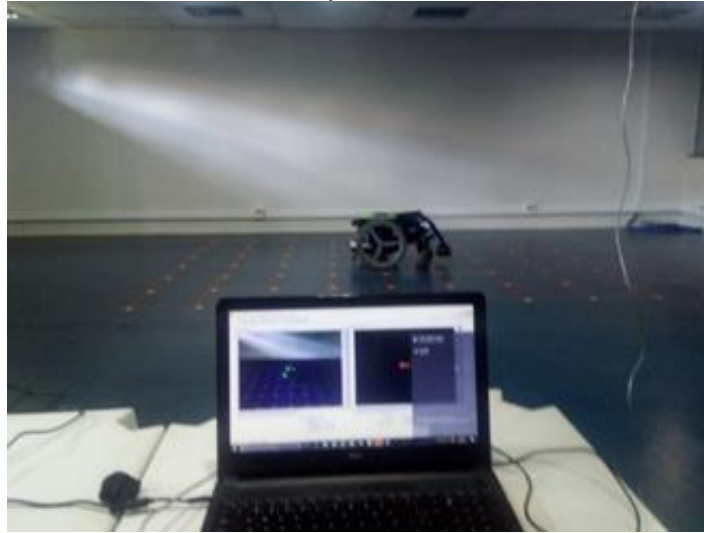


Fonte: Próprio autor (2017).

Uma etapa de processamento de imagem é explicada na seção 3.4. A partir do processamento de imagem, o programa calcula a posição do centroide dos círculos pequeno e grande, com coordenadas  $(x'_p, y'_p)$  e  $(x'_g, y'_g)$ , respectivamente, como mostrado na figura 31.



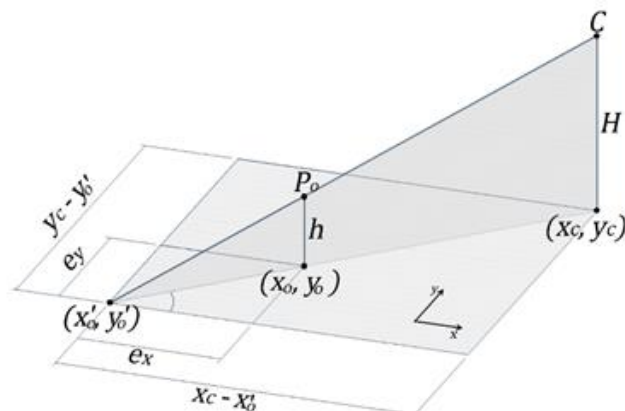
Figura 31 - Ambiente de testes utilizado para se obter os resultados



Fonte: Próprio autor (2017).

As coordenadas obtidas com o processamento de imagem não descrevem a real posição dos círculos devido ao erro de paralaxe produzido pela câmera, o qual é o erro de projeção relacionada ao posicionamento do observador (KARLSTROEM, 2007). Como a câmera de captura das imagens fica posicionada no teto do ambiente de testes, cria-se um ângulo para a captura das imagens dos círculos sobre a placa de localização. Esse ângulo de projeção deve ser minimizado para que a leitura seja a mais correta possível, sendo então realizado um ajuste através de equações matemáticas, conforme mostra Salarolli, Motta e Cuadros (2017). O referido erro de leitura realizado pela câmera é mostrado na figura 32.

Figura 32 - Leitura da posição do robô com câmera



Fonte: Salarolli; Motta; Cuadros (2017).

Como pode-se observar na figura 32,  $H$  corresponde à altura da câmera em relação ao chão,  $h$  é a altura da cadeira,  $(x_o, y_o)$  corresponde a localização real da cadeira,  $(x_c, y_c)$  corresponde a localização da câmera,  $e_x$  e  $e_y$  são os erros de medição em  $x$  e  $y$ , respectivamente, e  $(x'_o, y'_o)$  são as coordenadas da cadeira lidas, com erro, através do processamento da imagem. Utilizando-se das regras da geometria, chega-se a (51) e (52), de forma a corrigir o erro na medição e obter a coordenada real do objeto  $(x_o, y_o)$ ,

$$\begin{cases} x_o = x'_o - e_x \\ y_o = y'_o - e_y \end{cases} \quad (51)$$

$$\begin{cases} e_x = \frac{h}{H} \sqrt{(x_c - x'_o)^2 + (y_c - y'_o)^2} \cos \left[ \tan^{-1} \left( \frac{y_c - y'_o}{x_c - x'_o} \right) \right] \\ e_y = \frac{h}{H} \sqrt{(x_c - x'_o)^2 + (y_c - y'_o)^2} \sin \left[ \tan^{-1} \left( \frac{y_c - y'_o}{x_c - x'_o} \right) \right] \end{cases} \quad (52)$$

então, (51) e (52) foram utilizadas para  $(x'_p, y'_p)$  e  $(x'_g, y'_g)$ , assim, obtendo-se  $(x_p, y_p)$  e  $(x_g, y_g)$ , os quais são as posições reais do centroide dos círculos pequeno e grande, respectivamente. As coordenadas do centroide do robô  $(x_r, y_r)$  juntamente com sua orientação, formam a postura do robô, que pode ser calculada conforme mostrado em (53).

$$\begin{bmatrix} x_r \\ y_r \\ \theta_r \end{bmatrix} = \begin{bmatrix} (x_p + x_g) / 2 \\ (y_p + y_g) / 2 \\ \text{atan2} \left( (y_p - y_g), (x_p - x_g) \right) \end{bmatrix} \quad (53)$$

Com o intuito de realizar a verificação sobre a validade do método mostrado, a cadeira de rodas robótica foi posicionada, de forma manual, em 28 pontos distintos. Com isso, foram obtidos os erros absolutos máximos e médios da medição realizada pela câmera, os quais são mostrados na figura 32. A distância entre o ponto representando a leitura da câmera corrigida para o ponto da leitura com erro é dada por  $D$ .

Tabela 7 - Dados das variáveis utilizados em ambos os programas de simulação.

Variável	Erro médio absoluto	Erro máximo absoluto
$x$	0,4 ( $10^{-2}$ m)	1,2 ( $10^{-2}$ m)
$y$	0,5 ( $10^{-2}$ m)	1,6 ( $10^{-2}$ m)
$\theta$	0,01 (rad)	0,04 (rad)
$D$	0,7 ( $10^{-2}$ m)	1,9 ( $10^{-2}$ m)

Fonte: Salarolli; Motta; Cuadros (2017).

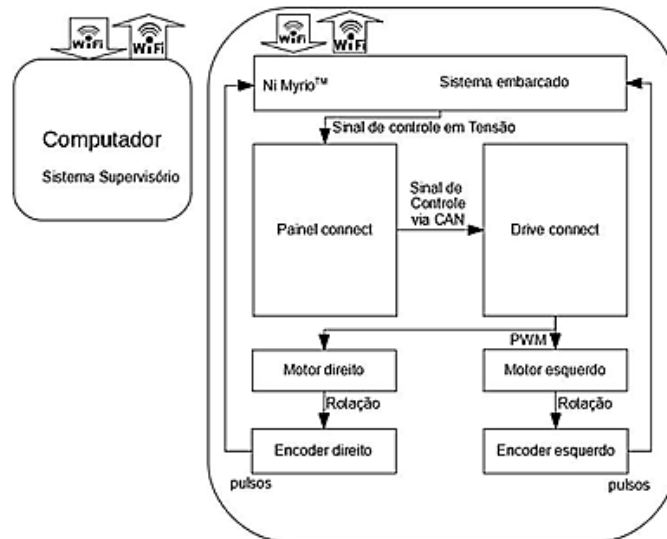
Como pode ser observado na tabela 7, os erros nas variáveis  $x$ ,  $y$ ,  $\theta$  e  $D$  são muito pequenos em relação às dimensões do robô e às dimensões da área de testes, podendo assim ser desprezados. Esta observação permite assumir que a postura lida com o processamento de imagens é a postura real do robô.

### 3.1.5 Cadeira de rodas robótica

Nesta seção é descrito como a cadeira de rodas elétrica foi transformada em uma cadeira de rodas robóticas. Um computador realiza a conexão com o sistema de controle de cadeira de rodas robótica via rede sem fio (WiFi) para rodar o sistema supervisor que recebe e envia informações do sistema embarcado na placa NI myRio™. A placa NI myRio™ envia o sinal de controle para o módulo painel *connect* utilizando a eletrônica nativa da cadeira, e o painel *connect* envia os valores dos sinais de controle para o módulo *drive connect*, via rede CAN (*Controller Area Network*).

O módulo *drive connect* envia o sinal de controle para os motores, de modo a movimentar as rodas da cadeira. Com o acionamento dos motores, a correia entra em movimento, fazendo com que os *encoders* girem produzindo os pulsos. Os pulsos retornam ao sistema embarcado com a intenção de realizar os cálculos da odometria. Essa iteração é mostrada no diagrama de blocos e está representada na figura 33.

Figura 33 - Diagrama de blocos da cadeira de rodas robótica



Fonte: Motta et al. (2017).

A cadeira elétrica *Styles SX20* tem a cor azul, largura de 40 cm, capacidade de bateria de 55 Ah e é fabricada pela empresa *Freedom®*. Esta cadeira foi escolhida de forma a manter a viabilidade do projeto, já que buscou-se obter um produto visando mínimo preço e que ao mesmo tempo tivesse a qualidade desejada para prover as características que se adequassem ao projeto. O produto é mostrado na figura 34.

Figura 34 - Cadeira motorizada *Styles SX20*



Fonte: Freedom (2017).

As cadeiras motorizadas da empresa *Freedom*® possuem uma tecnologia chamada *Freedom Connect*®. Esta tecnologia possui dois módulos independentes e microprocessados (FREEDOM, 2017). O módulo *driver connect* é o módulo de potência que administra as velocidades dos motores, o acionamento dos freios e o consumo e carga das baterias, sendo mostrado na figura 35.

Figura 35 - Módulo *driver connect*



Fonte: Freedom (2017).

O módulo *panel connect* é dedicado ao painel do *joystick*, o qual comporta funções de interação com o usuário e comunicação *web*, sendo mostrado na figura 36.

Figura 36 - Joystick da cadeira motorizada Styles SX20.



Fonte: Freedom (2017).

Os módulos inteligentes estão constantemente à procura de falhas; caso uma falha ocorra, um mecanismo de segurança é acionado e a cadeira é bloqueada imediatamente. As ocorrências de falhas podem ser identificadas através de um código no painel da cadeira. Todas as irregularidades são registradas nas memórias internas dos módulos e podem ser visualizadas através do *software* da *Freedom Connect*<sup>®</sup>. A comunicação dos módulos é realizada através de protocolo CAN (*Freedom Veículos Elétricos Ltda*, 2016). As características técnicas da cadeira são mostradas na tabela 8.

Tabela 8 - Características da cadeira motorizada.

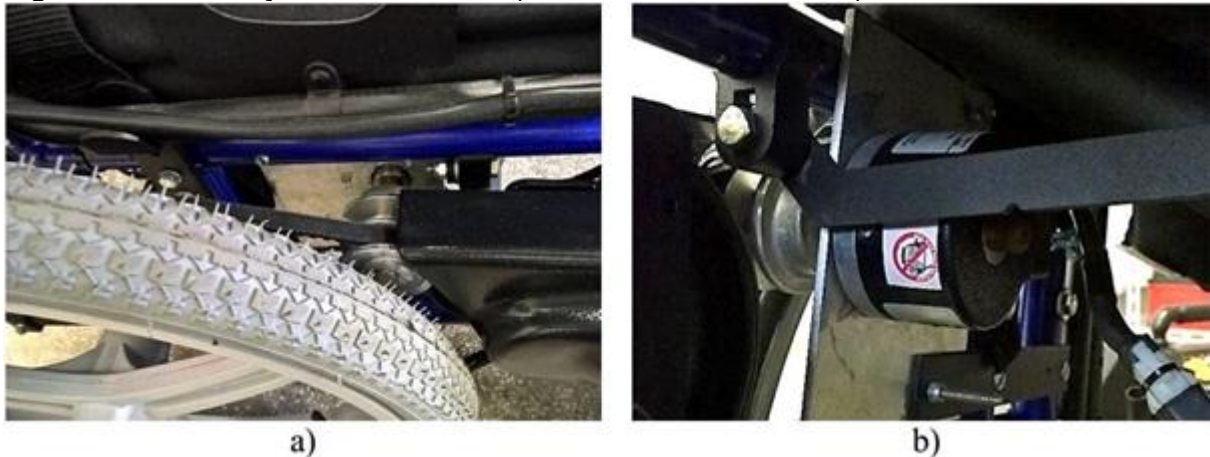
Características técnicas	<ul style="list-style-type: none"> <li>• Garfos dianteiros com suspensão</li> <li>• Chassis em alumínio</li> <li>• Estrutura de fechamento em X</li> <li>• Regulagem de <i>caster</i>, centro de gravidade e <i>tilt</i></li> <li>• Rodas traseiras 20", infláveis, aro fundido</li> <li>• Rodas dianteiras 8", infláveis, aro em nylon</li> <li>• Rodas de apoio 2 x 50 x 25mm</li> <li>• Eixos com rolamentos blindados</li> <li>• Apoio de pé em polipropileno, rebatíveis, destacáveis e com regulagem de altura</li> <li>• Apoio de braços rebatível com carenagem de proteção, facilitando o acesso do usuário</li> <li>• Assento e encosto acolchoados e revestidos em tecido nylon lavável</li> <li>• Drive e Joystick remoto microprocessados que pode ser instalado no braço lado direito ou esquerdo</li> <li>• Peso da cadeira 35,5 Kg</li> <li>• Peso da bateria 17,7 Kg</li> <li>• Peso Total 70,9 Kg</li> </ul>
Dimensões	<ul style="list-style-type: none"> <li>• Largura do assento 50cm</li> <li>• Profundidade do assento 42 cm</li> <li>• Altura do encosto 40 cm</li> <li>• Largura total Aberta: 71cm</li> </ul>
Capacidade de carga	<ul style="list-style-type: none"> <li>• Até 130 kg</li> </ul>
Motorização	<ul style="list-style-type: none"> <li>• Com dois motores elétricos de corrente contínua e imã permanente de 400W cada</li> <li>• Sistema de transmissão de baixa manutenção por correias V e Micro V</li> <li>• Sistema de freio motor regenerativo</li> <li>• Sistema de freio de estacionamento eletromagnético</li> </ul>
Autonomia	<ul style="list-style-type: none"> <li>• Autonomia ideal 20km / máxima 40km</li> <li>• Velocidade máxima de 7 km/h</li> </ul>
Alimentação	<p><b>Obs:</b> Em terreno plano e liso, uso intermitente e usuário até 80kg</p> <ul style="list-style-type: none"> <li>• Duas baterias 12V disponíveis: 55Ah</li> <li>• Carregador de baterias bivolt com saída: 24Vx4A</li> </ul>

Fonte: Freedom (2017).

A cadeira motorizada foi instrumentada sofrendo alterações para que suportasse a implementação de um projeto de uma cadeira robótica. A primeira alteração foi o posicionamento de dois *encoders* em ambas as laterais, de modo que as correias que movimentam as rodas tenham contato com o eixo do *encoder* e consigam aplicar movimento giratório, de maneira que os *encoders* possam gerar os pulsos no momento em que há giro nas rodas. Cada *encoder* possui dois canais e resolução de 5000 pulsos por volta. O *encoder* da roda direita foi ligado ao ENC0 (canal A pino 13, canal B no pino 11) do conector C, e o *encoder* da roda esquerda foi ligado ao ENC1 (canal A pino 15, canal B no pino 17) do conector C.

A quantidade de pulsos por volta ficou diferente para cada roda, assim, a resolução por volta ficou com 56350 pulsos por volta para a roda direita e 50880 pulsos por volta para a roda esquerda. O raio das rodas é de 24 cm e a distância entre os eixos é de 58 cm. A instalação do *encoder* pode ser vista na figura 37(a), pelo lado interno, como mostrado na figura 37(b).

Figura 37 - Instalação do *encoder*: a) Visão lado externo e b) Visão lado interno



Fonte: Motta et al. (2017).

Foi feita uma adaptação no *joystick* da cadeira, sendo posicionada uma chave que alterna seu funcionamento de manual, que liga o controle do *joystick*, para funcionamento automático, utilizando a programação contida na placa NI myRIO™. Foi utilizada no projeto a placa NI myRIO™ para realizar a programação da inteligência da cadeira elétrica. Esta placa tem a função de processamento em

tempo real do ARM® Cortex™-A9 dual-core e as E/S customizáveis do Xilinx FPGA (NATIONAL INSTRUMENTS, 2017). A placa NI myRIO™ foi ligada ao *joystick* da cadeira elétrica para que pudesse enviar os sinais de controle utilizando os PIDs nativos da cadeira de rodas.

Para realizar a instrumentação da cadeira de maneira que ela fosse transformada em cadeira de rodas robótica, houve um estudo detalhado sobre o funcionamento de seu painel. As funções são demonstradas na figura 38 e na tabela 9.

Figura 38 - Descrição do joystick



Fonte: Próprio autor (2017).

Tabela 9 - Descrição das funções do painel *connect*.

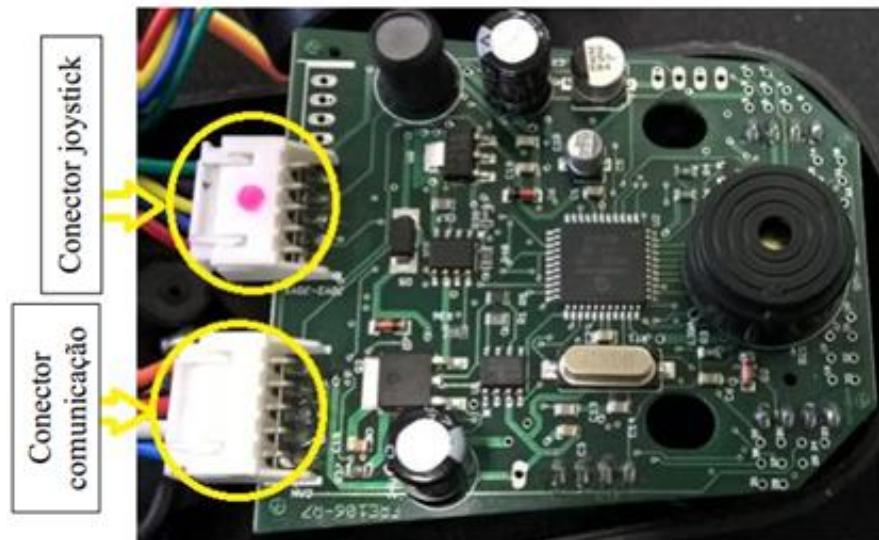
Referência	Legenda dos acionadores e indicadores
1	Indicador de carga da bateria da cadeira.
2	Indicador luminoso do farol.
3	Botão Liga-Desliga
4	Botão do Farol
5	Botão break/off
6	Indicador atenuador da velocidade máxima (30% a 100 %)
7	Indicador luminoso do modo turbo
8	Botão Menu
9	Botão da buzina
10	Botão modo turbo/econômico
11	Alavanca do <i>joystick</i>

Fonte: Próprio autor (2017).



Para a implementação do projeto foram analisadas as placas de cada um dos módulos para identificar os conectores para serem ligados a placa NI myRIO™. Na placa do módulo do painel *connect* existem dois conectores, identificados na figura 39 sendo que o conector pertencente ao *joystick* é que foi conectado a placa NI myRIO™. O conector de comunicação é responsável pelo envio do valor de referência ao módulo de potência dos motores (*Drive connect*) através de um barramento CAN. Ele também envia a alimentação ao painel *Connect*, além de alimentar e recarregar as baterias.

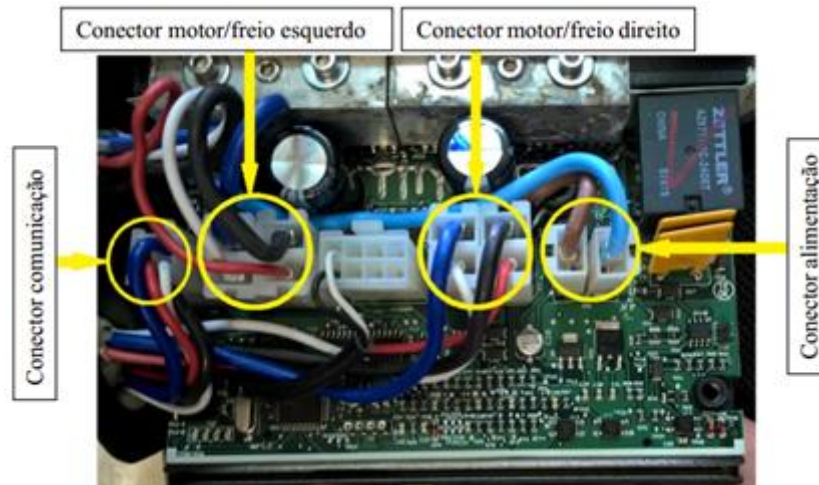
Figura 39 - Placa eletrônica do módulo do painel *connect*



Fonte: Próprio autor (2017).

O módulo de potência dos motores recebe o valor de referência e os aplica aos controladores dos motores. Nele também está conectada uma alimentação de 24 V das baterias através do conector de alimentação, conforme mostra a figura 40.

Figura 40 - Placa eletrônica do módulo do *drive connect*.



Fonte: Próprio autor (2017).

Cada conector contém fios com cores diferentes que representam as suas diferentes funções, conforme mostra a tabela 10.

Tabela 10 - Legenda de cores dos fios.

Localização	Cor	Significado
Conector de comunicação ( <i>Panel Connect</i> )	Azul	+24V
	Preto	0V
	Vermelho	CAN LOW
	Branco	CAN HI
	Laranja	COM SERIAL PC
Conector <i>joystick</i> ( <i>Panel connect</i> )	Azul	Pot_v
	Preto	0V
	Vermelho	+5V
	Verde	Desconhecido (manter 2,5V)
	Amarelo	Pot_w
Conector de comunicação ( <i>Drive Connect</i> )	Azul	+24V
	Preto	0V
	Vermelho	CAN LOW
	Branco	CAN HI
Conectores motor/freio esquerdo/direito ( <i>Drive Connect</i> )	Azul	Motor (+)
	Preto	Motor (-)
	Vermelho	Freio (+)
	Branco	Freio (-)
Conector alimentação ( <i>Drive Connect</i> )	Azul	+24V
	Marrom	0V

Fonte: Próprio autor (2017).

O sistema embarcado para o controle da cadeira é feito na placa NI myRIO™. A realização da comunicação com o NI myRIO™ é feita por rede WiFi, pois esse dispositivo possui uma placa de rede sem fio encapsulada. O carregamento do programa é feito pelo computador, além disso, outro programa roda no computador com a função de sistema supervisorio.

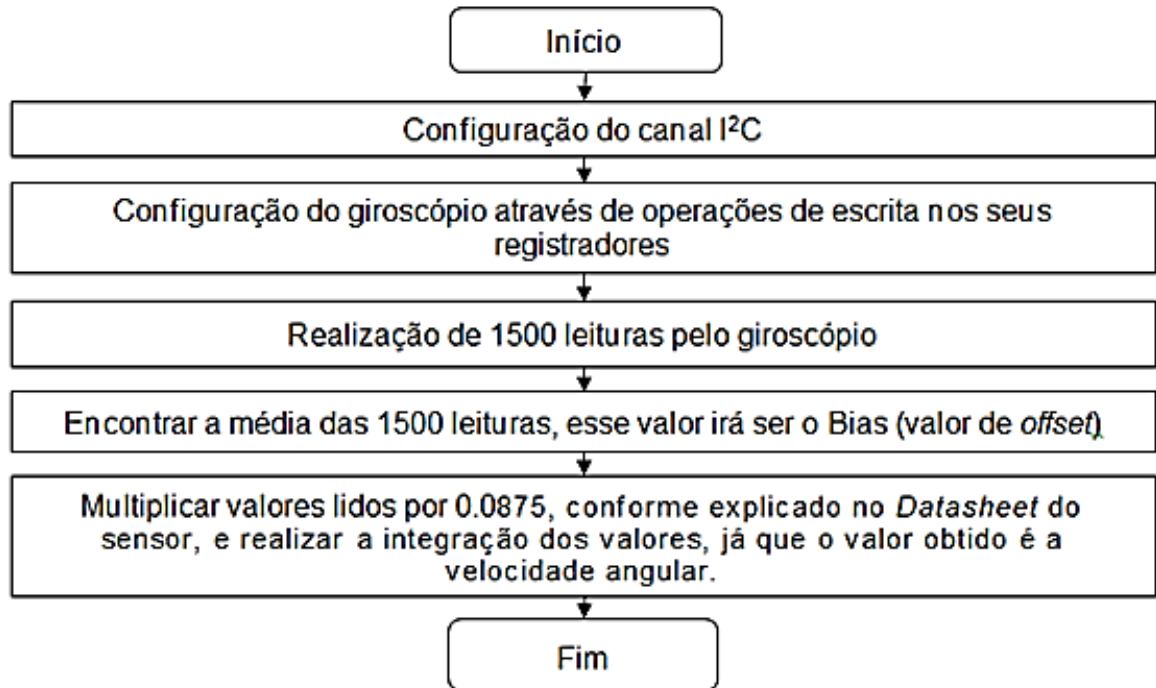
## 3.2 CALIBRAÇÃO DOS SENSORES

Todos os processos de medição que utilizam sensores estão sujeitos a variações conhecidas como “erros”. Utiliza-se sensores para a realização das leituras dos estados dos robôs, assim, é necessário realizar o tratamento das leituras dos sensores para que estas sejam as mais precisas possíveis. Portanto, tanto para o giroscópio quanto para o magnetômetro foram realizados procedimentos de calibração de modo a minimizar a incerteza da medição.

### 3.2.1 Giroscópio

Para calibrar o giroscópio foi necessária a leitura do *datasheet* de forma a realizar a configuração correta do sensor para realizar a leitura. O resultado da leitura do sensor é a velocidade angular bruta, a qual deve ser integrada para a obtenção da distância em graus. A etapa de calibração é descrita na figura 41.

Figura 41 - Procedimento de calibração do giroscópio



Fonte: Próprio autor (2017).

Em (54) é mostrada a equação utilizada para a leitura do giroscópio,

$$\theta = \int (0.00875 * x * \pi / 180) dt \quad (54)$$

onde,  $x$  é o valor bruto lido pelo giroscópio e  $\theta$  é o valor do ângulo lido (em radianos) acumulado durante o tempo.

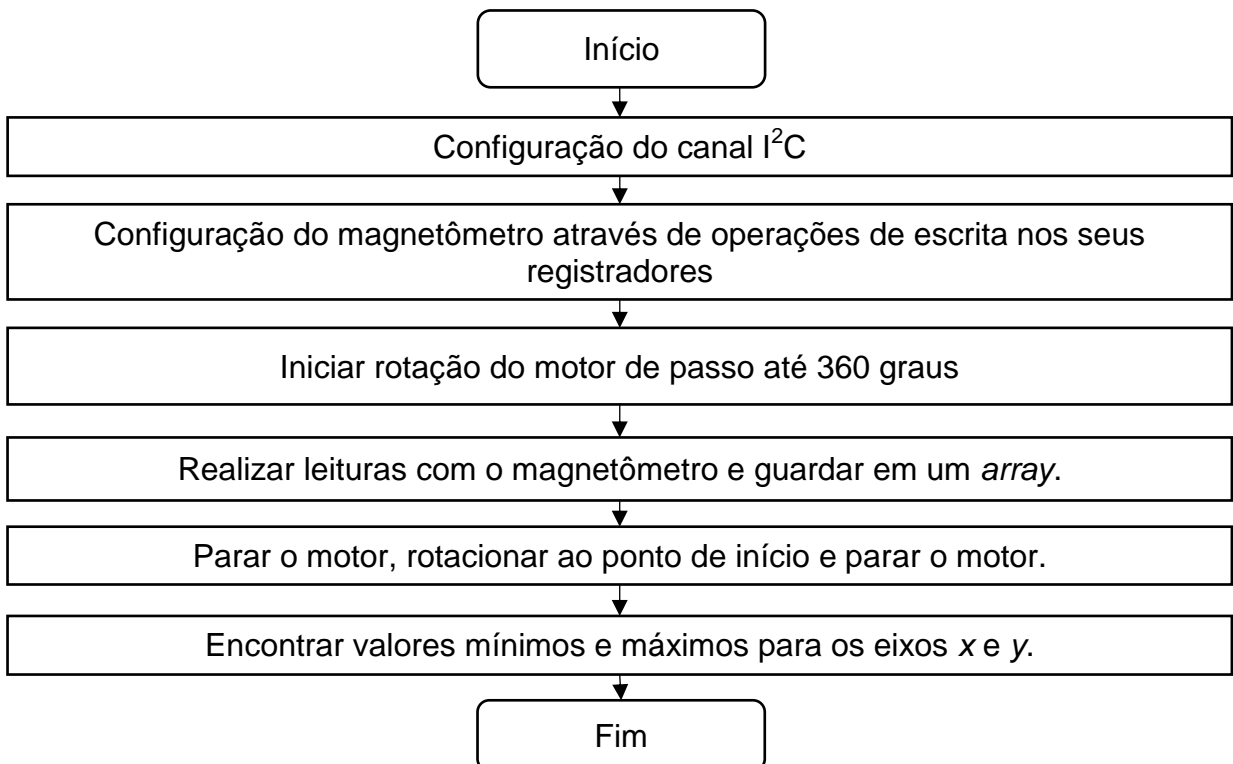
### 3.2.2 Magnetômetro

Para calibrar o magnetômetro foi necessária a leitura do *datasheet* de forma a realizar a configuração correta do sensor para realizar a leitura. A calibração deve ser realizada encontrando-se os valores mínimos e máximos de cada eixo  $x$  e  $y$  em relação ao campo magnético da terra, onde o eixo  $x$  se relaciona com o norte

magnético da Terra e, o eixo  $y$  se relaciona ao leste magnético da Terra. O eixo  $x$  tem seu valor máximo quando está somando o campo de sua bobina com o valor do campo magnético da Terra e, da mesma forma, tem seu valor mínimo quando o campo de sua bobina estiver em direção oposta ao campo magnético da Terra, ou seja, haverá uma subtração dos valores dos campos magnéticos. Para encontrar esses valores foi necessária a instalação do magnetômetro em cima do eixo de um motor de passo que o rotaciona enquanto os valores são lidos, sequencialmente, para obter os valores em cada direção.

Após os valores de máximo e mínimos serem encontrados, foi necessário achar o valor interpolado (interpolação linear), ou seja, cria-se um novo conjunto de dados relacionando-o a uma escala de 100 a -100 para os valores máximos e mínimos em cada eixo. Após a interpolação é encontrado o ângulo através da aplicação do arcotangente entre os eixos  $x$  e  $y$ , obtendo-se o respectivo ângulo de orientação da pose da cadeira robótica. A etapa de calibração é descrita na figura 42.

Figura 42 - Procedimento de calibração do magnetômetro.



Fonte: Próprio autor (2017).

### 3.3 ANÁLISE E COMPARAÇÃO DOS DADOS COM *INTEGRAL ABSOLUTE ERROR* (IAE)

Um índice de desempenho é uma avaliação quantitativa sobre a performance de um sistema, e é escolhido de modo que a ênfase seja dada as especificações de sistema importantes. Essa medida de desempenho de um sistema é necessária para:

1. Operação de sistemas de controle adaptativos;
2. Otimização paramétrica de sistemas de controle;
3. Controle ótimo.

Para a análise e comparação dos resultados do erro foi utilizado o índice de desempenho Integral Absolute Error (IAE), que é calculado através da integração do erro absoluto no período de tempo de amostragem. Em (55), é mostrado o cálculo do IAE.

$$IAE = \int_0^{\infty} |ea(t)| dt \quad (55)$$

Este índice foi escolhido devido a facilidade de implementação computacional e seu bom desempenho ao comparar controladores. O índice de desempenho IAE também reduzir a ponderação do grande erro inicial e penalizar os pequenos erros ocorrendo mais tarde na resposta mais forte (JAGATHEESAN; ANAND, 2012).

### 3.4 INSTRUMENTAÇÃO, MODELAGEM E *TUNNING* DOS GANHOS DOS CONTROLADORES DE VELOCIDADES UMA DA CADEIRA DE RODAS ROBÓTICA

Para obtenção dos dados foi realizada uma configuração prévia da cadeira de rodas robótica para se atingir um bom desempenho. A potência da cadeira foi regulada em nível dois, ou seja, foi aplicado no *joystick* da cadeira a potência 2. O recurso de turbo da cadeira foi desabilitado para que a resposta ficasse mais suave.

Foi obtido o modelo das velocidades da cadeira de rodas elétrica através de experimentação e análise de dados. Foram aplicados degraus de 1,3 V até 3,7 V, tanto na velocidade angular quanto na velocidade linear. Com os dados obtidos na velocidade angular foi feita uma média entre cada referência, obtendo-se o resultado mostrado na tabela 11.

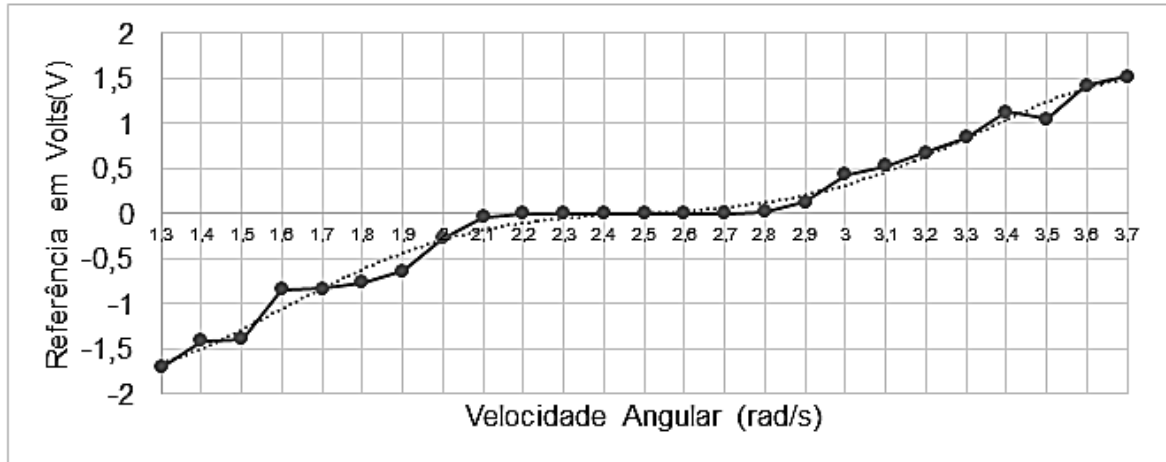
Tabela 11 - Dados de velocidades versus referência da velocidade angular.

Sinal de Controle (u)	Velocidade Angular (rad/s)
1,3	-1,707282945
1,4	-1,408545517
1,5	-1,387528009
1,6	-0,843808046
1,7	-0,834183011
1,8	-0,763170747
1,9	-0,635025907
2	-0,270819469
2,1	-0,037868155
2,2	0,000536974
2,3	0
2,4	0
2,5	0
2,6	1,54E-05
2,7	6,76E-05
2,8	0,017359745
2,9	0,125009418
3	0,433880406
3,1	0,535563779
3,2	0,68309087
3,3	0,846310649
3,4	1,127780074
3,5	1,047295733
3,6	1,424009934
3,7	1,52105822

Fonte: Próprio autor (2017).

Após colocar os dados da tabela 11 em um gráfico, percebe-se a não linearidade do modelo da velocidade angular, como mostrado na figura 43.

Figura 43 - Gráfico de representação da velocidade angular com diferentes referências



Fonte: Próprio autor (2017).

Com os dados conseguidos na velocidade linear foi feita uma média entre cada referência, obtendo-se o resultado mostrado na tabela 12.

Tabela 12 - Dados de velocidades versus referência da velocidade linear.

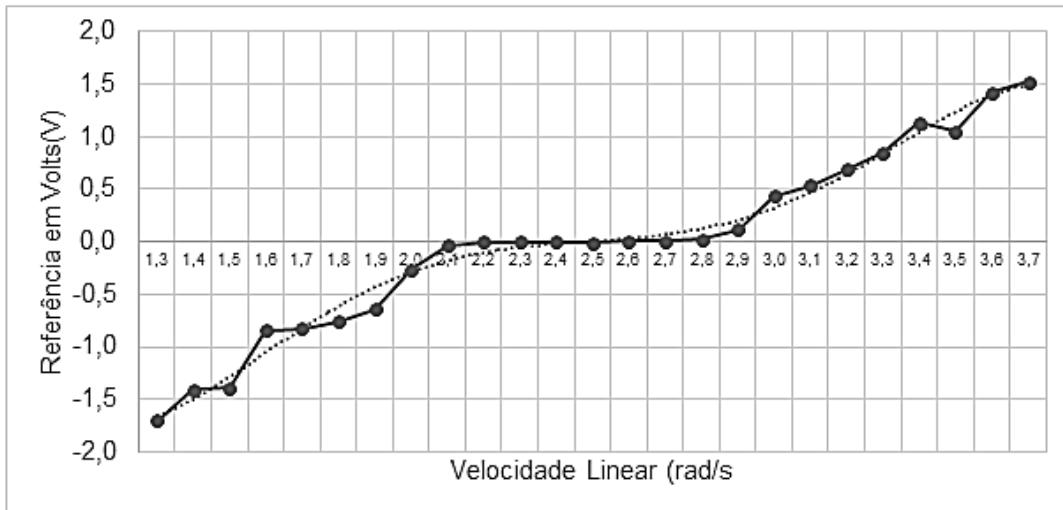
Sinal de Controle (u)	Velocidade Linear (rad/s)
1,3	-1,703515739
1,4	-1,407105426
1,5	-1,386228277
1,6	-0,845164008
1,7	-0,837379403
1,8	-0,767132929
1,9	-0,626147866
2	-0,274580452
2,1	-0,042387534
2,2	0,002475651
2,3	0,002475651
2,4	-0,00074653
2,5	-0,003973743
2,6	-0,000103692
2,7	0,004095999
2,8	0,013337542
2,9	0,123371982
3	0,433494536
3,1	0,536222719
3,2	0,684450786
3,3	0,852750528
3,4	1,125065501
3,5	1,041723544
3,6	1,422235573
3,7	1,522784638

Fonte: Próprio autor (2017).



Após colocar os dados da tabela 12 em um gráfico percebe-se a não linearidade do modelo da velocidade angular, como mostrado na figura 44.

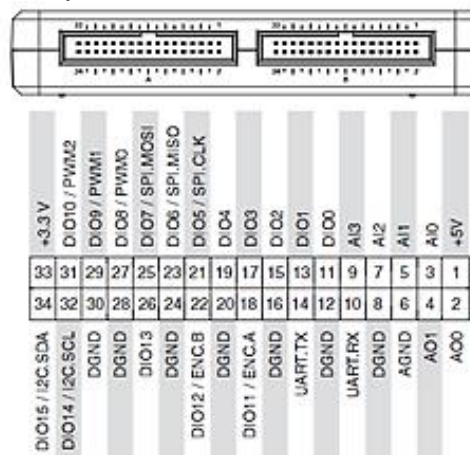
Figura 44 - Gráfico de representação da velocidade linear com diferentes referências



Fonte: Próprio autor (2017).

O dispositivo embarcado usado para realizar o processamento do sistema computacional foi o NI MYRIO™, o qual possui dois conectores A e B, com pinos que podem ser de uso geral para I/O, ou especializados, para serem utilizados por exemplo com rede I<sup>2</sup>C, conforme mostra a figura 45.

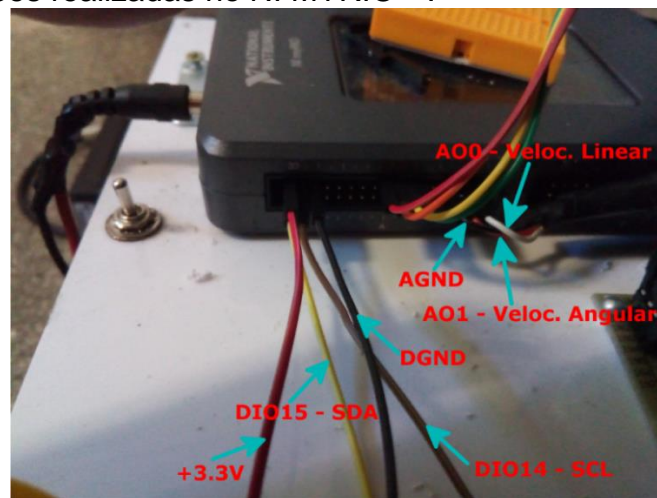
Figura 45 - Conectores do dispositivo embarcado NI MYRIO™



Fonte: Doering (2014).

Foi necessário realizar a ligação dos comandos do painel *connect*, que tem como entrada três condutores que são representados por três cores diferentes. O condutor branco foi ligado ao pino 02 - AO0 (responsável pelo sinal de controle da velocidade linear). O condutor vermelho foi ligado na porta 4 - AO1 (responsável pelo sinal de controle da velocidade angular), e o conector preto foi ligado ao pino 06 - AGND (valor de referência de terra - GND), conforme mostra a figura 46.

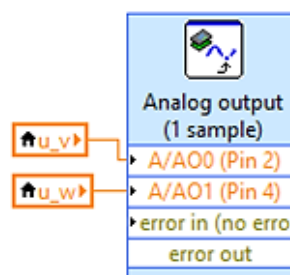
Figura 46 - Conexões realizadas no NI MYRIO™.



Fonte: Próprio autor (2017).

Para que a placa NI MYRIO™ possa enviar os sinais de controle para o painel *connect*, foi utilizado o bloco *analog output* presente no Labview™, conforme mostra a figura 47.

Figura 47 - Bloco *analog output* do Labview™



Fonte: Próprio autor (2017).

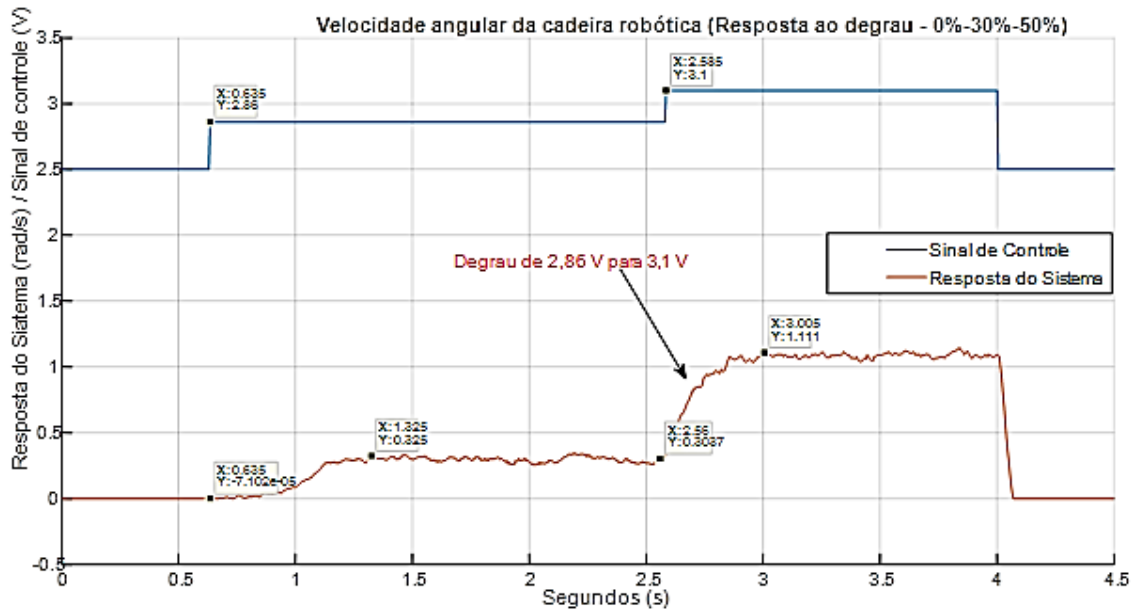
Foi realizado um ajuste para o valor do sinal de controle das velocidades, o qual varia entre 1,3 a 3,7 Volts. As referências para as velocidades da cadeira de rodas são dadas pelo *joystick*. O valor lido no ponto central do *joystick* é de 2,7 V para zerar a velocidade linear, e 2,7 V para zerar a velocidade angular, sendo que o valor mínimo aceito é 1,3 V e o valor máximo é de 3,7 V.

No programa desenvolvido foi criada uma equação, mostrada em (56), cuja função é de se dar entrada a um sinal de controle de -100 (valor mínimo) a 100 (valor máximo), relacionado às saídas mínimas e máximas de voltagens. Caso o módulo *connect* receba um valor fora do intervalo entre o valor máximo e o valor mínimo correspondentes aos seus sinais de controle, emitirá um *bip* sonoro e terminará sua operação.

$$AO(x)=1.2 \cdot x/100+2.5 \quad (56)$$

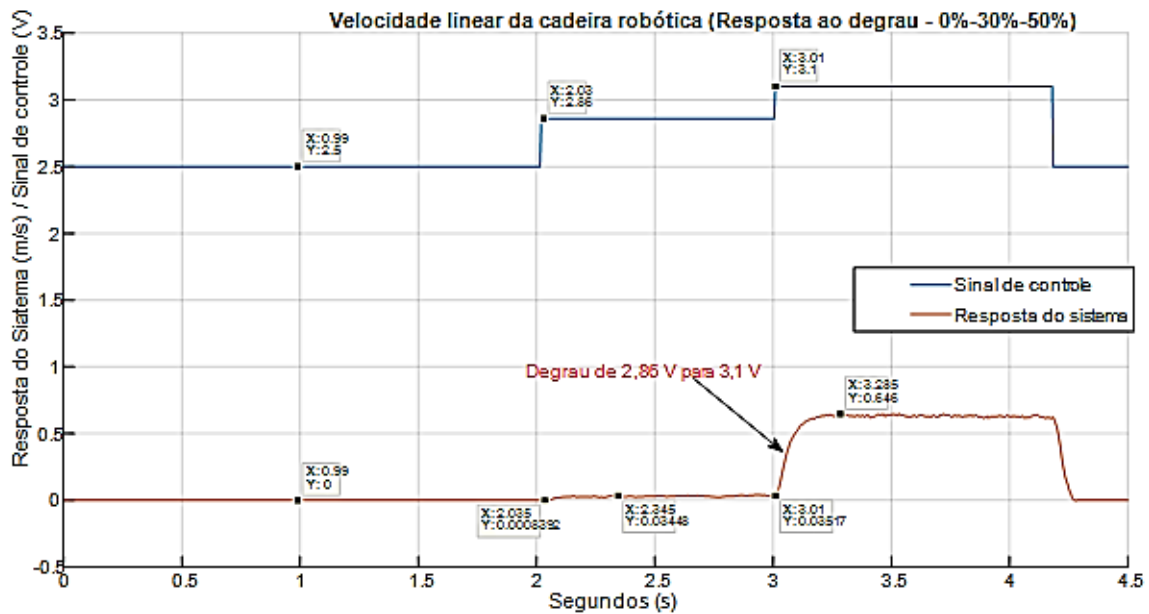
Para a construção de um controlador PI (proporcional, integral), foram aplicados degraus para se achar as funções de transferência, tanto para a velocidade linear quanto para a velocidade angular, referentes ao painel *connect*. Foram realizados testes de resposta ao degrau em malha aberta para cada velocidade. Como as referências dos comandos do painel *connect* são em Volts, manteve-se a velocidade linear nula, ou seja, recebendo 2,7 V, e foi comandado um degrau na velocidade linear inicial de 1,3 V até 3,1 V, sendo o mesmo procedimento realizado para a velocidade angular. Os resultados obtidos são apresentados nas figuras 48 e 49.

Figura 48 - Respostas ao degrau em malha aberta da velocidade angular da cadeira de rodas.



Fonte: Próprio autor (2017).

Figura 49 - Respostas ao degrau em malha aberta da velocidade linear da cadeira de rodas



Fonte: Próprio autor (2017).

O sistema pode ser aproximado por um sistema de primeira ordem. A forma geral de uma função de transferência de primeira ordem é mostrada em (57),

$$G(s) = \frac{K}{\tau s + 1}, \quad (57)$$

onde K é o ganho do sistema e  $\tau s$  é o tempo de subida que corresponde ao valor da saída corresponde a 63,2% do valor de saída do pico do processo. Em (58) é mostrada a forma de cálculo do ganho K e de  $\tau s$ ,

$$K = \frac{\Delta y}{\Delta u}. \quad (58)$$

Assim, é possível deduzir a equação de primeira ordem referente à velocidade linear da cadeira de rodas robótica conforme (59),

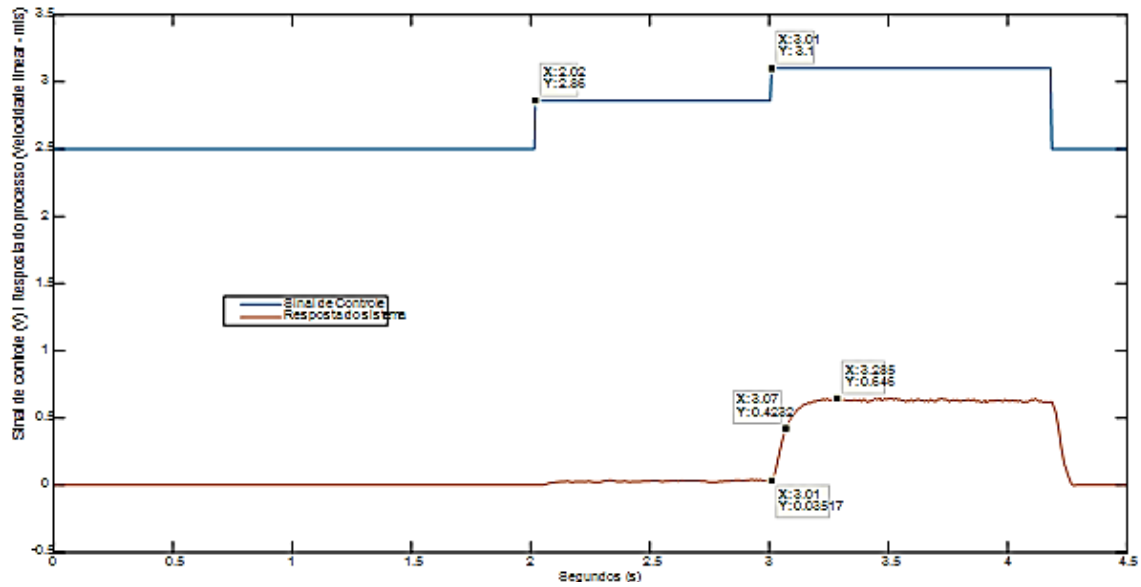
$$G_v(s) = \frac{2.545}{0.6s + 1}. \quad (59)$$

E também é possível deduzir a equação de primeira ordem referente à velocidade angular da cadeira de rodas robótica, conforme mostra (60),

$$G_w(s) = \frac{3.18}{0.12s + 1}. \quad (60)$$

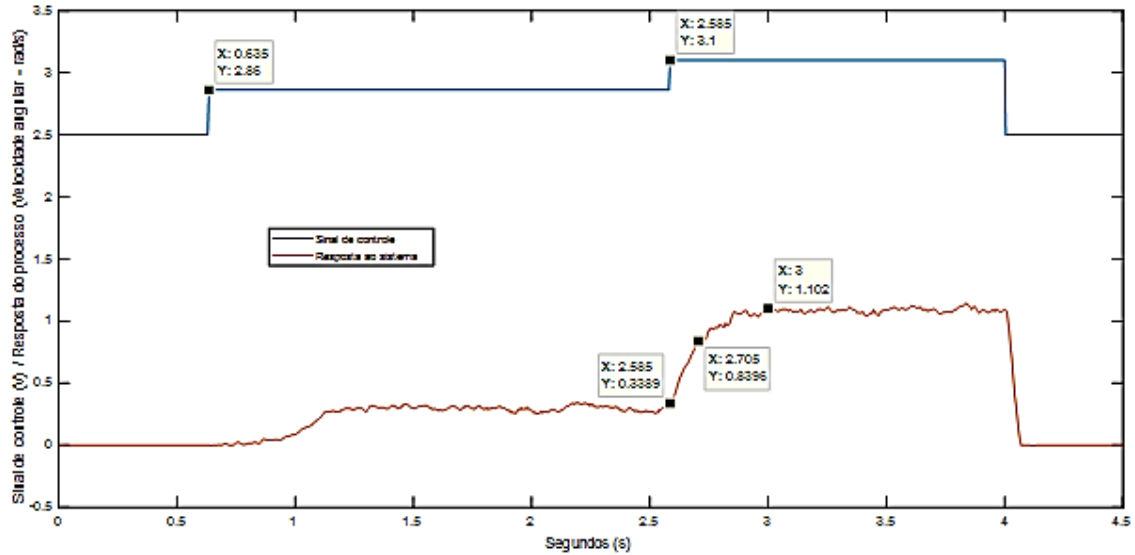
Foram realizados testes de resposta ao degrau em malha aberta sobre as funções de transferência das velocidades angulares e lineares da cadeira robótica. Como as referências dos comandos do painel *connect* são em Volts, manteve-se a velocidade linear nula, ou seja, recebendo 2,7 V, e foi comandado um degrau na velocidade linear inicial de 1,3 V até 3,1 V, sendo o mesmo procedimento realizado para a velocidade angular. Os resultados obtidos são apresentados nas figuras 50 e 51.

Figura 50 - Respostas ao degrau em malha aberta da velocidade linear da cadeira de rodas



Fonte: Próprio autor (2017).

Figura 51 - Respostas ao degrau em malha aberta da velocidade angular da cadeira de rodas



Fonte: Próprio autor (2017).

Para a realização da sintonia foi utilizado o método IMC para encontrar os ganhos do controlador PID. O  $\lambda$  escolhido foi de aproximadamente 5 ms, e os ganhos foram ajustados. Os parâmetros de sintonia usados nos testes práticos são mostrados na tabela 13.

Tabela 13 - Ganhos dos PID's das velocidades.

	PID da Velocidade Angular	PID da Velocidade Linear
Proporcional	0,48	45
Integral	0,55	0,9909
Derivativo	0	0

Fonte: Próprio autor (2017).

### 3.5 ERRO DE POSIÇÃO

O erro de posição do robô móvel é dado pela distância Euclidiana bidimensional. Para pontos bidimensionais,  $P=(p_x, p_y)$  e  $Q=(q_x, q_y)$  a distância é computada como mostrado em (61),

$$e = \sqrt{(XrV - Xr)^2 + (YrV - Yr)^2}, \quad (61)$$

onde  $p_x$  é representado por  $XrV$ ,  $q_x$  é representado por  $Xr$ ,  $p_y$  é representado por  $YrV$ ,  $q_y$  é representado por  $Yr$ . Para calcular o erro absoluto é utilizado o módulo em (61), com isso chega-se a (62),

$$ea = \left| \sqrt{(XrV - Xr)^2 + (YrV - Yr)^2} \right| \quad (62)$$

### 3.6 FILTRO DE KALMAN ESTENDIDO

O filtro de Kalman discreto é composto pelas variáveis:  $x$  (vetor de estados),  $u$  (vetor de entrada),  $z$  (vetor de medições),  $P$  (matriz de covariância dos erros da estimativa),  $K$  (matriz de ganhos de Kalman),  $Q$  e  $R$  (matrizes de covariância dos ruídos de processo e medição, respectivamente).  $Q$  e  $R$  são usados como parâmetros de

sintonia do EKF. O sobrescrito ‘-’ sinaliza um valor a priori, ou seja, significa o valor da variável antes da realização da medição; já o sobrescrito ‘^’ indica que aquela variável é estimada. As funções  $f$  e  $h$  bem como as matrizes  $F$  e  $H$  serão apresentadas adiante. Para este trabalho,  $x$  e  $z$  foram determinados conforme mostrado em (63) e (64),

$$\hat{x}_k = \hat{\theta}_k = \hat{\theta}_{Fusão_k}, \quad (63)$$

$$z_k = [\theta_{Odometria_k} \quad \theta_{Giroscópio_k}]^T. \quad (64)$$

Nesta Dissertação de Mestrado, a variável  $x$ , que é vetor de estados, é representada pela variável de estado de posições do robô móvel  $q = [x \ y \ \theta]^T$ . Em (9) é mostrada a função de transição de estados a qual utiliza o modelo cinemático do robô mostrado em (44) ao qual é não linear, com isso, justifica-se o uso do EKF que foi explicado na seção 0. A variável  $u$ , que é vetor de entrada, é utilizada como parâmetro de entrada da função de transição de estados. A variável  $W$  é o ruído inerente do processo, ou seja, os erros contidos no modelo cinemático do robô móvel em comparação com a representação real. Os símbolos sobrescritos  $k$  e  $k-1$  representam o valor da variável no instante atual e no instante anterior, respectivamente.

A matriz  $P$  representa a covariância do erro e está explicada na seção **Erro! Fonte e referência não encontrada.** Esta matriz tem a forma de uma matriz identidade com o tamanho da variável de estado no momento da sua inicialização, conforme mostrado em (65),

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}_{3 \times 3}, \quad (65)$$

a forma de calcular a matriz  $P$  é mostrada na Figura(b).  $K$  representa o ganho de Kalman explicado na seção **Erro! Fonte e referência não encontrada.** Esta matriz tem o tamanho de número de estados do robô móvel pelo número de estados utilizados pela fusão de dados. Um exemplo dessa matriz após seu cálculo é mostrado em (66),



$$K = \begin{bmatrix} 0.99 & 0 & 0 & 0 & 0 \\ 0 & 0.99 & 9.51e-06 & 9.51e-06 & 3.42 \\ 0 & 9.51e-06 & 0.42 & 0.42 & 0.15 \end{bmatrix}_{3 \times 5}, \quad (66)$$

a Forma de calcular o ganho de Kalman é mostrado na figura 2(c). A matriz  $R$  é a covariância do ruído da medição, essa matriz é quadrada e seu tamanho é dado pelo número de estados utilizados para a fusão, ou seja, o número de estados utilizados pelos sensores. Estes valores, geralmente, são obtidos nas documentações dos sensores e são ajustados por meio de experimentação. No caso aplicado são cinco estados, três utilizados pela odometria que são  $q=[x \ y \ \theta]^T$ , já para o giroscópio e o magnetômetro são apenas dois representados pelo estado theta que são representados como  $q=[\theta]^T$ , conforme mostrado em (67),

$$R = \begin{bmatrix} odo_x=0.55 & 0 & 0 & 0 & 0 \\ 0 & odo_y=0.55 & 0 & 0 & 0 \\ 0 & 0 & odo_\theta=0.55 & 0 & 0 \\ 0 & 0 & 0 & giro_\theta=0.55 & 0 \\ 0 & 0 & 0 & 0 & mag_\theta=0.90 \end{bmatrix}_{3 \times 5}, \quad (67)$$

onde  $odo_x$ ,  $odo_y$ ,  $odo_\theta$  representam os estados  $x$ ,  $y$  e  $\theta$  estimados pela odometria,  $giro_\theta$  representa o estado  $theta$  obtido pelo giroscópio,  $mag_\theta$  representa o estado  $theta$  obtido pelo magnetômetro.

A matriz  $Q$  representa a covariância do ruído do sensor. Esta matriz é uma matriz diagonal e quadrada com tamanho igual ao número de estados  $q=[x \ y \ \theta]^T$ . O valor dessa matriz deve ser achado de forma experimental, e é montada conforme mostrado em (68),

$$Q = \begin{bmatrix} 0.17 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 0.17 \end{bmatrix}_{3 \times 3}, \quad (68)$$

As funções  $f$  e  $h$ , mostradas em (69) e (70), representam o modelo do estado e das medições, respectivamente. As matrizes jacobianas, que linearizam o modelo e as medições, são representadas por  $F$  e  $H$ , conforme mostrado em (71) e (72).

$$f(\hat{\theta}_{k-1}, w_k) = \tilde{\theta}_k = \hat{\theta}_{k-1} + w_k T \quad (69)$$

$$h(\hat{x}_k) = [\hat{x}_k \quad \hat{x}_k]^T = [\hat{\theta}_k \quad \hat{\theta}_k]^T, \quad (70)$$

$$F_k = J_f(x_{k-1}) = \frac{\partial f(\theta_{k-1}, w_k)}{\partial \theta_{k-1}} = 1, \quad (71)$$

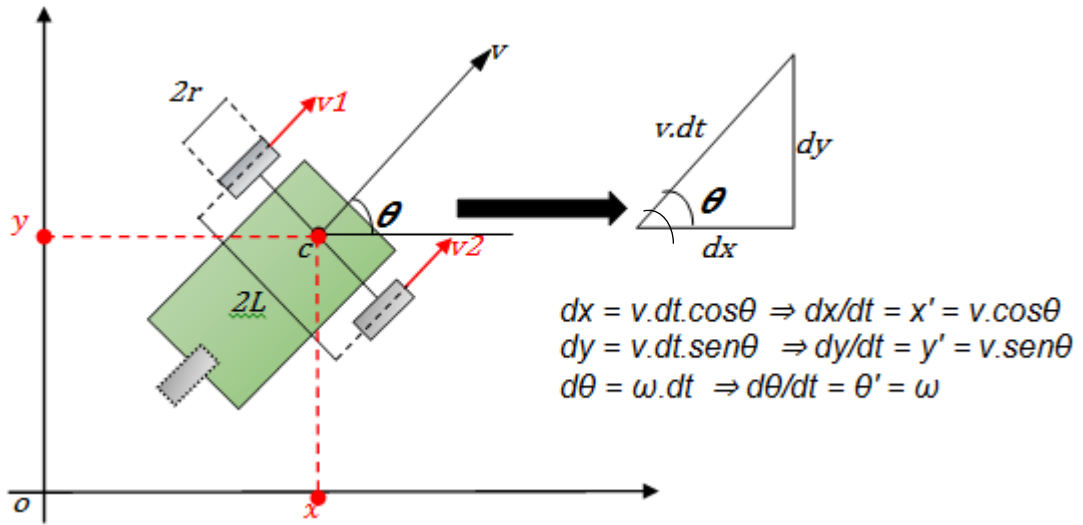
$$H_k = J_h(x_{k-1}) = \frac{\partial h(\theta_k)}{\partial \theta_k} = [1 \quad 1 \quad 1], \quad (72)$$

onde, para chegar ao resultado de  $F_k=1$  em (62) e  $H_k=[1 \quad 1 \quad 1]$  em (63) são calculadas derivadas parciais, das quais se obtém sempre os mesmos valores, com isso, os valores são constantes não sendo necessária a realização de cálculos a cada iteração. Para simplificar, a matriz  $F_k$  é considerada com valor igual a um, uma vez que daria o mesmo resultado caso fosse usada conforme a teoria que geraria uma matriz identidade de tamanho três por três. Essa matriz jacobiana é montada, sendo uma coluna para cada função de transição do modelo e cada linha para cada variável do modelo o resultado de cada elemento da matriz é a derivada parcial de cada função em relação a variável, mostrado em (73),

$$\begin{aligned} dx &= v \cdot dt \cdot \cos\theta & dy &= v \cdot dt \cdot \sin\theta & d\theta &= \omega \cdot dt \\ \begin{matrix} x \\ y \\ \theta \end{matrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & & & & \end{aligned} \quad (73)$$

Em (44) é mostrado o modelo do robô, sendo que está baseado na variação do deslocamento em  $x$ ,  $y$  e  $\theta$ . O deslocamento em  $x$  é denominado  $dx$ , o deslocamento em  $y$  é denominado  $dy$ , o deslocamento em  $\theta$  é denominado  $d\theta$ , cada deslocamento é dado por um cálculo matemático conforme mostra a figura 52.

Figura 52 - Origem do modelo cinemático do robô móvel utilizado



Fonte: Próprio autor (2017).

## 4 IMPLEMENTAÇÃO

Terminadas as etapas de estudo sobre os métodos utilizados, foi iniciada a etapa de implementação da aplicação. Inicialmente, foram implementados programas para apuração de resultados utilizando-se os três controladores de trajetória em ambiente de simulação. A posteriori, estes foram implementados nas plataformas robóticas e, em seguida, houve a sintonia dos controladores de trajetória com o método de tentativa e erro, sendo finalmente colhidos os resultados.

Para o melhor entendimento do algoritmo do EKF, implementou-se uma simulação. O EKF foi simulado no *software* Matlab™ para que se pudesse entender o seu funcionamento. Com isso, foram testadas as fusões e entendidas as adaptações dos tamanhos das matrizes em cada fusão de dados dos sensores. Finalmente, com o total domínio prático, foi iniciada a construção do EKF da aplicação final.

### 4.1 COMPARAÇÃO DE CONTROLADORES DE TRAJETÓRIA

Para a realização dos experimentos houve a necessidade de comparar e escolher o melhor controlador de trajetória para a aplicação proposta. Com esse objetivo foram criados programas para comparar os três controladores: *Fuzzy*, *Backstepping* e de ganhos fixos utilizando o robô *Starter kit 2.0*. Primeiramente foram implementados em simulação e, em seguida, na prática.

#### 4.1.1 Resultados dos controladores de trajetória

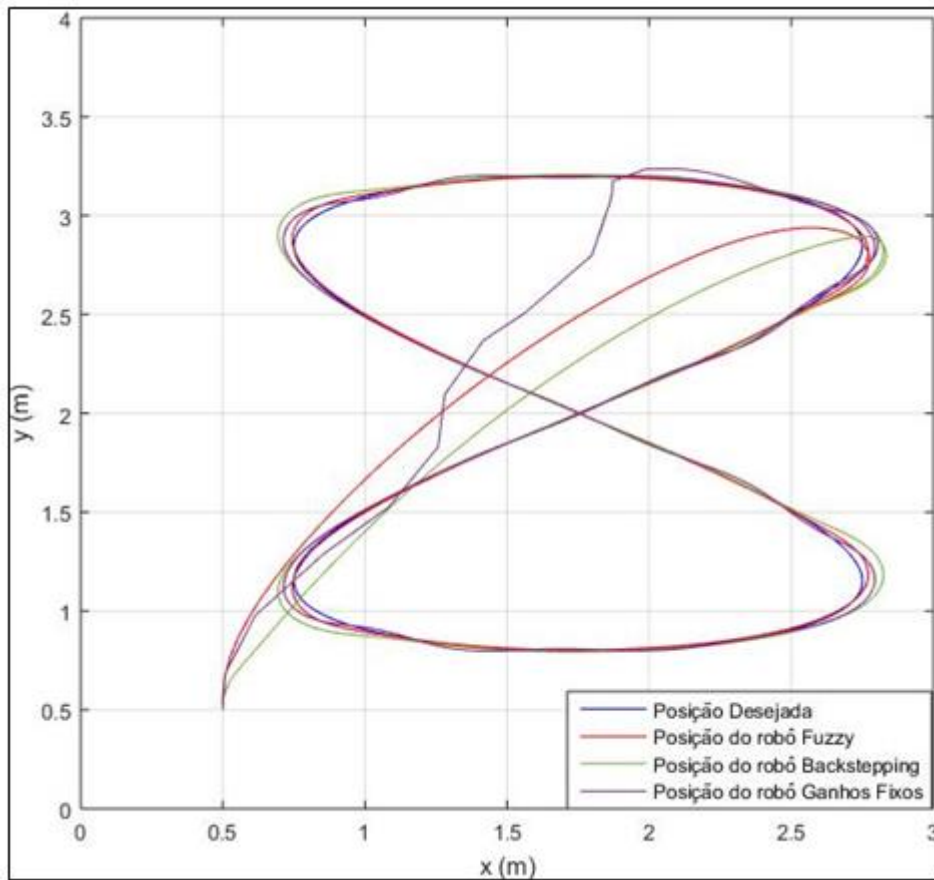
Os parâmetros dos controladores foram encontrados utilizando-se o método de tentativa e erro. Com os controladores ajustados chegou-se ao melhor controlador

para essa aplicação, que é o *Backstepping*, sendo que todos foram comparados com os resultados do sistema de processamento de imagem.

#### 4.1.2 Dados da simulação

O estudo simulado foi realizado para os três controladores mostrados na seção 2. As sintonias dos controladores foram realizadas utilizando-se métodos de tentativa e erro. A melhor sintonia do controlador *Fuzzy* foi  $K_v = 0,9$  e  $K_w = 1,5$ . A melhor sintonia do controlador *Backstepping* foi  $V_r = 0,1$ ,  $W_r = 0,001$ ,  $K_1 = 1,35$ ,  $K_2 = 290$ ,  $K_3 = 13$ , e a melhor sintonia do Controlador Cinemático de Seguimento de Trajetória de Ganhos Fixos, foi  $kg = -1,8$ ,  $po = 2,56$ ,  $a = 0,17$  m. Os erros de trajetória encontrados são mostrados na figura 53 para os três controladores estudados. Esses erros foram analisados e comparados utilizando-se o índice IAE mostrado na seção 0. Os dados mostrados correspondem à medição, que na simulação é dada pelo próprio ambiente de simulação, porém na experimentação prática é realizada pela câmera. Nos testes foi utilizada como trajetória a curva *lemniscate*. A curva *lemniscate* é muito utilizada na literatura devido ao seu formato, ao qual proporciona o teste de toda a dinâmica do robô (HERNÁNDEZ et al., 2015).

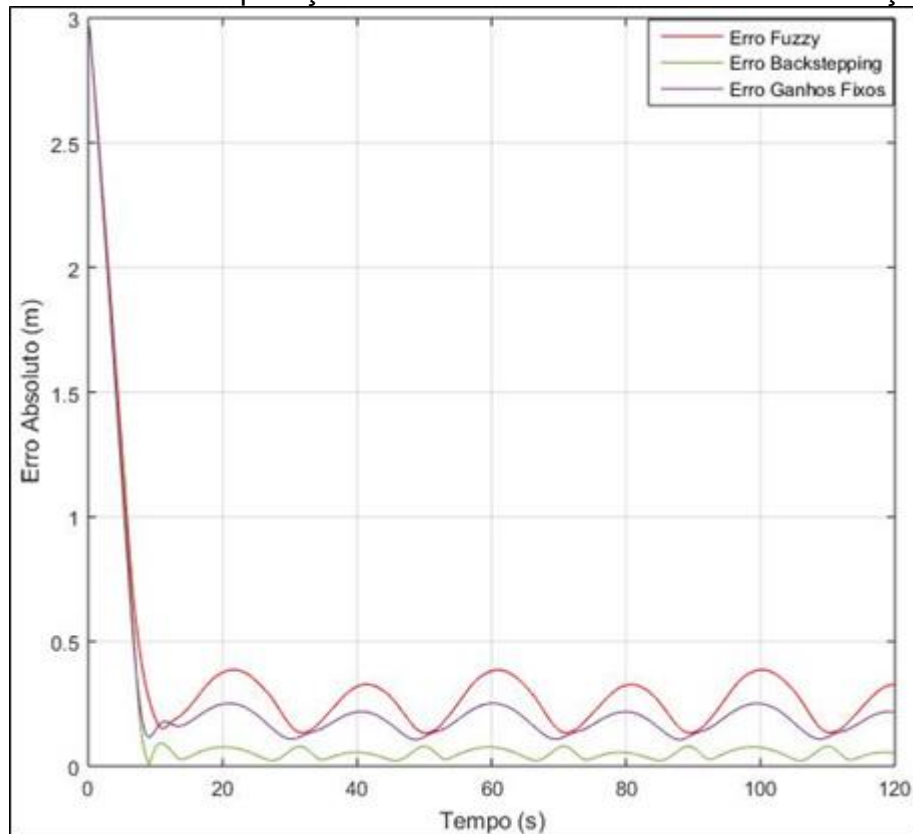
Figura 53 - Gráfico do experimento em simulação da trajetória.



Fonte: Próprio autor (2017).

Na seção 0 é mostrado o erro absoluto. As variáveis  $X_r$  e  $Y_r$  representam as posições em  $x$  e  $y$  que representam a posição do robô, e as variáveis  $X_{rV}$  e  $Y_{rV}$  representam as posições em  $x$  e  $y$  da posição desejada, conforme mostra a figura 54.

Figura 54 - Gráfico de comparação de erros de controladores em simulação



Fonte: Próprio autor (2017).

A tabela 14 contém os indicadores de desempenho dos três controladores, obtidos para cada variável durante as simulações.

Tabela 14 - Valores de IAE de cada controlador na simulação.

Variável	Backstepping	Cinemático CCFG	Fuzzy
IAE	21.69	44.31	59.76

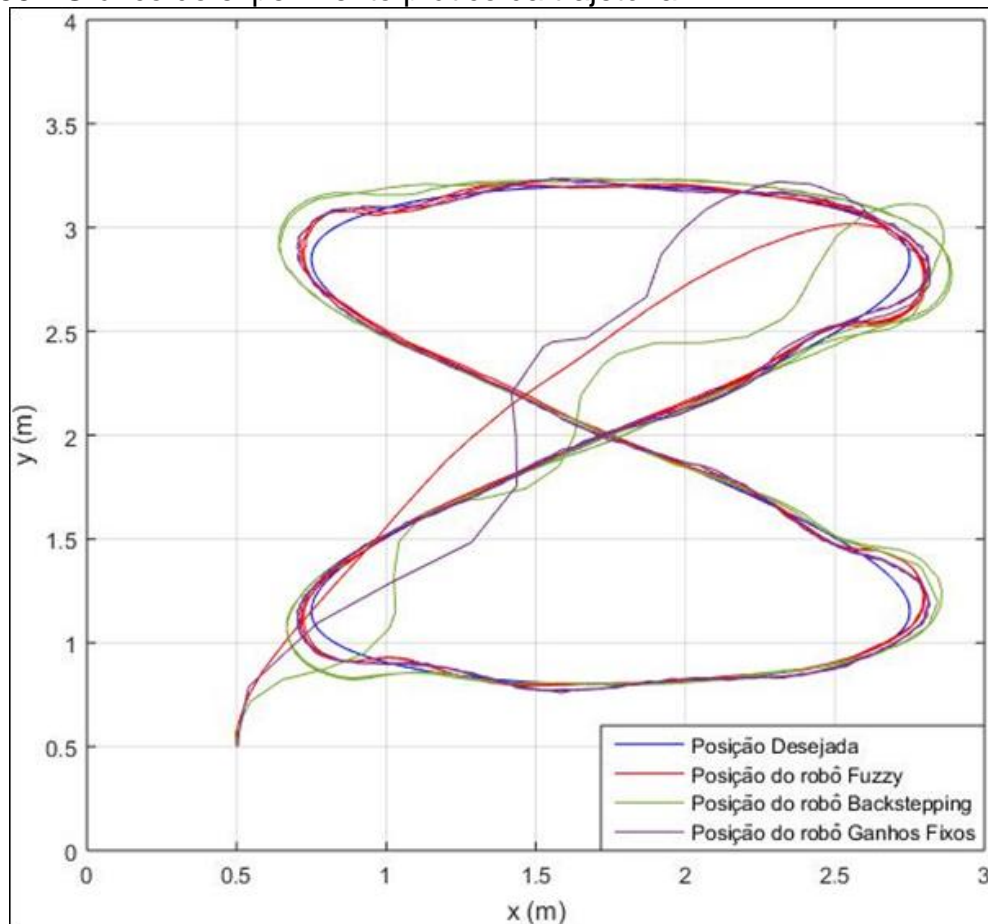
Fonte: Próprio autor (2017).

#### 4.1.3 Dados do experimento prático

O experimento prático foi realizado para os três controladores mostrados na seção 0. Os dados de sintonia utilizados foram os mesmos da simulação, com exceção dos dados do controlador de ganhos fixos, cuja sintonia é de  $a=0,17m$ . Esta diferença se deu devido à variação de localização obtida na simulação em relação à localização obtida na prática, através da aquisição da imagem realizada pela câmera.

Na figura 55 é mostrada a trajetória desejada em relação à trajetória do robô para os controladores anteriormente citados.

Figura 55 - Gráfico do experimento prático da trajetória.

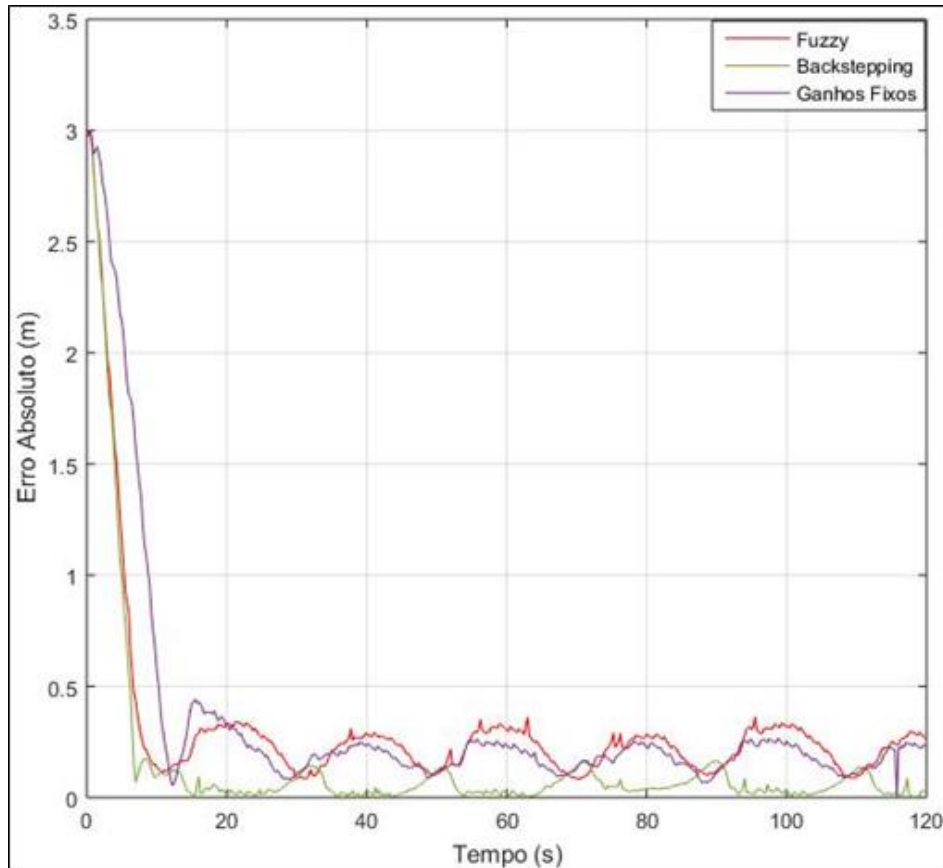


Fonte: Próprio autor (2017).

Na figura 56 são mostrados os dados de IAE para os controladores utilizados.

Figura 56 - Gráfico do erro absoluto do experimento prático





Fonte: Próprio autor (2017).

A tabela 15 contém os indicadores de desempenho dos três controladores, obtidos para cada variável durante os experimentos práticos.

Tabela 15 - Valores de IAE de cada controlador nos experimentos práticos.

Variável	Backstepping	Cinemático CCFG	Fuzzy
IAE	21.44	43.79	52,82

Fonte: Próprio autor (2017).

Como pode-se observar, tanto na tabela 14 quanto na tabela 15 o controlador *Backstepping* obteve o melhor desempenho devido ao menor valor do erro absoluto. Após a análise dos resultados conclui-se que o controlador *Backstepping* apresentou o melhor desempenho na simulação. A fim de comprovar que os dados de

simulação se repetiriam nos experimentos práticos, foram utilizados os mesmos dados de sintonia para os controladores, sendo que o demonstrou o melhor resultado na prática foi o controlador *Backstepping*.

## 4.2 IMPLEMENTAÇÃO DO CONTROLADOR DE TRAJETÓRIA NA CADEIRA DE RODAS ROBÓTICA

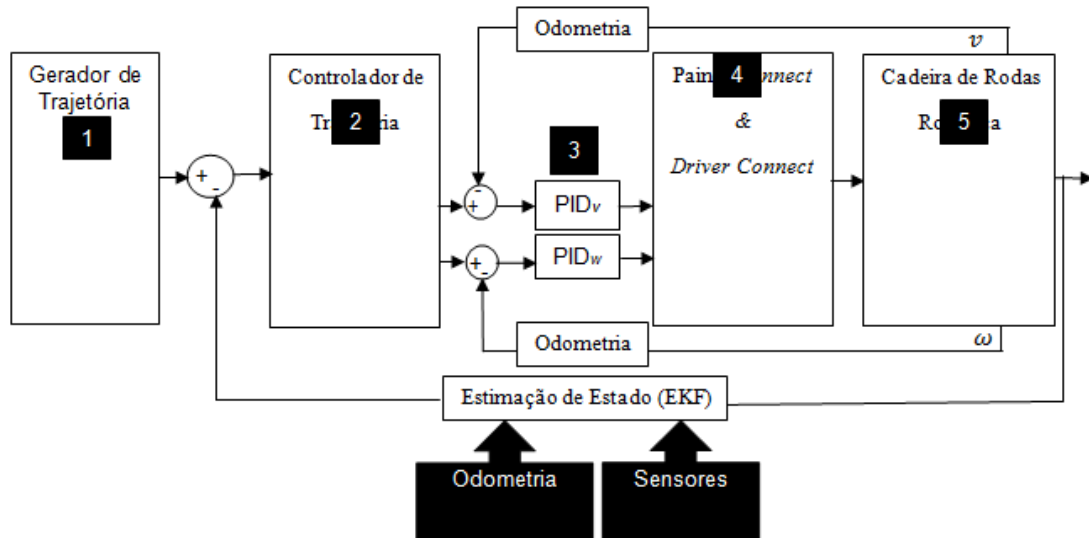
Nesta seção é explicada a implementação realizada na cadeira de rodas elétrica com o objetivo de transformá-la em uma cadeira de rodas robótica. Na seção 0 foram explicados os materiais e os métodos aplicados para a instrumentação, modelagem e *tunning* dos ganhos dos controladores de velocidades uma da cadeira de rodas robótica. Após a etapa mostrada na seção 0 foi feito o controle de trajetória, para isso, foi necessário criar um esquema de controle adaptado que é iniciado com o gerador de trajetória, conforme a etapa 1 da figura 57. O controle adaptado tem a função de traçar a trajetória que a cadeira robótica deverá seguir no tempo.

O controlador de trajetória tem a função de realizar o controle de navegação da cadeira de rodas em relação à trajetória traçada. As saídas do controlador de trajetória são as velocidades angular e linear da aplicação robótica, conforme mostra a etapa 2 da figura 57. Os controladores PIDs das velocidades angular e linear, mostrados na seção 0, da cadeira de rodas robótica são mostrados na etapa 3 da figura 57. As saídas dos PIDs são aplicadas ao painel *connect*, conforme mostra a etapa 4 da figura 57, e o painel *connect* envia sinais de controle para o *drive connect*. Na última etapa o *drive connect* envia sinais de controle para serem aplicados aos atuadores da cadeira de rodas robótica, como mostrado na etapa 5 da figura 57.

O estado do robô serve de *feedback* para a estimação do estado com EKF com o uso de *encoders*, odometria e os sensores, corrigindo o erro da localização da cadeira em relação ao gerador de trajetória. Devido ao ruído causado na leitura dos

*encoders*, o ganho derivativo foi configurado em zero, pois o valor de uma derivada no pico de um sinal ruidoso tende ao infinito, o que causaria um valor muito grande, tornando a ação de controle irrealizável.

Figura 57 - Esquema completo do controle da cadeira de rodas robótica



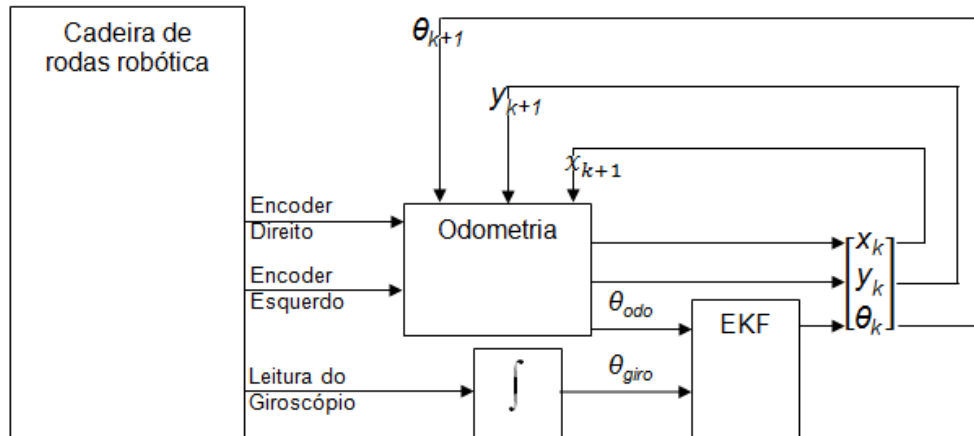
Fonte: Próprio autor (2017).

A pose do robô é dada pela matriz de estados  $q = [x \ y \ \theta]^T$ . Para a implementação da fusão de dados na cadeira de rodas robótica, foi escolhido o estado  $\theta$ , que é o estado representativo da orientação da cadeira de rodas robótica. Para implementação da fusão de dados, primeiramente, foi implementada uma fusão apenas com a odometria e o giroscópio. Então entre as três variáveis da postura do robô, apenas  $\theta$  foi fusionada.

A escolha da variável  $\theta$  foi escolhida por ser de crucial importância, pois é usada para efeito dos cálculos das variáveis  $x$  e  $y$ . Para realizar-se os testes, foram implementadas duas fusões. A fusão 1 é a fusão dos dados da odometria com os dados do giroscópio. A fusão 2 é a fusão dos dados da odometria, do giroscópio e do magnetômetro. Após implementadas as fusões, elas foram comparadas entre si e com a odometria.

A fusão 1 utiliza a leitura dos *encoders* para o cálculo da odometria. A leitura realizada pelo giroscópio obtém um valor para  $\theta$ , o qual é fusionado com o valor  $\theta$  da odometria, conforme mostra o diagrama da figura 58.

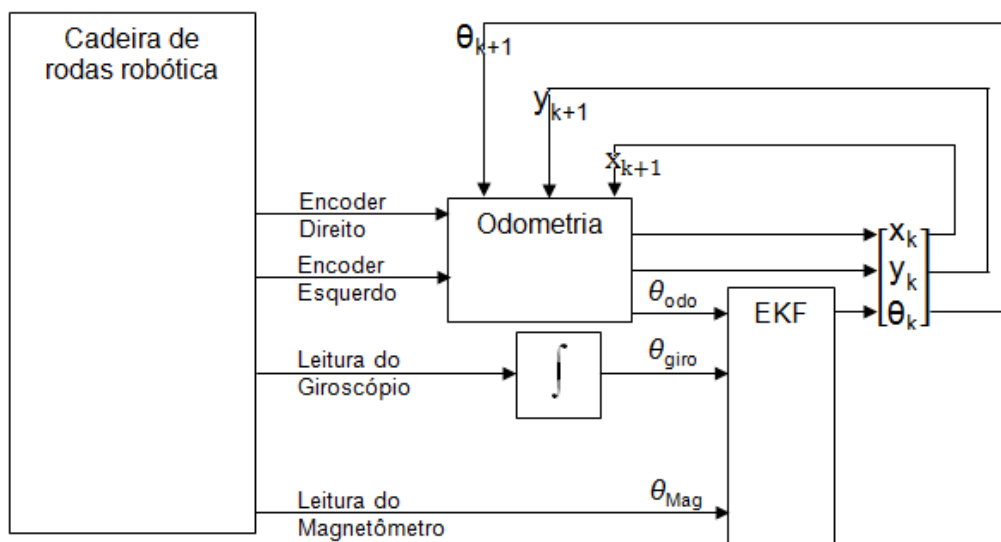
Figura 58 - Esquema de fusão sensorial implementado com odometria e giroscópio



Fonte: Próprio autor (2017).

Na fusão 2 foi incluído o magnetômetro ao EKF para obter outro valor para  $\theta$ , acrescentando mais uma entrada ao filtro, conforme mostra a figura 59.

Figura 59 - Esquema de fusão sensorial implementado com odometria, giroscópio e magnetômetro



Fonte: Próprio autor (2017).

### 4.3 RESULTADOS OBTIDOS

Com todo o ambiente de testes montado e com os ganhos do controlador PID das velocidades ajustados, foi realizado o método de tentativa e erro para encontrar os parâmetros de sintonia do controlador de trajetória *Backstepping* com a cadeira de rodas suspensa, conforme mostra a figura 60.

Figura 60 - Sintonia do Backstepping com a cadeira suspensa



Fonte: Próprio autor (2017).

Os ganhos do controlador *Backstepping* encontrados com a cadeira suspensa foram utilizados como ganhos iniciais para a sintonia da cadeira sobre o chão, com isso, todo o processo de sintonia foi repetido. Os melhores ganhos encontrados são mostrados na tabela 16.

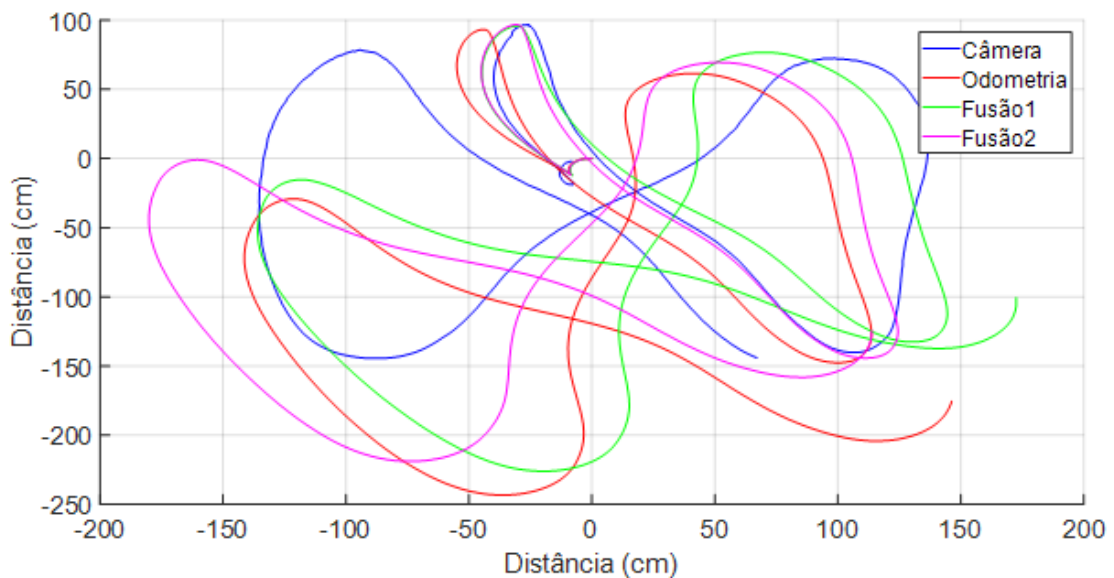
Tabela 16 - Ganhos do controlador de trajetória *Backstepping*

Parâmetros de sintonia	Valores dos ganhos
$K1$	0,64
$K2$	0,058
$K3$	17
$Vr$	0,07
$Wr$	0,05

Fonte: Próprio autor (2017).

Após os ajustes do controlador de trajetória, foram iniciados os testes com as fusões de dados. A fusão de dados apenas com sinais da odometria e giroscópio para o estado *theta* (fusão 1) foi implementada utilizando o EKF, e a segunda fusão com a odometria, giroscópio e magnetômetro, igualmente para o estado *theta* (fusão 2), também foi implementada com EKF, conforme mostrado na figura 61.

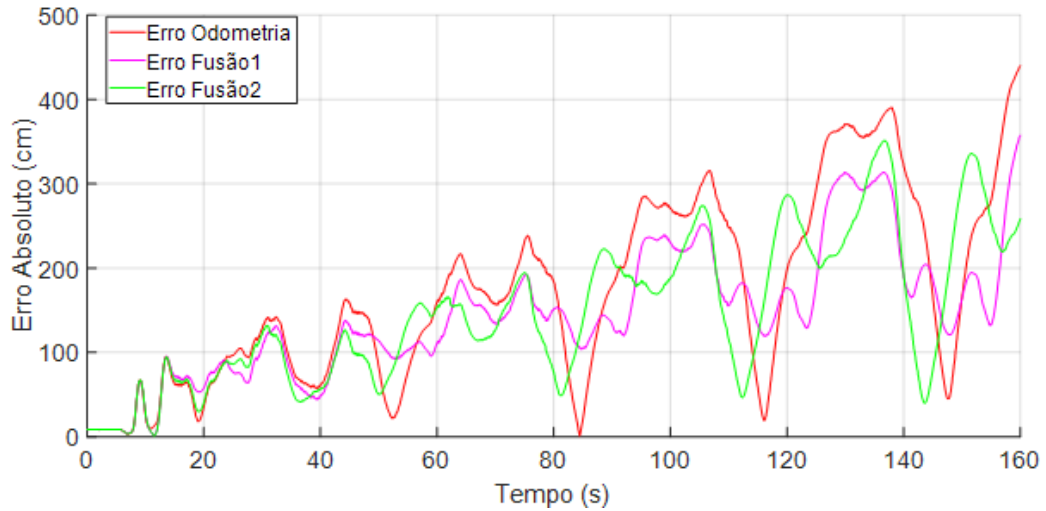
Figura 61 - Comparação da pose da câmera, da odometria e das duas fusões de dados



Fonte: Próprio autor (2017).

Como mostrado em (54), foi calculado o erro absoluto no tempo para odometria, fusão 1 e para a fusão 2, sendo que os resultados obtidos foram mostrados na figura 62.

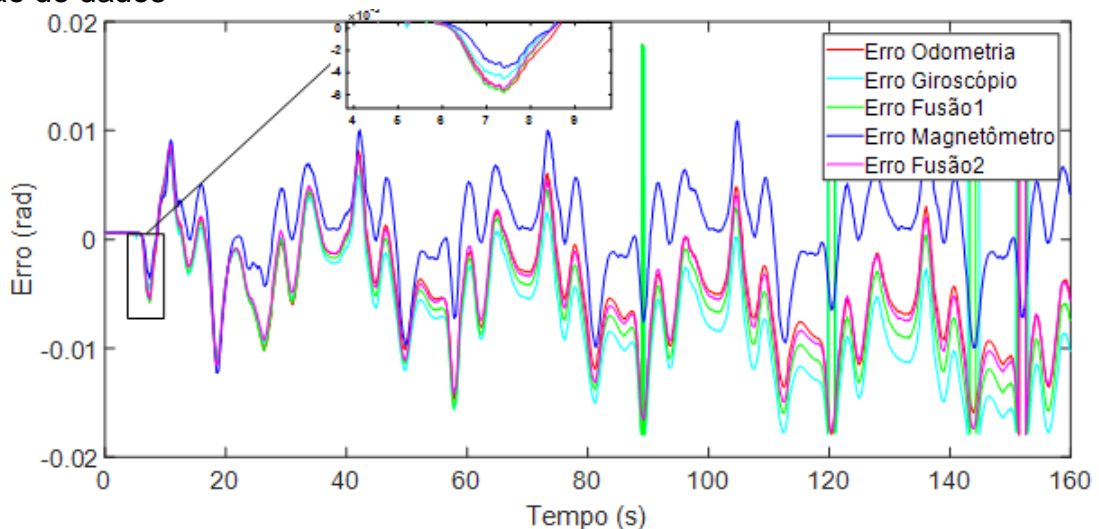
Figura 62 - Cálculo do Erro Absoluto no tempo para as duas fusões



Fonte: Próprio autor (2017).

Os erros nos valores dos estados *thetas*, referentes à odometria, ao giroscópio e ao magnetômetro, tanto para a fusão 1 quanto para a fusão 2, foram todos comparados com a pose obtida pela câmera. O motivo da fusão 2 ficar ruim é que o magnetômetro produz mais erros na sua medição que o giroscópio quando o programa é iniciado. Apesar da leitura do magnetômetro ser pior que a do giroscópio, a longo prazo o magnetômetro apresenta-se mais confiável, uma vez que o giroscópio acumula os erros com o passar do tempo, como mostra a figura 63.

Figura 63 - Comparação dos erros dos *thetas* da pose da câmera, da odometria e da fusão de dados



Fonte: Próprio autor (2017).

Na tabela 17 são mostrados os índices IAE calculados para os dados de pose obtidos pela odometria, pela fusão1, e pela fusão 2, em relação à pose obtida pela câmera, sendo mostrado o do índice IAE da odometria, fusão 1 (odometria e giroscópio) e fusão 2 (odometria, giroscópio e magnetômetro). Apesar da melhora da fusão 2 em relação à odometria, não houve melhora da fusão 2 em relação à fusão 1. Em (52) é mostrado o cálculo do IAE.

Tabela 17 - IAE de comparação dos dados obtidos pela odometria e pelas fusões de dados com os dados obtidos pela câmera.

Variável	Odometria	Fusão de dados 1	Fusão de dados 2
IAE	$3.3981 \times 10^4$	$2.8144 \times 10^4$	$2.8592 \times 10^4$

Fonte: Próprio autor (2017).

As variâncias do erro utilizadas para atingir os resultados mostrados foram escolhidas utilizando-se o método de tentativa e erro. Ao se iniciar as tentativas de ajustes dos valores das variâncias correspondentes à matriz  $R$ , que é a matriz de covariância do modelo e dos ruídos da medição no EKF, foram colocados valores altos, ou seja, 0,99, por exemplo, o qual foi reduzido de acordo com o resultado do erro de medição de cada método utilizado. Os valores encontrados para os valores da variância do erro são mostrados na tabela 18.

Tabela 18 - Valores da variância do erro da matriz  $R$ .

	Modelo	Odometria	Giroscópio	Magnetômetro
Variância do erro	0,17	0,55	0,55	0,9

Fonte: Próprio autor (2017).

Surgiram alguns desafios ao se utilizar o magnetômetro, como, por exemplo, utilizar um método de calibração do sensor, isolá-lo das fontes de campos magnéticos e, no



momento do experimento, foi necessário calibrá-lo com os motores da cadeira ligados, devido às interferências magnéticas exercidas por esses atuadores. No momento da partida dos motores das rodas da cadeira, também foi visível um maior erro lido pelo magnetômetro (devido à alta corrente de partida dos motores), enquanto o erro lido pelo giroscópio era pequeno. No entanto, o giroscópio, por ser incremental, acumula mais erros, o que não acontece a longo prazo com o magnetômetro.

## 5 CONCLUSÕES

Nesta Dissertação de Mestrado, foi criado um ambiente de testes para realizar os experimentos práticos de determinação da pose de uma cadeira de rodas baseado em sinais de odometria, giroscópio e magnetômetro. Uma câmera foi posicionada no teto do ambiente de testes para realizar o processamento de imagem com o intuito de validar os dados dos experimentos. O erro de *paralaxe* medido pela câmera foi corrigido através de equações matemáticas. Foi construído um *grid* no chão para melhorar a calibração com quadrados de cinco centímetros distribuídos em intervalos de cinquenta centímetros entre si com o propósito de utilizar um método de calibração automatizada. Com todos os métodos aplicados alcançou-se um erro na determinação da pose de cerca de dois centímetros pelo processamento de imagens. Com isso, o ambiente de testes foi implementado com sucesso e os experimentos práticos puderam ser realizados para a coleta de dados confiáveis.

Antes de implementar a fusão de dados para determinar a pose da cadeira de rodas, foram feitos estudos da implementação de controladores de trajetória num ambiente de simulação e num robô móvel real. Os resultados práticos mostraram que o controlador *BackStepping* teve melhor desempenho; assim este controlador foi usado para os testes na cadeira de rodas. Entretanto, é importante destacar que o objetivo principal da implementação da fusão sensorial não é avaliar o controlador e sim o desempenho da estimativa da pose do robô.

Dois esquemas de fusões sensoriais, baseados na utilização do Filtro de Kalman estendido, foram implementados. A primeira fusão sensorial utilizou os dados obtidos pela odometria e pelo giroscópio. A segunda fusão implementada acrescentou o magnetômetro à fusão anterior. Houve a necessidade de implementar métodos de calibração para os dois sensores utilizados para que eles obtivessem os dados da forma mais correta possível.

Dos resultados pode-se concluir que as duas formas de fusão de dados usando odometria obtiveram melhores índices de desempenho quando comparados com o uso único da odometria. Assim, o uso da fusão sensorial usando o Filtro de Kalman

estendido foi implementado com sucesso numa cadeira de rodas robótica. Entretanto, o resultado da fusão incluindo odometria, giroscópio e magnetômetro teve um desempenho próximo à fusão usando odometria e giroscópio. Acredita-se que este resultado foi devido aos ruídos provenientes dos motores da cadeira de rodas. Assim, será necessário aprofundar os estudos para melhorar a leitura realizada pelo magnetômetro.

Observando os resultados obtidos na cadeira de rodas robótica verificou-se que os erros vão se incrementando à medida que o tempo passa. Assim, será necessário utilizar *landmarks* para corrigir esses erros a cada certa distância, ou utilizar um filtro de partículas para a navegação de ambientes *indoor*.

O desenvolvimento dessa Dissertação de Mestrado produziu os seguintes artigos:

- “Uma abordagem didática para a instrumentação de cadeira de rodas aplicada a aprendizagem na engenharia”, aceito no CONBENGE 2017.
- “*Comparative of trajectory controllers: fuzzy, fixed gains and backstepping for robots with differential traction using image processing*”, aceito no SBAI 2017.
- “Fusão dos dados do *dead reckoning* e do giroscópio usando o filtro de kalman estendido aplicado à localização de uma cadeira de rodas autônoma”, aceito no SBAI 2017.

## REFERÊNCIAS

ANTONY, Richard T. **Principles of data fusion automation**. [S.l.]: Artech House, 1995.

BEZERRA, Clauber Gomes. **Localização de um robô móvel usando odometria e marcos naturais**. 2004. 112 f. Dissertação (Mestrado em Engenharia Elétrica) - Universidade Federal do Rio Grande do Norte, Natal, 2004.

DOERING, Ed. **NI myRIO-project essentials guide**. [S.l.]: National Instruments, 2014.

FARIA, H. G. et al. Fuzzy Trajectory Tracking Controller for Differential Drive Robots. In: INDUSCON 2016 is the 12th IEEE/IAS International Conference on Industry Applications, 2016, Curitiba. 12th IEEE/IAS International Conference on Industry Applications, 2016.

FIERRO, R.; LEWIS, F. L. Control of a nonholonomic mobile robot using neural networks. **IEEE Transactions on Neural Networks**, v. 9, n. 4, p. 589-600, jul. 1998.

FREEDOM, **Cadeira de rodas motorizada Freedom Styles**. Disponível em: <<http://www.freedom.ind.br/produto/saude/cadeiras-de-rodas-motorizadas/freedom-styles/#description>>. Acesso em: 9 mar. 2017.

FUKAO, T.; NAKAGAWA, H.; ADACHI, N. Adaptive tracking control of a nonholonomic mobile robot. **IEEE Transactions on Robotics and Automation**, v. 16, n. 5, p. 609-615, 2000.

GANGANATH, N.; LEUNG, H. Mobile robot localization using odometry and kinect sensor. **IEEE**, jan. 2012. p. 91-94.

GELB, Arthur. **Applied optimal estimation**. [S.l.]: MIT Press, 1974.

GOIS, João Francisco M. **Sistema integrado de magnetômetros para a navegação de veículos autônomos**. 2015. Dissertação (Mestrado em Engenharia Eletrotécnica e de Computadores) - Universidade de Coimbra, Coimbra, Portugal, 2015.

GREWAL, M. S; ANDREWS, A. P. **Kalman Filtering: theory and practice using Matlab**. 3th ed. New York: Wiley-IEEE Press, 2008.

HEINEN, Farlei José. **Sistema de controle híbrido para robôs móveis autônomos**. 2002. Dissertação (Mestrado em Computação Aplicada) - Universidade do Vale do Rio do Sinos, São Leopoldo, RS, 2002.

HERNÁNDEZ, Francisco Rosales et al. Synchronization strategy for differentially driven mobile robots: discrete-time approach. **International Journal of Robotics and Automation**, v. 30, n. 1, 1 jan. 2015.

IBARRA-MANZANO, M. A. Intelligent algorithm for parallel self-parking assist of a mobile robot. **IEEE**, nov. 2012. p. 37-41.

ISHLINSKII, A. Iu. **Orientation, gyroscopes, and inertial navigation**. Moscow: Izdatel'stvo Nauka, 1976. 671 p.

JAGATHEESAN, K.; ANAND, B. Dynamic performance of multi-area hydro thermal power systems with integral controller considering various performance indices methods. **IEEE**, dez. 2012. p. 474-478.

JIANG, Z .P. NIJMEIJER, H. Tracking control of mobile robots: a case study in backstepping. **Automatica**, v. 33, n. 7, p. 1393-1399, 1 jul. 1997.

JULIER, S.; UHLMANN, J.; DURRANT-WHYTE, H. F. A new method for the nonlinear transformation of means and covariances in filters and estimators. **IEEE Transactions on Automatic Control**, v. 45, n. 3, p. 477-482, mar. 2000.

KALMAN, R. E. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, v. 82, n. 1, p. 35-45, 1960.

KANAYAMA, Y. et al. A stable tracking control method for an autonomous mobile robot. **IEEE Comput. Soc. Press**, p. 384-389.

KARLSTROEM, Adriana. **Estimação de posição e quantificação de erro utilizando geometria epipolar entre imagens**. 2007. Dissertação (Mestrado em Engenharia Mecânica) – Escola Politécnica, Universidade de São Paulo, São Paulo, 2007.

KING, R. **Mobile robotics experiments with DaNI**. Colorado, USA: School of Mines and National Instruments, 2011.

KONVALIN, C. **Compensating for tilt, hard-iron and soft-iron effects**. Sens. Mag, p. 1-11, 2009.

LEVY, Simon D. The **Extended Kalman Filter**: an interactive tutorial for non-experts. Disponível em: <[https://home.wlu.edu/~levys/Kalman\\_tutorial](https://home.wlu.edu/~levys/Kalman_tutorial)>. Acesso em: 14 jan. 2017.

LIGGINS, Martin; HALL, David; LLINAS, James. **Handbook of multisensor data fusion: theory and practice**. Florida, USA: CRC Press, 2008.

MARTINS, F. N. **Modelagem e compensação da dinâmica de robôs móveis e sua aplicação em controle de formação**. 2009. Tese (Doutorado em Engenharia Elétrica). Universidade Federal do Espírito Santo, Vitória, ES, 2009.

MICHEL, Olivier; ROHRER, Fabien; HEINIGER, Nicolas. **Cyberbotics' Robot Curriculum**. Disponível em: <<http://doer.col.org/handle/123456789/4117>>. Acesso em: 7 nov. 2017.

MOTTA, Vinícius da R. et al. Comparative of trajectory controllers: fuzzy, fixed gains and backstep-ping for robots with differential traction using image processing. SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI), Porto Alegre, 2017.

MOTTA, Vinícius da R. et al. Uma abordagem didática para a instrumentação de cadeira de rodas aplicada a aprendizagem na engenharia. CONGRESSO BRASILEIRO DE EDUCAÇÃO EM ENGENHARIA (CONBENGE). 45., Joinville, SC, 2017.

NASCIMENTO, Rafaella Cristianne Alves. **Localização de robôs móveis em ambientes fechados em tempo real utilizando câmeras montadas no teto**. 2014. Dissertação (Mestrado em Engenharia Elétrica e de Computação) - Universidade Federal do Rio Grande do Norte, Natal, 2014.

NATIONAL INSTRUMENTS. **Overview of the LabVIEW Robotics Simulator**. Disponível em: <<http://www.ni.com/white-paper/14133/en/>>. Acesso em: 19 abr. 2017.

\_\_\_\_\_. **NI myRIO**. Disponível em: <<http://www.ni.com/myrio/pt/>>. Acesso em: 10 mar. 2017.

PANIAGUA, Leonardo Jose Marin. **Navegación de un robot móvil de configuración diferencial basada en fusión sensorial**. Valencia, Espanha: Universidade Politécnica de Valencia, 2012.

POLOLU. **MinIMU-9 Gyro, Accelerometer, and Compass (L3G4200D and LSM303DLH Carrier)**. Disponível em: <<https://www.pololu.com/product/1268>>. Acesso em: 3 out. 2017.

PORTELA, Thales da Cruz. **Arquitetura de hardware dedicada para o cálculo da distância frontal baseado em segmentação por cores usando visão estéreo aplicada em robótica móvel**. 2014. Monografia (Graduação em Engenharia Eletrônica) - Universidade de Brasília, DF, 2014.

RESENDE, C. Z. et al. Controlador de seguimento de trajetória para robôs móveis com ganhos dinâmicos. SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI). 10., 2011, São João Del Rei. [Anais...]. São João Del-Rei, p. 983-988, 2011.

ROMERO, R. A. F. et al. **Robótica móvel**. Rio de Janeiro: LTC, 2014.

SALAROLLI, Pablo F.; MOTTA, Vinícius da R.; CUADROS, Marco Antonio de S. Leite. Fusão dos dados do dead reckoning e do giroscópio usando o filtro de kalman estendido aplicado à localização de uma cadeira de rodas autônoma. SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI). 2017, Porto Alegre. [Anais...]. Porto Alegre, 2017.

SECCHI, Humberto Alejandro. **Una introducción a los robots móviles**. Instituto de Automática (INAUT). Argentina: Universidade Nacional de San Juan (UNSJ), 2008.

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

SIMON, Dan. **Optimal state estimation: Kalman, H [infinity] and nonlinear approaches**. [S.l.]: Wiley-Interscience, 2006.

THRUN, Sebastian. Probabilistic robotics. **Communications of the ACM**. v. 45, n. 3, p. 52-57, 1 mar. 2002.

TOMMASI, Eduardo Enrico V. Estudo comparativo de controladores de seguimento de trajetória para robôs de tração diferencial: fuzzy, ganhos fixos e backstepping. SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE (SBAI). 2015, Natal, RN. [Anais...]. Natal, RN, 2015.

VALGAS FILHO, João Hilário de Ávila. **Uma metodologia de correção dinâmica de erros de odometria em robôs móveis**. 2002. Dissertação (Mestrado em Ciência da Computação) - Universidade Federal de Minas Gerais, Belo Horizonte, 2002.

VIANNA, Ilca Oliveira de Almeida. **Metodologia do trabalho científico: um enfoque didático da produção científica**. São Paulo: EPU, 2001.

WELCH, Greg; BISHOP, Gary. **An introduction to the Kalman filter**. North Carolina, USA: Department of Computer Science, University of North Carolina, 2006.

YE, Jun. Tracking control for nonholonomic mobile robots: integrating the analog neural network into the backstepping technique. **Neurocomputing**, v. 71, n. 16-18, p. 3373-3378, 1 out. 2008.

YU, W. et al. Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle. Robotics. **IEEE Transactions on Robotics**, v. 26, n. 2, p. 340-353, 2010.