

Henrique Ricardo Rocha Lasevitch

**Controle Não-Linear do Veículo
Omni-Direcional via Servovisão e Sensor Inercial**

Porto Alegre - RS, Brasil

2014

Henrique Ricardo Rocha Lasevitch

Controle Não-Linear do Veículo Omni-Direcional via Servovisão e Sensor Inercial

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Sinais, Sistemas e Tecnologia da Informação

Linha de Pesquisa: Automação e Sistemas.

Pontifícia Universidade do Rio Grande do Sul – PUCRS

Faculdade de Engenharia

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Aurélio Tergolina Salton

Porto Alegre - RS, Brasil

2014



CONTROLE NÃO-LINEAR DO VEÍCULO OMNI-DIRECIONAL VIA SERVOVISÃO E SENSOR INERCIAL

CANDIDATO: HENRIQUE RICARDO ROCHA LASEVITCH

Esta Dissertação de Mestrado foi julgada para obtenção do título de MESTRE EM ENGENHARIA ELÉTRICA e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Pontifícia Universidade Católica do Rio Grande do Sul.

DR. AURELIO TERGOLINA SALTON - ORIENTADOR

BANCA EXAMINADORA

**DR. ALEXANDRE DE MORAIS AMORY - PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIA DA COMPUTAÇÃO - PPGCC - PUCRS**

DR. ALEXANDRE ROSA FRANCO - DO PPGE/FENG - PUCRS

À Deborah, ao meu pai, minha mãe e irmã por serem os meus grandes guias nessa longa jornada de aprendizado que é a vida.

Agradecimentos

Primeiramente gostaria de agradecer ao meu orientador Prof. Aurélio Tergolina Salton, por ter tornado possível a realização deste trabalho através da sua disposição e vasto conhecimento.

Agradeço também toda a comissão coordenadora e a equipe administrativa do PPGEE.

Agradeço ao Prof. Jeferson Flores, por todo o auxílio e orientações. Agradeço aos mestrandos, bolsistas e voluntários os quais tive o prazer de trabalhar no GACS e posteriormente no EASE.

Relato também um agradecimento especial aos grandes colegas e amigos Nelso Bedin e Leandro Lisboa pelos conselhos, ajudas, parceria e pelo tempo em que estivemos na Argentina; ao Leandro Disiuta e a Nathassia pelas conversas, almoços e trabalhos ao longo destes dois anos.

Sem vocês, estes dois anos não seriam tão agradáveis. Finalmente, um agradecimento especial a minha família pelo apoio, incentivo e dedicação para tornar este sonho uma realidade. À minha amada Deborah C. Jahn, por todos os finais de semana em casa para me acompanhar enquanto escrevia a dissertação, por todo amor, carinho e apoio, imprescindíveis para a conclusão do trabalho.

*“Mantenha seus pensamentos positivos, porque seus pensamentos tornam-se suas
palavras.
Mantenha suas palavras positivas, porque suas palavras tornam-se suas atitudes.
Mantenha suas atitudes positivas, porque suas atitudes tornam-se seus hábitos.
Mantenha seus hábitos positivos, porque seus hábitos tornam-se seus valores.
Mantenha seus valores positivos, porque seus valores...
Tornam-se seu destino.”*
(Mohandas Karamchand Gandhi)

Resumo

Veículos auto guiados são robôs capazes de se deslocar de maneira inteligente afim de cumprir diversas tarefas. Atualmente, pesquisas voltadas não somente para a operação em indústrias, mas também para veículos que possam atuar em ambientes inóspitos, complexos ou mesmo com mais agilidade e força que um ser humano estão sendo desenvolvidas em toda a parte do mundo. Esses veículos necessitam não só de uma maneira inteligente de se locomover, mas precisam também de sensores capazes de perceber mudanças no ambiente e realizar sua localização com sucesso. Este trabalho desenvolve um veículo Omni-Direcional - veículo capaz de se movimentar em um plano XY mantendo sua orientação inicial e também capaz de rotacionar em seu próprio eixo - tratando desde seu modelamento matemático, até o controle de sua trajetória, passando pelo sensoriamento utilizando um sensor de visão e um sensor inercial com um Filtro de Kalman Estendido para realizar a fusão destas informações. Será aplicado um controlador não-linear de tempo ótimo denominado PTOS em cada um de seus graus de liberdade. Para isso, um protótipo foi utilizado para ensaios reais e comparação com simulações feitas em um ambiente virtual.

Palavras-chaves: Veículo Autônomo, AGV, PTOS, Servovisão, Filtro de Kalman

Abstract

Auto guided vehicles are robots capable of moving in an intelligent way in order to accomplish different tasks. Nowadays, researches not only aiming industry operation, but also vehicles that may act in hostile environments, either complex or even with more skills and strength than a human being are being developed everywhere in the world. These vehicles need not only a smart way of displacement, but also need sensors capable of sensing changes in the environment and locate the device with success. This work develops an Omni-Directional vehicle - a vehicle capable of displacement in an XY plane without changing its original orientation and also capable of rotating about its own axis - going from the development of its mathematical model to the control strategy of the trajectory, going through sensing via Visual Servoing and inertial sensors fused with a Kalman Filter. Also, a prototype was used in experiments for comparison with simulations performed in a virtual environment.

Key-words: Autonomous Vehicle, AGV, PTOS, Visual Servoing, Kalman Filter.

Lista de ilustrações

Figura 1 – Detalhe da roda Mecanum.	17
Figura 2 – Esquema de montagem das rodas.	18
Figura 3 – Direções do veículo Omni-Direcional.	19
Figura 4 – Esquemático dos dispositivos do veículo Omni-Direcional.	20
Figura 5 – Ambiente de Desenvolvimento Integrado - Arduino	22
Figura 6 – Algoritmo de funcionamento do software do veículo.	23
Figura 7 – Algoritmo de funcionamento do software supervisorio.	24
Figura 8 – Esquemático completo do veículo, computador e câmera.	25
Figura 9 – Rotação entre um sistema de coordenadas e outro.	29
Figura 10 – Sinal de controle do TOC.	32
Figura 11 – Primeira etapa da lei de controle PTOS.	34
Figura 12 – Segunda etapa da lei de controle PTOS.	34
Figura 13 – Terceira etapa da lei de controle PTOS.	35
Figura 14 – Rodas e Sistema de Coordenadas Local do Veículo.	36
Figura 15 – Etapas de Pré-processamento da Imagem.	43
Figura 16 – Marca a ser captada pelo sensor de visão.	44
Figura 17 – Posição no eixo Y da câmera e filtrada.	47
Figura 18 – Filtro de Kalman Extendido aplicado ao modelo.	49
Figura 19 – Tempo de execução do sistema antes e após ser corrigido.	51
Figura 20 – Algoritmo de ajuste do tempo de amostragem.	52
Figura 21 – Identificação do Eixo X	54
Figura 22 – Identificação do Eixo Y	55
Figura 23 – Identificação do Eixo θ	56
Figura 24 – Esquemático do sistema completo.	57
Figura 25 – Simulação do modelo no eixo X e com controle Proporcional.	59
Figura 26 – Simulação do modelo no eixo X e com controle Proporcional-Derivativo.	60
Figura 27 – Simulação do modelo no eixo X e com controle PTOS.	61
Figura 28 – Simulação do modelo no plano XY e com controle Proporcional.	61
Figura 29 – Simulação do modelo no plano XY e com controle Proporcional-Derivativo.	62
Figura 30 – Simulação do modelo no plano XY e com controle PTOS.	62
Figura 31 – Simulação do modelo no plano XY, incluindo giro e com controle PTOS.	63
Figura 32 – Movimentos nos três eixos com referências distintas.	64
Figura 33 – Manobra no plano XY com giro com controlador PTOS.	64
Figura 34 – Quadro a quadro da manobra em "L".	65

Lista de tabelas

Tabela 1 – Diferenças entre o Filtro de Kalman e o EFK	49
Tabela 2 – Tabela de Relação dos sinais de cada roda.	52
Tabela 3 – Tabela dos Parâmetros Físicos do Sistema.	55

Lista de abreviaturas e siglas

AGV	<i>Auto Guided Vehicle</i>
DAR	<i>Differential Algebraic Representation</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
IDE	<i>Integrated Development Environment</i>
IMU	<i>Inertial Measurement Unit</i>
LFR	<i>Linear Fractional Representation</i>
PD	Proporcional-Derivativo
PTOS	<i>Proximate Time Optimal Servomechanism</i>
PWM	<i>Pulse Width Modulation</i>
TOC	<i>Time Optimal Control</i>
UART	<i>Universal Asynchronous Receiver Transmitter</i>
VANT	<i>Veículo Aéreo Não Tripulado</i>

Lista de símbolos

A'	transposta da matriz A
$sat(\cdot)$	função saturação
$sign(\cdot)$	função sinal
$ a $	valor absoluto do escalar a
\mathbb{R}^n	espaço euclidiano de ordem n

Sumário

1	Introdução	13
2	Preliminares	17
2.1	Idealização do Sistema Omni-Direcional	17
2.1.1	A Roda Mecanum	17
2.1.2	Hardware Utilizado	20
2.1.3	Software Desenvolvido	21
2.1.4	Sistema Supervisório	22
2.1.5	Funcionamento Completo	24
2.2	Representação de Rotação de Corpos	24
2.2.1	Translação	25
2.2.2	Rotação de Corpos	26
2.2.2.1	Matriz de Rotação	26
2.2.2.2	Ângulos de Euler	27
2.2.2.3	Ângulo e Eixo	28
2.2.2.4	Representação em Quatérnions	28
2.3	Servovisão	29
2.4	Servomecanismos e Controladores de Tempo Ótimo	30
2.4.1	Servomecanismos	30
2.4.2	Controlador de Tempo Ótimo	31
2.4.3	<i>Proximate Time Optimal Servomechanism</i>	32
3	Modelo	36
3.1	Coordenadas Locais	36
3.2	Coordenadas Globais	37
4	Sensoriamento e Fusão de Sensores	40
4.1	Servovisão	41
4.1.1	Configuração	41
4.1.2	Pré-Processamento	42
4.1.3	Identificação dos Elementos e Estimação da Posição	43
4.1.3.1	Posição do Veículo	44
4.1.3.2	Orientação do Veículo	44
4.2	<i>Inertial Measurement Unit</i>	45
4.3	O Filtro de Kalman	45
4.4	O Filtro de Kalman Extendido	46
5	Implementação e Controle do Sistema	50

5.1	Identificação do Sistema	50
5.1.1	Discretização dos Sistemas	50
5.1.2	Determinação do Tempo de Amostragem	50
5.1.3	Aquisição de Dados para Identificação	52
5.1.4	Método dos Mínimos Quadrados	53
5.2	Resultados da Identificação	53
5.3	Controle	56
5.3.1	Tipos de Controladores	57
5.3.2	Implementação dos Controladores	58
6	Resultados e Discussões	59
7	Conclusão	66
	Referências	68

1 Introdução

Veículos auto guiados (*Automated Guided Vehicles*, AGV) são robôs capazes de se deslocar de maneira inteligente afim de cumprir as mais variadas tarefas. Eles são comumente usados em ambientes industriais por aumentar a eficiência e reduzir custos de processos (SICILIANO; KHATIB, 2008). Não restritos a este meio, também são utilizados academicamente a fim de aumentar produção literária sobre este tipo de robô e suprir as necessidades que surgem relacionadas a transporte de cargas, de pessoas e também de realizar tarefas de investigação onde o ambiente impossibilita a ação do ser humano, seja em outros planetas (WELCH; LIMONADI; MANNING, 2013), ambientes radioativos (KIM et al., 2006) ou no fundo do mar (PSARROS et al., 2009).

Sua utilização partiu do início da década de 50, com um veículo usado para reboque de cargas desenvolvido pela *Barrett Electronics*. Arthur Barrett, fundador da empresa de mesmo nome, era reconhecido por sua paixão pelo desenvolvimento de produtos e pela inovação. No ano de 1954 foi ao mercado o que é hoje reconhecido como o primeiro AGV, o "*Guide-O-Matic*". Esse veículo navegava rastreando o sinal de fios montados inicialmente no teto da fábrica, passando depois a serem montados no chão. Na época, o que era chamado de "veículo sem motorista" revolucionou a maneira de transporte dentro das indústrias. A partir de então, sua participação no interior de estoques e de fábricas cresceu juntamente com suas funções nesses ambientes.

Inicialmente propostos para o transporte de pequenas cargas, foi sendo notada a possibilidade de utilização em diversas outras tarefas expandindo assim sua participação na indústria e especializando suas habilidades. Atualmente, podem ser encontrados veículos de reboque, transporte de cargas unitárias - equipados com decks que facilitam a remoção e transferência da carga -, transporte de pallets, empilhadeiras e até mesmo veículos híbridos: empilhadeiras que podem funcionar tanto automaticamente como sendo operadas por um humano. Em relação a essas habilidades, os AGVs se adequam a tarefas com as seguintes características:

- Movimento repetitivo de materiais dentro de um espaço;
- Entrega regular de cargas estáveis;
- Quando pontualidade na entrega da carga é essencial e atrasos causam ineficiência;
- Operação com ao menos dois turnos;
- Processos onde o rastreamento do material é importante.

Dentre os tipos de indústrias onde o uso de AGVs se destaca estão as indústrias farmacêutica, química, manufatureira, automotiva, papelaria e de impressões, alimentícia, hospitalar e de armazenamento. Áreas onde o transporte eficiente e de baixo custo é essencial para a produção.

O constante aumento na utilização desses equipamentos gerou um novo mercado, empresas desenvolvedoras de AGVs já são reconhecidas como uma nova grande oportunidade. Indústrias conservadoras como a automotiva e a indústria de alimentos já incorporaram em seus ambientes essas máquinas, isso é visto como um sinal que o AGV já é uma realidade comprovada para diminuição de custos dentro da manufatura.

Porém sua presença não se limita ao ambiente industrial, veículos guiados automaticamente são alvos de pesquisas inovadoras para aprimorar seus comportamentos e maneiras de interagir com o ambiente. Diversos protótipos desenvolvidos por universidades (LUETTEL; HIMMELSBACH; WUENSCHÉ, 2012) e também por empresas visam aprimorar as técnicas de operação destes equipamentos.

A empresa *Seegrid*, localizada na cidade de Pittsburgh, Estados Unidos, é uma empresa fundada a partir do laboratório *Mobile Robot Lab*. da Universidade Carnegie Mellon (CMU) da mesma cidade. Essa empresa é líder mundial em AGVs guiados por visão e com um rendimento que aumenta 400 por cento ao ano. Já a empresa *iRobot* foi fundada a partir do *Massachusetts Institute of Technology* (MIT) com a visão de tornar robôs práticos uma realidade. Sua presença nos lares americanos já é grande - com o robô *Roomba* - e continua se expandindo, atingindo novos mercados, inclusive o brasileiro. A *National Aeronautics and Space Administration* (NASA) desenvolveu o robô *Curiosity* para explorar a superfície de Marte. Todos esses são exemplos de robôs auto guiados de sucesso, embora possam não levar o nome de AGV, a tecnologia necessária para navegação em uma fábrica é análoga à navegação em outros ambientes.

No Brasil, existem grupos de pesquisa relacionados ao desenvolvimento de veículos autônomos. Um carro autônomo desenvolvido na Universidade de Minas Gerais (UFMG) é consequência de pesquisas como (SANTOS, 2009) e (LIMA, 2010). A Universidade de São Paulo (USP), através do Laboratório de Robótica Móvel desenvolveu o projeto CaRINA, gerando produções como (SAITO; SABATINE, 2009) e (HATA et al., 2009). Existem ainda outros projetos brasileiros, como o projeto Carro Autônomo da Universidade Federal do Espírito Santo, a Equipe do Veículo Autônomo da Escola Politécnica da Universidade de São Paulo e o projeto Sistema Embarcado de Navegação Autônoma (SENA) do Laboratório de Mecatrônica do Departamento de Engenharia Mecânica da Escola de Engenharia de São Carlos (EESC-USP) que visa o desenvolvimento de um veículo de passeio capaz de evitar acidentes, assistindo o motorista em situações de risco iminente; ampliar a capacidade cognitiva do motorista auxiliando-o de forma cooperativa a melhorar sua capacidade e qualidade de direção e em casos extremos, movimentar o

veículo de forma autônoma (sozinho) em ambientes urbanos.

Institutos como o DARPA (*Defense Advanced Research Projects Agency*) organizam desafios para estimular o crescimento de veículos autônomos, aumentando cada vez mais a complexidade do ambiente em que eles estão inseridos. O que começou em ambientes industriais com uma tarefa específica e uma grande limitação no espaço em que se localizam, desenrolou-se para a operação em ambientes inóspitos, complexos e atualmente existem veículos capazes de se deslocar juntamente com o trânsito de uma cidade. A empresa *Google* recentemente apresentou ao público seu último protótipo de carro autônomo, um carro onde não existe um lugar para o motorista. Os usuários devem indicar ao sistema aonde necessitam ir, e o carro irá de maneira automática. Essas invenções só foram possíveis através da visão de que essas tarefas podem ser exercidas automaticamente, a fim de deixar o processo criativo para o homem.

Para possibilitar a navegação autônoma de um robô, diversos sistemas são necessários. A fim de realizar sua localização no espaço em que se encontra, é necessário um sistema de sensoriamento. Esse sistema é o responsável por extrair a informação de onde o robô se localiza dentro do ambiente através dos mais variados tipos de sensores. Os primeiros AGVs, utilizavam um fio para indicar a rota do robô. Localizado no assoalho do percurso, esse fio transmitia um sinal de rádio que o AGV deveria seguir (SICILIANO; KHATIB, 2008). Essa técnica mostrou pouca flexibilidade caso exista a necessidade de mudar o percurso do AGV, já que grande tempo é necessário para fazê-lo. Ao passar dos anos, outras técnicas surgiram para deixar o trajeto mais flexível, um exemplo é a navegação por fita, onde uma fita magnética ou colorida é posta no chão para o AGV seguir.

Outro tipo é a localização baseada em laser. Ela é feita ao montar fitas reflexivas em paredes, máquinas ou postes, enquanto o AGV carrega um transmissor laser rotativo, que recebe o sinal refletido nas fitas para se localizar. O Sistema de Posicionamento Global (GPS) (ZHANG et al., 2008) é outra alternativa para localização em veículos autônomos. Navegação inercial também é utilizada em AGVs (WANG, 2006). Transponders são fixados no chão para atualizar a localização do veículo, e um giroscópio detecta mudanças na direção do mesmo, fazendo com que ele se desloque com precisão. Entre outros tipos de navegação é possível destacar a navegação visual, onde um sensor visual é combinado com o AGV a fim de identificar padrões para realizar a localização. Sistemas com Navegação Guiada por Visão também são chamados de sistemas de Servovisão (DUROLA et al., 2009) e tem recebido atenção especial devido a sua ampla gama de possibilidades. Ainda é comum fusão de algumas dessas técnicas, para aumentar a confiabilidade do sistema.

Outro fator importante na navegação autônoma do veículo é a estratégia de controle usada para definir sua trajetória. Dependendo do estilo, dos requisitos do ambiente e do próprio robô, sua modelagem matemática pode se tornar complexa, fazendo com que

uma estratégia de controle avançada seja necessária para estabilizar o sistema com mais rapidez e robustez. Rodas são o meio de locomoção mais comum para robôs industriais e aplicações que exigem velocidade, porém outros tipos também são utilizados, como esteiras - para aumentar a tração e manobrabilidade -, ou mecanismos que imitam pernas - para possibilitar uma agilidade maior do veículo e locomoção em ambientes diversificados. Essa característica influencia diretamente na movimentação e nos graus de liberdade do veículo, fazendo com que seu modelo matemático se torne mais complexo. No entanto, qualquer robô que tenha um deslocamento suave ao longo de um grau de liberdade, pode ter esse eixo considerado um sistema integrador duplo (SA; CORKE, 2012), facilitando a identificação de parâmetros do sistema e possibilitando a implementação de um controlador específico para esse tipo de sistema, o *Proximate Time Optimal Servomechanism* (PTOS).

Independentemente da aplicação, navegação autônoma é um problema complexo do campo da robótica aplicada, se estendendo por diversos campos da engenharia, incluindo controle de sistemas, processamento de sinais, visão computacional, ciência da computação e engenharia mecânica. Ao aumentar o nível de complexidade do ambiente ainda há o complicador de detalhes e situações inesperadas que o robô poderá enfrentar, fazendo-se necessário sistemas altamente robustos contra distúrbios e falhas.

Este trabalho tem como objetivo o desenvolvimento do protótipo de um robô guiado automaticamente, levando em consideração sua modelagem matemática, sensoria-mento e controle da trajetória através da técnica PTOS. Como mencionado anteriormente, esse tipo de robô tem grande aplicação na indústria e se mostra um mercado altamente promissor. Não somente isso, o desenvolvimento de novas técnicas para esse tipo de robô é essencial para o crescimento da pesquisa brasileira na área, visto que a tecnologia presente no mercado nacional é em sua maioria importada, evidenciando que, apesar dos exemplos citados, a pesquisa e desenvolvimento nesse campo em âmbito nacional ainda é escassa.

O trabalho está organizado em sete capítulos, sendo este o capítulo introdutório. O seguinte capítulo apresenta os conhecimentos básicos necessários para o acompanhamento do restante do trabalho, bem como uma revisão bibliográfica situando o leitor no estado em que se encontrar as pesquisas sobre o assunto. O Capítulo 3 desenvolve o modelo matemático necessário para o desenvolvimento e controle do robô, especificando as ferramentas utilizadas para tal. Seguindo para o capítulo seguinte, são apresentadas as técnicas de sensoriamento utilizadas para a extração dos dados necessários do sistema, juntamente com o filtro que realiza a fusão destes sensores. O quinto capítulo mostra as técnicas de controle utilizadas para o cálculo de trajetória do robô, justificando a escolha do controlador de tempo ótimo *PTOS* e utilizando uma comparação com controladores mais simples. O sexto capítulo apresenta os resultados obtidos no trabalho e discussões relevantes sobre o tema. O trabalho é então encerrado com um capítulo de conclusão.

2 Preliminares

Este capítulo apresentará ao leitor o sistema base usado no trabalho e conhecimentos prévios necessários para o acompanhamento do mesmo. O sistema utilizado e seu funcionamento são explicados em detalhes, os softwares desenvolvidos para a pesquisa são igualmente apresentados e descritos neste capítulo. Conhecimentos como representações de rotação de corpos, servovisão e controladores de tempo ótimo também são expostos aqui.

2.1 Idealização do Sistema Omni-Direcional

O sistema que será trabalhado é o robô Omni-Direcional. Um chassi robótico composto de um corpo de acrílico, quatro rodas Mecanum acionadas por motores de corrente contínua e um sistema de processamento embarcado envolvendo sensores inerciais e a comunicação com o computador compõem o dispositivo. Será usado como base um protótipo Omni-Direcional, com um sistema eletrônico modificado para melhor interação com o sistema de controle, que será feito a partir do software Matlab R2011b. A comunicação entre as duas partes será feita usando o módulo comercial de comunicação sem fio XBee.

2.1.1 A Roda Mecanum

Desenvolvidas pelo sueco Ilon (1975) e também chamada de rodas Ilon, as rodas Mecanum diferem das convencionais devido a direção da tração exercida sobre o solo. O deslocamento gerado não é perpendicular ao eixo do sentido de rotação das rodas, fazendo

Figura 1 – Detalhe da roda Mecanum.



Fonte: o Autor.

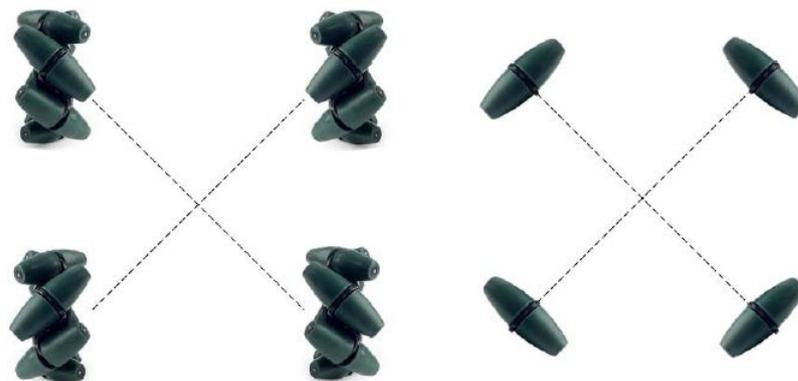
com que uma combinação desses dispositivos permita ao veículo mover-se em mais de um sentido apenas usando a combinação de diferentes rotações em cada roda. Ela é adequada a espaços reduzidos devido a manobrabilidade que propicia ao veículo, por essa razão, é utilizada em máquinas transportadoras dentro de indústrias onde esse requisito existe. Veículos como o *KUKA omniMove*[®] e o *Airtrax Cobra*[®] já fazem uso desta tecnologia dentro das indústrias suprimindo o mercado de transportes de cargas industriais de maneira inovadora e reduzindo custos de logística.

A roda é composta de um disco envolvido por pequenos roletes ao longo de sua circunferência, conforme visto na Fig. 1. Esses roletes tem seu eixo de rotação deslocado em relação ao eixo de rotação da roda. Isso faz com que a tração exercida por esta roda não seja perpendicular ao seu próprio eixo, mas sim deslocada em α° dele. Para o funcionamento adequado do sistema, duas das rodas devem ter ângulo α suplementar ao das outras duas e os pares de mesmo α devem ser dispostos nas suas diagonais, conforme detalhado na Fig. 2.

A combinação de quatro rodas montadas em um chassi retangular possibilita um movimento isolado em cada um dos dois sentidos de um plano XY bem como a rotação ao longo de seu próprio eixo. Essa característica, vista na Fig. 3, caracteriza um veículo holonômico, ou seja, que não possui restrição em qualquer tipo de movimento (SICILIANO; KHATIB, 2008), justificando então seu nome de veículo Omni-Direcional. Dada a disposição das rodas no chassi e levando em consideração a velocidade e sentido de rotação de cada uma delas, é possível determinar a força resultante em cima do corpo do veículo.

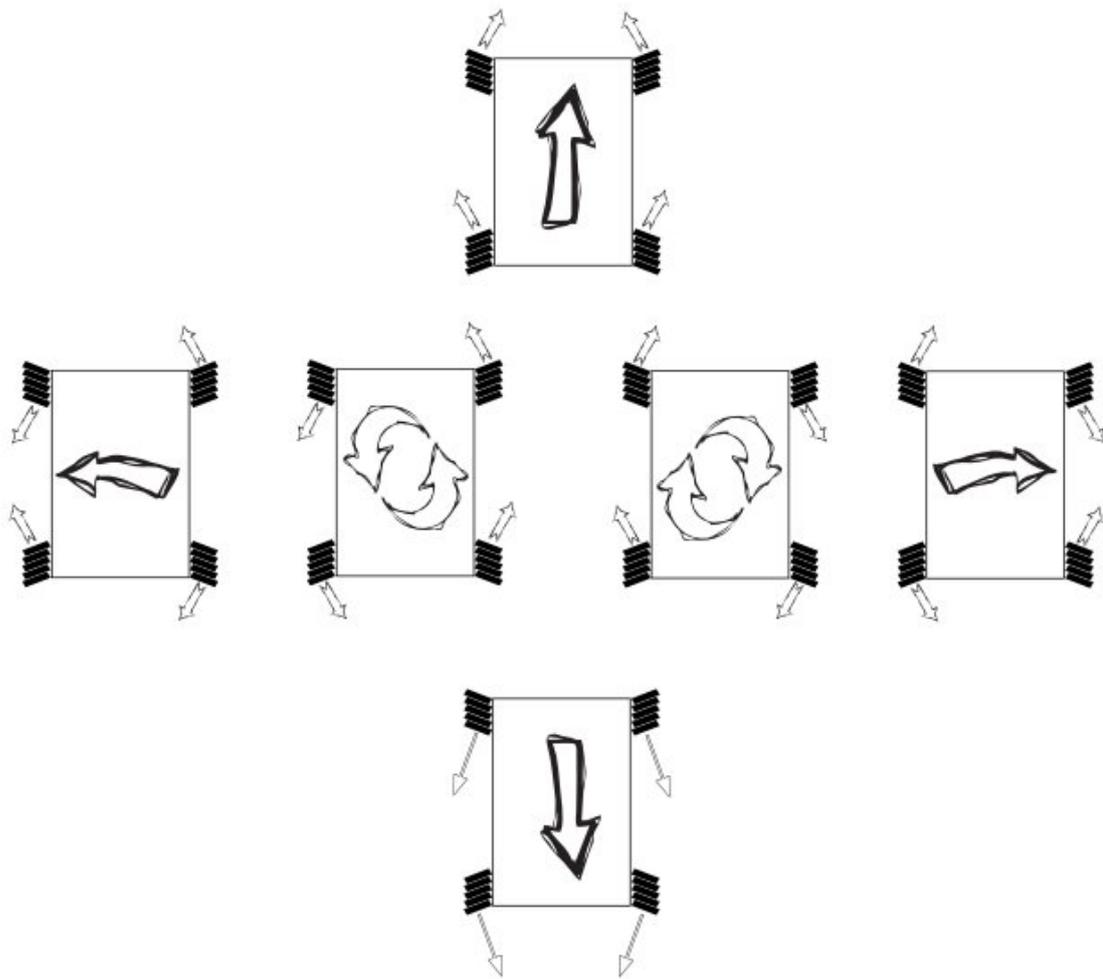
Para obter uma força resultante para frente, movimenta-se todas as rodas para a frente. De maneira análoga, a movimentação para trás se dá ao rotacionar as rodas para trás. Os movimentos laterais são atingidos com uma combinação diferente das forças exercidas por cada roda, ao inverter o sentido de um dos pares diagonais o veículo

Figura 2 – Esquema de montagem das rodas.



Fonte: o Autor.

Figura 3 – Direções do veículo Omni-Direcional.



Fonte: o Autor.

movimentará-se para um dos lados, conforme visto na Fig. 3. Ainda é possível realizar movimentos de rotação ao rotacionar as rodas de um lado em um sentido, e as do lado oposto no sentido oposto, realizando um giro com centro de rotação no centro do veículo.

Considerando a roda superior direita a roda w_1 , a superior esquerda a roda w_2 , a roda inferior esquerda a roda w_3 e a roda inferior direita a roda w_4 , é possível então demonstrar um conjunto de equações que relacionam cada roda com sua influência em cada um dos eixos x , y e θ , sendo o último a rotação sobre seu próprio eixo. Esse conjunto

de equações forma a cinemática do sistema, dada por:

$$\begin{cases} u_1 = -x + y + \theta \\ u_2 = x + y - \theta \\ u_3 = -x + y - \theta \\ u_4 = x + y + \theta \end{cases} \quad (1)$$

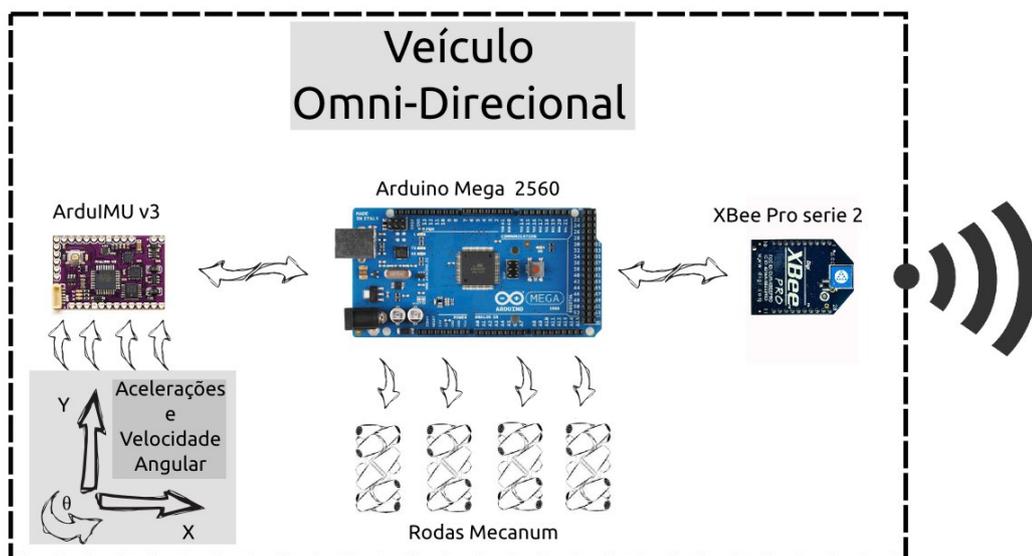
2.1.2 Hardware Utilizado

Além das quatro rodas montadas na plataforma de acrílico, o protótipo possui um sistema de processamento embarcado, responsável pela aquisição de dados inerciais e da comunicação com o computador que executa o Sistema Supervisório. Para a movimentação, quatro servomotores modificados são conectados as rodas e controlados pelo microprocessador.

Estes quatro motores recebem um sinal modulado por largura de pulso (do inglês, *Pulse Width Modulation - PWM*) proveniente do microprocessador. O dispositivo usado para processamento das informações é a plataforma de prototipagem eletrônica *open-source* chamada Arduino. O modelo utilizado, Arduino Mega, possui um microprocessador ATmega2560 que contém 54 pinos digitais de entrada e saída no qual 16 podem ser usados para modulação por largura de pulso. Este processador ainda possui quatro portas *Universal Asynchronous Receiver Transmitter (UART)* e opera em frequência de 16MHz.

Uma das portas de comunicação será usada com outro dispositivo embarcado no

Figura 4 – Esquemático dos dispositivos do veículo Omni-Direcional.



Fonte: o Autor.

veículo. Uma Unidade de Medida Inercial (do inglês, *Inertial Measurement Unit - IMU*) está presente a fim de captar as acelerações em que o sistema está sujeito. A *ArduIMU*, a *IMU* escolhida, possui um acelerômetro de três eixos, além de um giroscópio de três eixos e também um magnetômetro. Ela foi desenvolvida com base no sistema *open-source* Arduino para uso em Veículos Aéreos Não Tripulados (VANTs) e já possui comunicação serial facilitada para outras plataformas Arduino, fazendo com que seu uso em conjunto com o sistema seja direto.

Um módulo *XBee* de comunicação serial sem fio será utilizado para realizar a troca de informações com o computador que controlará o sistema de maneira remota. Esse módulo, através do microprocessador embarcado, envia as informações ao computador e recebe suas respostas, utilizando para tal uma comunicação serial bi-lateral. Um esquemático mostrando o funcionamento de todos os hardwares é mostrado na Fig. 4, onde o processador se comunica via serial com os sensores da *ArduIMU*, atua sobre as quatro rodas do veículo e troca mensagens com o computador através do módulo serial *XBee*.

O Sistema Supervisório fica localizado em um computador remoto que através de uma webcam, capta imagens do espaço de trabalho do veículo e por meio do software Matlab identifica dados como posição e velocidade do veículo e gerencia seu deslocamento. Este sistema se comunica com um módulo *XBee* conectado ao computador, enviando as mensagens de controle para o veículo.

2.1.3 Software Desenvolvido

O software responsável pelas ações do veículo é executado diretamente no microprocessador do sistema Arduino. Ele foi desenvolvido usando a linguagem própria para o sistema, através do Ambiente de Desenvolvimento Integrado (do inglês, *Integrated Development Environment - IDE*) do Arduino, visto em detalhe, na Fig. 5.

A função deste programa é receber a informação vinda do Sistema Supervisório, processá-la e comandar cada uma das quatro rodas de acordo com as mensagens recebidas. Ainda é atribuição do software receber a informação das medidas de aceleração e velocidade radial provenientes da *IMU* e repassá-las ao computador.

Neste nível de processamento se localizam somente instruções básicas de comando, estimativas de posição e velocidade são feitas somente no Sistema Supervisório. Caso o veículo detecte uma queda na comunicação ou experimente mais de dois segundos sem comunicação ele automaticamente para suas rodas a espera de que se restabeleça o contato com o computador.

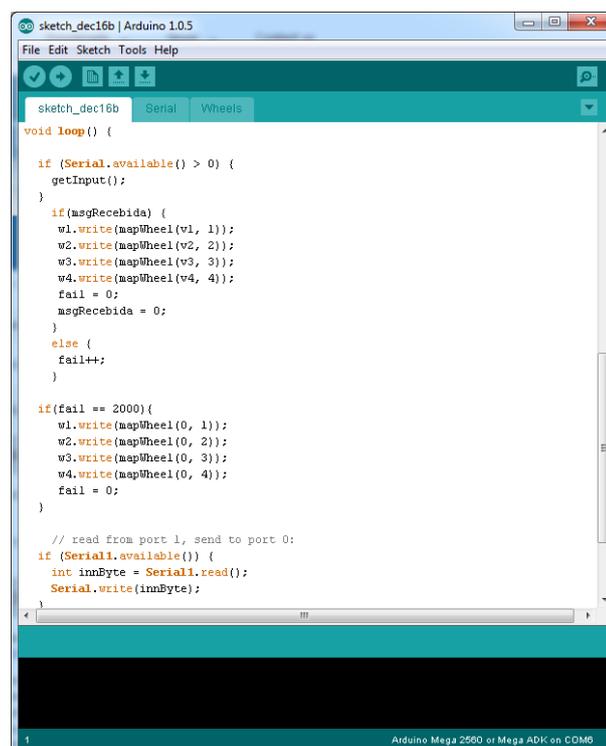
A execução do software, descrita pelo esquema mostrado na Fig. 6 se dá da seguinte maneira: Primeiramente, é executada uma rotina de inicialização, onde a comunicação

com o computador é estabelecida, o contato com a *ArduIMU* e com os motores é feito. Os motores são inicializados em posição inerte por segurança. Após estabelecido o contato com o programa central, o software espera até receber a primeira mensagem indicando o valor da velocidade a ser aplicada em cada roda, que por segurança, inicialmente também é zero. O próximo passo é atualizar o sinal de velocidade de cada roda e verificar por falhas no sistema. Caso tenha acontecido um erro na comunicação uma só vez, o programa não atualizará a velocidade das rodas. Caso o programa esteja a dois segundos ou mais sem receber mensagens do computador, ele parará o veículo. Por fim, o programa recebe a informação proveniente da IMU e a repassa ao computador, após este passo, ele torna a esperar a informação vinda do Sistema Supervisor.

2.1.4 Sistema Supervisor

O sistema encarregado de obter a posição do veículo através da imagem e sua aceleração através da IMU, calcular sua trajetória ótima e enviar ao veículo os comandos é chamado de Sistema Supervisor. Ele foi desenvolvido em sua íntegra no software Matlab R2011b pois este já facilita a comunicação com outros dispositivos necessários para o funcionamento do sistema, como o sensor visual e o módulo de comunicação sem fio, além de ser adequado a cálculos matemáticos complexos como por exemplo os cálculos necessários para processamento da imagem, filtragem dos sinais dos sensores e execução

Figura 5 – Ambiente de Desenvolvimento Integrado - Arduino



```
sketch_dec16b | Arduino 1.0.5
File Edit Sketch Tools Help
sketch_dec16b Serial Wheels
void loop() {
  if (Serial.available() > 0) {
    getInput();
  }
  if(msgRecebida) {
    w1.write(mapWheel(v1, 1));
    w2.write(mapWheel(v2, 2));
    w3.write(mapWheel(v3, 3));
    w4.write(mapWheel(v4, 4));
    fail = 0;
    msgRecebida = 0;
  }
  else {
    fail++;
  }
  if(fail == 2000){
    w1.write(mapWheel(0, 1));
    w2.write(mapWheel(0, 2));
    w3.write(mapWheel(0, 3));
    w4.write(mapWheel(0, 4));
    fail = 0;
  }
  // read from port 1, send to port 0:
  if (Serial1.available()) {
    int innByte = Serial1.read();
    Serial.write(innByte);
  }
}
```

Fonte: o Autor.

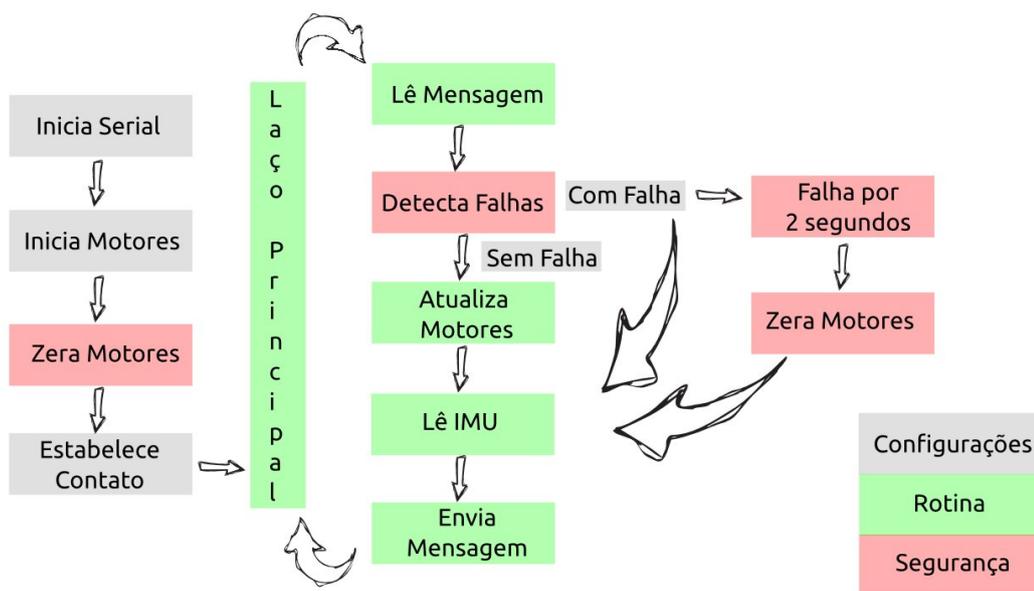
do sistema de controle.

Este programa inicia com uma etapa de configuração, onde os sistemas auxiliares são inicializados e ajustados para o correto andamento. Aqui a câmera localizada acima do veículo é configurada e sua janela de operação é estabelecida. A comunicação com o módulo XBee também é inicializada. Em seguida, uma mensagem inicial é enviada ao veículo para estabelecer contato e posteriormente, o laço de funcionamento começa a ser executado.

Neste laço é onde o veículo efetivamente será movimentado, uma referência de posição é pré-estabelecida e o sistema executa a rotina de sensoriamento para obter a posição, velocidade e aceleração do veículo. As informações de posição são extraídas da imagem gerada pela câmera e os dados de aceleração do veículo são enviados pelo sistema de comunicação sem fio. Um filtro é encarregado de mesclar as informações atuais com as anteriores e extrair dados confiáveis de posição e velocidade necessários para o controle do veículo. Caso um dos dois sensores falhe, o filtro ainda leva em consideração uma estimativa do modelo matemático do sistema para extrapolar a informação ausente.

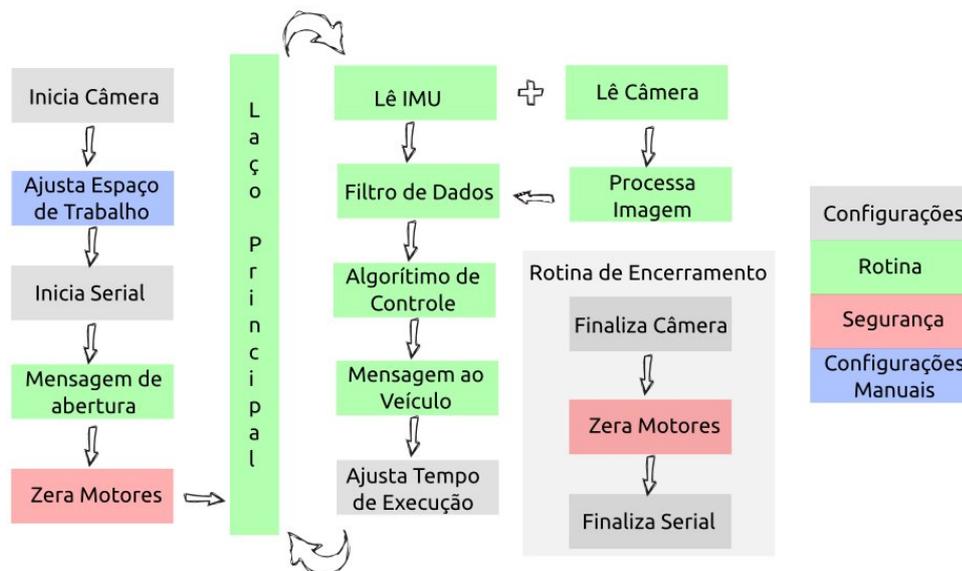
Os dados de posição e velocidade são então usados em um sistema de controle para computar as novas informações de velocidade para cada uma das rodas, a fim de que o sistema alcance a posição desejada. Isso é feito iterativamente por um período determinado pelo usuário. Ao finalizar o laço, o programa ainda executa uma rotina de encerramento. Esse algoritmo é detalhado na Fig. 7, informações mais detalhadas sobre as rotinas específicas de sensoriamento e controle serão abordadas nos capítulos subsequentes.

Figura 6 – Algoritmo de funcionamento do software do veículo.



Fonte: o Autor.

Figura 7 – Algoritmo de funcionamento do software supervisorio.



Fonte: o Autor.

2.1.5 Funcionamento Completo

O sistema como um todo tem sua rotina de execução baseada na comunicação do computador com o veículo, este encarregado de receber as mensagens e realizar o movimento solicitado, e o computador encarregado de obter a posição através do sensor visual, calcular a trajetória desejada e enviar essa informação ao veículo.

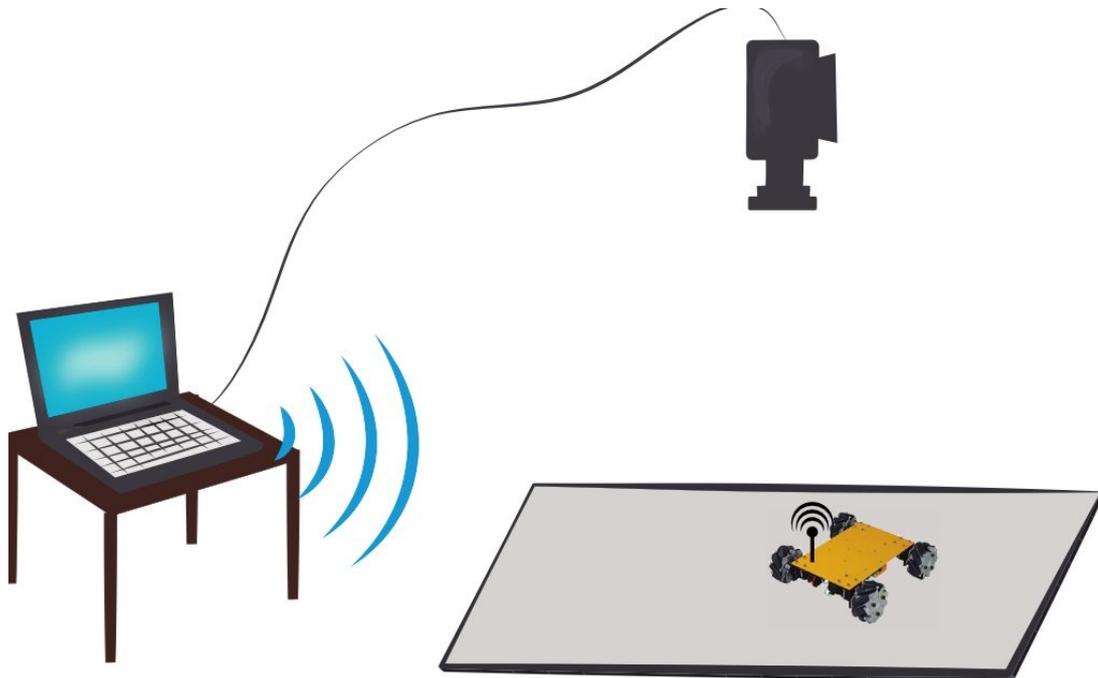
Uma webcam, atuando como sensor de visão, capta a imagem de um retângulo projetado no solo onde o veículo atua conforme a Fig. 8, e a envia ao computador, que realiza um processamento a fim de extrair a posição do veículo. Ele também recebe diretamente do veículo a informação das acelerações a que este está sujeito. Após realizada a união desses dados, um algoritmo de controle gera o sinal de velocidade a cada uma das rodas, e envia novamente ao veículo que deve mover-se até o ponto desejado. O ciclo de funcionamento tem duração estipulada pelo usuário, bem como as referências desejadas.

2.2 Representação de Rotação de Corpos

A cinemática espacial pode ser vista como o estudo sobre as diferentes formas de representar a posição de um corpo no espaço. Translações e rotações são usadas em conjunto para expressar essas representações. Um corpo pode ter sua posição expressada em três variáveis indicando o deslocamento ao longo de um espaço tridimensional e em três outras variáveis que indicam a rotação ao longo de cada um desses eixos.

O número mínimo de coordenadas necessárias para representar um corpo no es-

Figura 8 – Esquemático completo do veículo, computador e câmera.



Fonte: o Autor.

paço Euclidiano é seis: sendo elas as três para translações no espaço, e as outras três para rotações no mesmo. Porém algumas representações empregam um número maior de variáveis devido a existência de relações auxiliares entre as coordenadas. Essas variáveis excedentes definem o número de relações auxiliares independentes em um conjunto de coordenadas.

Um sistema de coordenadas i consiste em um ponto de origem O_i e em uma tríade de vetores mutuamente ortogonais denominados $(\hat{x}_i, \hat{y}_i, \hat{z}_i)$. Eles podem ser fixos no objeto de estudo ou em um referencial pré-estabelecido. A pose do corpo a ser representado pode sempre ser expressa com um sistema de coordenadas deslocado em relação a um sistema de origem.

2.2.1 Translação

Translação é o deslocamento no qual nenhum ponto de um corpo permanece na sua posição original e onde todas as linhas retas deste corpo permanecem paralelas as suas orientações iniciais. A translação de um corpo no espaço pode ser representada pela combinação das suas posições antes e após o deslocamento. De maneira análoga, a posição futura de um corpo pode ser representada como uma translação a partir de sua posição inicial. Portanto, qualquer representação da posição pode ser usada para representar um deslocamento, e vice-versa.

Para representar o deslocamento de um corpo até a posição j desde a origem O_i é

possível denominar um vetor 3×1 de posições tal que:

$${}^j p_i = \begin{pmatrix} {}^j p_i^x \\ {}^j p_i^y \\ {}^j p_i^z \end{pmatrix} \quad (2)$$

Onde os componentes desses vetores são as coordenadas cartesianas de O_i no sistema onde a origem se encontra em O_j , ou seja, são as projeções do vetor ${}^j p_i$ nos eixos correspondentes. As componentes desse vetor também poderiam ser expressas em coordenadas polares ou cilíndricas, dependendo da exigência do sistema. Tais representações têm vantagens para aplicação em braços robóticos cujos mecanismos utilizam juntas cilíndricas ou esféricas.

2.2.2 Rotação de Corpos

Uma rotação é o deslocamento no qual ao menos um ponto do corpo permanece na sua posição inicial e nem todas as linhas retas deste corpo permanecem paralelas as suas orientações iniciais. Por exemplo, um corpo em uma órbita circular rotaciona sobre um eixo no centro do seu trajeto e todos os pontos em cima deste eixo de rotação são pontos que permanecem em suas posições originais. Como no caso de posição e translação, qualquer representação de orientação pode ser usada para criar uma representação de rotação e vice-versa.

Dado que diferentes maneiras de representar uma rotação existem e embora não haja uma abordagem ótima para todos propósitos, as vantagens de cada uma delas podem ser usadas para a resolução de diferentes problemas. Os tipos mais utilizados na robótica em geral serão apresentados, sendo dado um destaque maior a representação através de Quatérnions, muito utilizada na navegação aeroespacial e por consequência disso, também muito utilizada em veículos robóticos auto guiados (ZHOU; BOLANDHEMMAT, 2007), (KOENIG; DONG; DESOUZA, 2008).

2.2.2.1 Matriz de Rotação

A orientação de um sistema de coordenadas i em relação a um sistema j pode ser expressa usando os vetores base $(\hat{x}_i \ \hat{y}_i \ \hat{z}_i)$ em termos dos vetores $(\hat{x}_j \ \hat{y}_j \ \hat{z}_j)$. Levando assim a $({}^j \hat{x}_i \ {}^j \hat{y}_i \ {}^j \hat{z}_i)$ que quando escrito como uma matriz 3×3 é conhecido como Matriz de Rotação. Os componentes da matriz ${}^j R_i$ serão os produtos vetoriais dos vetores base dos dois sistemas de coordenadas.

$${}^j R_i = \begin{pmatrix} \hat{x}_i \cdot \hat{x}_j & \hat{y}_i \cdot \hat{x}_j & \hat{z}_i \cdot \hat{x}_j \\ \hat{x}_i \cdot \hat{y}_j & \hat{y}_i \cdot \hat{y}_j & \hat{z}_i \cdot \hat{y}_j \\ \hat{x}_i \cdot \hat{z}_j & \hat{y}_i \cdot \hat{z}_j & \hat{z}_i \cdot \hat{z}_j \end{pmatrix} \quad (3)$$

Visto que os vetores base são vetores unitários e o produto vetorial de quaisquer dois vetores unitários é o cosseno do ângulo entre eles, os componentes dessa matriz são geralmente chamados de cossenos diretores. Uma rotação do sistema i sobre o eixo \hat{z}_j de θ graus será então:

$$R_Z(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

Enquanto uma rotação do sistema i sobre o eixo \hat{y} será:

$$R_Y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix} \quad (5)$$

E sobre o eixo \hat{x} será:

$$R_X(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix} \quad (6)$$

As matrizes de rotação podem ser combinadas através de multiplicações de matrizes para representar rotações sucessivas em mais de um eixo. A rotação do sistema i até o sistema k , passando pelo sistema j , pode ser representada como:

$${}^k R_i = {}^k R_j R_i \quad (7)$$

Em resumo, ${}^j R_i$ é a matriz de rotação que transforma o vetor expresso no sistema de coordenadas i para o vetor expresso em j . Ela propicia uma representação da orientação i em relação a orientação j e portanto, pode ser considerada uma rotação de i até j .

2.2.2.2 Ângulos de Euler

Em uma representação minimalista da orientação, a rotação do sistema i até o sistema j pode ser representada por um vetor de três ângulos (α, β, γ) . Esses ângulos recebem o nome de ângulos de Euler quando cada um deles representa uma rotação sobre cada um dos eixos de um espaço cartesiano. Neste caso, o local de cada eixo depende da rotação anterior. A ordem das rotações nos eixos é de grande importância nessa representação, já que ela influencia a rotação seguinte. Tomando como exemplo os símbolos (α, β, γ) para representar uma rotação ao longo dos eixos Z , Y e X o ângulo α será o valor em que o eixo \hat{z} deve ser rotacionado. O ângulo β será o valor para o eixo \hat{y} que já

teve sua posição alterada pela rotação anterior e o ângulo γ será o valor para o eixo \hat{x} , que foi duplamente deslocado pelas duas rotações anteriores.

Contudo, a rotação por ângulos de Euler sempre exibirá uma singularidade quando a primeira e a última rotação acontecerem sobre o mesmo eixo. Essa singularidade é chamada de *gimbal lock* e ela representa a perda de um grau de liberdade quando dois dos eixos rotacionados ficam em uma posição paralela um ao outro. Uma maneira de evitar esse problema é o uso de quatérnions para a representação de rotações.

2.2.2.3 Ângulo e Eixo

Outra maneira de representar uma rotação é através de um único ângulo θ combinado com um vetor unitário \hat{w} . Neste caso, um sistema pode ser rotacionado de i até j em θ° e ao longo do eixo \hat{w} . Esse vetor pode ser referenciado também como o resultado de uma rotação ao longo de três eixos diferentes, sendo ele então o eixo equivalente desta rotação. Essa é uma representação super abundante pois tem mais de três variáveis representando três graus de liberdade. Ela dá origem a um outro tipo de representação, com o uso de quatérnions que de maneira análoga, utiliza quatro parâmetros, porém seu formato é modificado para facilitar a matemática envolvida na representação.

2.2.2.4 Representação em Quatérnions

Um quatérnion é um número complexo de quatro dimensões que pode ser usado para representar a rotação de um corpo em um espaço tri-dimensional. Uma orientação arbitrária do sistema j relativo ao sistema i pode ser alcançada através de uma rotação de ângulo θ em torno de um eixo ${}^i\hat{r}$ definido no sistema i . Isso é representado graficamente na Figura 9 onde os vetores unitários \hat{x}_i , \hat{y}_i e \hat{z}_i mutualmente ortogonais e os vetores unitários \hat{x}_j , \hat{y}_j e \hat{z}_j também mutualmente ortogonais definem os eixos dos sistemas de coordenadas i e j respectivamente. O quatérnion ${}^i_j\hat{q}$ que descreve a orientação do sistema i relativo ao sistema j é definido na equação (8) onde r_x , r_y e r_z definem as componentes do vetor unitário ${}^i\hat{r}$ nos eixos x , y e z do sistema i , e θ é a rotação em torno do eixo ${}^i\hat{r}$.

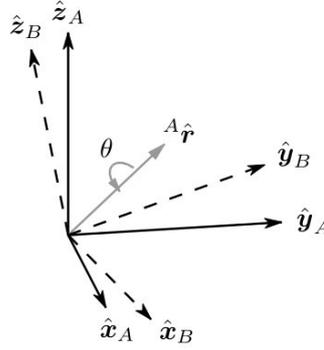
$${}^i_j\hat{q} = [q_1 \quad q_2 \quad q_3 \quad q_4] = \left[\cos \frac{\theta}{2} \quad -r_x \sin \frac{\theta}{2} \quad -r_y \sin \frac{\theta}{2} \quad -r_z \sin \frac{\theta}{2} \right] \quad (8)$$

A matemática envolvendo quatérnions geralmente precisa que o quatérnion esteja normalizado, portanto é convencional que os quatérnions que representam orientação sejam todos de tamanho unitário, conforme visto na equação (9).

$$q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1 \quad (9)$$

O conjugado de um quatérnion, representado por $*$, pode ser usado para trocar o sistema de referência descrito pela orientação. Por exemplo, ${}^j_i\hat{q}$ é o conjugado de ${}^i_j\hat{q}$ e

Figura 9 – Rotação entre um sistema de coordenadas e outro.



Fonte: o Autor.

descreve a orientação do sistema i relativo ao sistema j . O conjugado de ${}^i_j\hat{q}$ é definido pela equação (10).

$${}^i_j\hat{q}^* = {}^j_i\hat{q} = [q_1 \quad -q_2 \quad -q_3 \quad -q_4] \quad (10)$$

Assim, conforme a definição apresentada na equação (8), é possível criar uma matriz de rotação envolvendo quatérnions como mostrado em (COOKE et al., 1992). Essa matriz então é definida como:

$$Q = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_0q_2 + q_1q_3) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (11)$$

O uso de quatérnions para a rotação do modelo será devido ao custo computacional envolvido. Enquanto uma matriz de rotação possui em seu cálculo senos e cossenos, a matriz composta por quatérnions possui apenas a multiplicação entre seus estados, diminuindo o custo computacional.

2.3 Servovisão

Servovisão remete ao uso de processamento digital de imagens para controlar o movimento de um robô (CHAUMETTE; HUTCHINSON, 2006). A informação visual é adquirida por uma câmera que pode estar tanto no corpo do robô (chamado de eye-in-hand) como em uma posição externa observando seu espaço de trabalho. A servovisão consiste no uso de um sensor visual, como uma webcam, para adquirir uma imagem e identificar elementos nessa imagem que possibilitem estimar a posição do robô. No caso de uma câmara posicionada acima do veículo, a imagem consiste em uma foto do espaço de trabalho, e do corpo do robô. Os elementos na imagem podem variar de formas

geométricas tais como arestas, linhas, vértices e círculos (NAYAR; NENE; MURASE, 1995) até a estimativa do fluxo óptico e identificação de objetos através de visão estéreo. O sistema encarregado do controle irá processar a imagem para estimar a posição do robô e usar essa informação na estimativa do seu erro de posição para então atuar sobre os motores. O controlador usado para comandar o robô até a posição desejada pode variar de algoritmos de predição simples desenhados para alcançar maior eficiência computacional até controladores sofisticados, auto-ajustáveis e que levam em conta a dinâmica do sistema. Muitos destes métodos necessitam calibração prévia tanto dos parâmetros intrínsecos do sensor visual (como distância focal) como dos parâmetros extrínsecos (como rotação e translação em respeito ao manipulador).

2.4 Servomecanismos e Controladores de Tempo Ótimo

Nesta seção será apresentado um sistema servomecânico conhecido como integrador duplo. Um controlador desenvolvido para ele também será discutido explicitando seu desenvolvimento teórico e sua implementação prática, conhecida como *Proximate Time Optimal Servomechanism* (PTOS).

2.4.1 Servomecanismos

A planta integrador duplo é um dos sistemas fundamentais no estudo de controle de sistemas. Ela representa um sistema de um único grau de liberdade podendo ser tanto translacional como rotacional. Aplicações do integrador duplo incluem movimentação sem atrito e sem restrições, tais como a rotação de uma nave (HUGHES, 2012) e movimento de guias com braço giratório (GUSTAFSSON, 1996).

O sistema descrito em espaço de estados é:

$$\begin{cases} \dot{x} = Ax + Bsat(u) \\ y = Cx \end{cases} \quad (12)$$

onde

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{c}{M} \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} \quad (13)$$

onde os estados x_1 e x_2 são posição e velocidade, respectivamente, de um corpo de massa M e c é o atrito entre o corpo e a superfície de deslocamento.

Nota-se ainda que a entrada u do sistema está sujeita a função de saturação $sat(\cdot)$

definida como:

$$\text{sat}(u) = \begin{cases} u, & |u| < \epsilon \\ \epsilon \text{sign}(u), & |u| \geq \epsilon \end{cases} \quad (14)$$

e

$$\text{sign}(u) = \begin{cases} -1, & u < 0 \\ 0, & u = 0 \\ 1, & u > 0 \end{cases} \quad (15)$$

Assumindo ainda o uso de um compensador de atrito para eliminar o termo $\frac{c}{M}$ como em (ZHENG; FU, 2008), (SALTON et al., 2011), ou considerando o atrito c tão pequeno a ponto de ser desprezível, o sistema pode ser reescrito como:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = b \text{sat}(u) \end{cases} \quad (16)$$

sendo $b = \frac{1}{M}$. Esse sistema é então conhecido como integrador duplo.

2.4.2 Controlador de Tempo Ótimo

O controlador de tempo ótimo (do inglês, *Time Optimal Control* - TOC) busca obter a resposta mais rápida possível em um sistema integrador duplo reagindo a uma entrada do tipo degrau. Para isso, ele baseia-se na ideia de saturar a aceleração do sistema durante a metade do percurso e ao longo da segunda metade, desacelerar ao máximo o sistema.

Assim, o menor tempo para o sistema atingir sua referência se dará fazendo esta troca do sinal de controle entre o seu nível máximo e mínimo, conforme pode ser visto na Fig. 10.

Essa técnica ainda é conhecida como *bang-bang*, justamente pelo fato de ter somente duas ações extremas em sua operação.

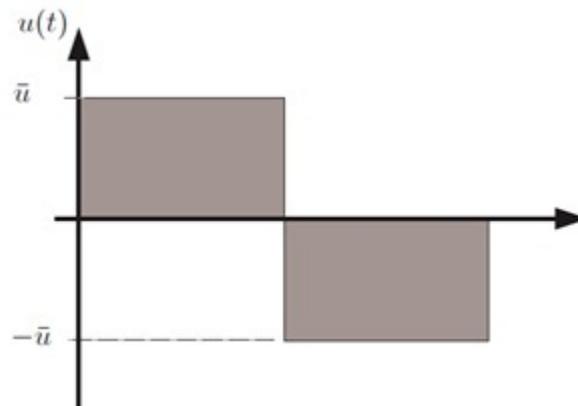
A lei de controle do TOC é definida como:

$$\begin{cases} u = -\bar{u} \text{sign}(\text{sign}(e) \sqrt{\frac{2\bar{u}|e|}{M}} + \dot{y}) \\ e = y - r \end{cases} \quad (17)$$

sendo \bar{u} o valor sem sinal máximo do sinal de controle u , e o parâmetro e representa o sinal de erro entre a referência r do sistema e a saída y .

Um sistema integrador duplo controlado pelo TOC atinge sua referência de posição no menor tempo permitido pelas suas leis. No entanto, sua implementação prática pode

Figura 10 – Sinal de controle do TOC.



Fonte: o Autor.

ser um problema devido ao efeito que se faz presente na etapa em que o sistema se encontra muito próximo a sua referência, chamado de *chattering*. Esse efeito consiste na troca abrupta entre os níveis máximo e mínimo do sinal de controle quando o sistema aproxima-se de sua referência, a variação do sinal de controle acaba gerando uma oscilação sustentável na saída, fazendo com que o sistema não atinja a referência com exatidão e podendo ocasionar falhas no atuador do sistema, como desgastes mecânicos e grandes perdas de calor nos circuitos eletrônicos (UTKIN; LEE, 2006). Essa oscilação muitas vezes é indesejada, fazendo com que a implementação prática do controlador TOC seja inviabilizada nessas situações. A fim de contornar esse problema, controladores de tempo quase ótimo surgiram, ou seja, leis de controle baseadas na teoria do TOC, porém com alterações que permitem a aplicação prática em um sistema real apareceram para resolver o efeito *chattering* criado pelo TOC.

A principal exigência para criar um controlador baseado no TOC que seja aplicável em sistemas reais é fazer com que a saída do sistema não assuma somente os dois níveis de saturação. Levando isto em consideração e expandindo a ideia da lei de controle TOC, foi proposto em 1987 o *Proximate Time Optimal Servomechanism*.

2.4.3 *Proximate Time Optimal Servomechanism*

O controlador proposto em (WORKMAN; KOSUT; FRANKLIN, 1987) é baseado na lei de controle TOC porém com modificações essenciais para viabilizar sua implementação prática. Este controlador, chamado de *Proximate Time Optimal Servomechanism* (PTOS) foi desenvolvido especificamente para o sistema integrador duplo. Essa é a grande vantagem do uso deste controlador se comparado com outras técnicas não específicas, porém ao experimentar mudanças na planta a ser controlada ele não garante um bom funcionamento, na medida em que um controlador mais abrangente se adaptará melhor a

uma mudança no sistema.

Ele é de grande importância dentro dessa classe de sistemas devido a alta performance obtida no tempo de acomodação e, principalmente, por considerar a saturação do sinal na sua lei de controle. Isso faz com que essa técnica não apresente dependência do sinal de referência no processo de ajuste dos ganhos do controlador, diferenciando ele de outros sistemas de controle como LQG, CNF, PID que possuem relação entre o valor de seus ganhos e o valor de saturação do sinal de controle.

A criação do PTOS então, teve como exigência que se mantivesse as características principais do controlador TOC porém buscando eliminar o efeito indesejado *chattering*. Ele é dividido em três passos para explicar seu funcionamento:

- Para evitar o *chattering* primeiramente deve-se fazer com que o sinal u de controle possa assumir outros valores além dos dois níveis de saturação. Para isso, elimina-se a função $sign(\cdot)$ de fora da equação (17) da lei de controle TOC, já que ela limita o sinal de controle nos dois estados citados. Além disso, se adiciona um parâmetro livre k para regular a entrada de controle, resultando em:

$$u = k(-f(x_1) - x_2), \quad (18)$$

onde

$$f(x_1) = sign(x_1)\sqrt{2b\bar{u}|x_1|} \quad (19)$$

sendo x_1 o erro de seguimento de referência, definido como $e = x_1 - r$.

Este primeiro passo introduz um alto ganho que satura o controlador e o faz convergir para a curva de troca - onde deve mudar o valor da máxima aceleração para o da máxima desaceleração - em tempo quase ótimo $x_2 = -f(x_1)$ com máxima aceleração. Contudo, ao alcançar a curva de troca, a entrada de controle permanecerá em zero, conforme ilustrado na Fig. 11. Deve-se então adicionar um termo para que o sinal de controle varie de \bar{u} não a 0, mas até $-\bar{u}$.

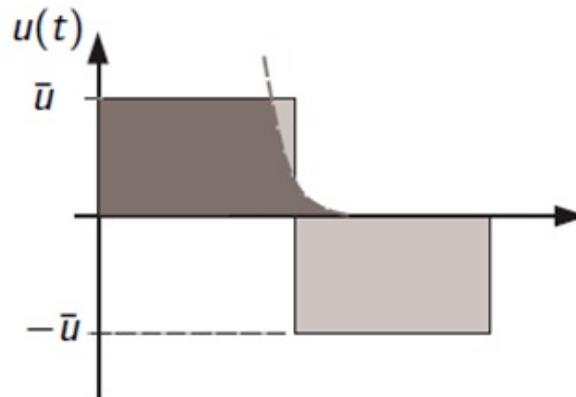
- O segundo passo é saturar negativamente o sinal do controlador. Para isso, adiciona-se um termo $sign(x_1)\bar{u}$ na função $f(x_1)$. O sinal então se comportará conforme visto na Fig. 12, faltando agora garantir o seguimento de referência do controlador. A lei do controlador resultará em:

$$u = k(-f_p(x_1) - x_2), \quad (20)$$

onde f_p é:

$$f(x_1) = sign(x_1)(\sqrt{2b\bar{u}|x_1|} - \frac{\bar{u}}{k}) \quad (21)$$

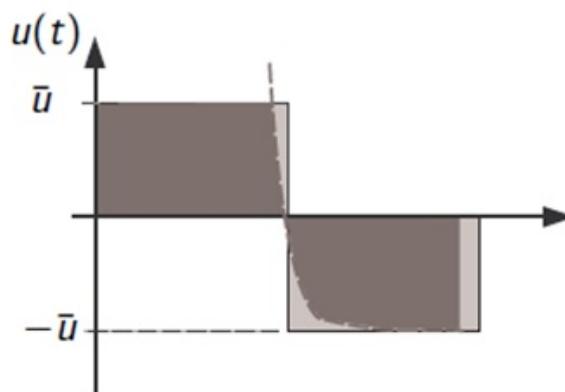
Figura 11 – Primeira etapa da lei de controle PTOS.



Fonte: o Autor.

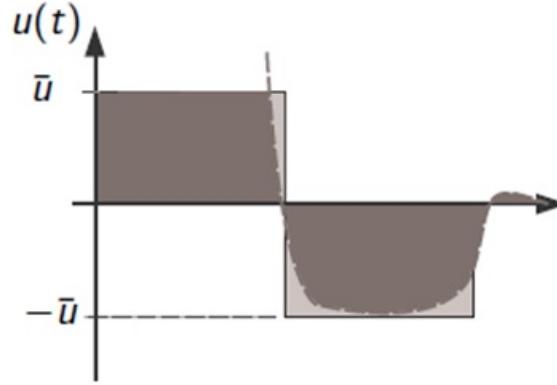
• Por fim, para garantir a estabilidade assintótica deve ser feita uma troca de função no controlador na medida em que o sistema se aproxima da sua referência, conforme visto na Fig. 13. Assim, a função vista na equação (21) será trocada para um controlador menos agressivo proporcional-derivativo (PD). Porém ao realizar essa troca, o resultado apresentará um pequeno sobressinal ao atingir a referência. Para solucionar esse problema, na função da equação (21) se adiciona um termo α chamado fator de desconto da aceleração. Este fator diminui a agressividade da lei de controle fazendo com que ele acelere por um tempo inferior ao tempo ideal, permitindo ao sistema de controle, quando feita a troca para a função PD, atingir a referência sem a presença do sobressinal, já que ele passará menos tempo acelerando e mais tempo desacelerando o sistema. Um problema gerado pela inclusão do coeficiente α é que, justamente por diminuir o tempo de aceleração, a performance do controlador não será ideal. Finalmente, a lei de controle do PTOS é dada

Figura 12 – Segunda etapa da lei de controle PTOS.



Fonte: o Autor.

Figura 13 – Terceira etapa da lei de controle PTOS.



Fonte: o Autor.

por:

$$u = k_2(-f_{ptos}(x_1) - x_2) \quad (22)$$

onde:

$$f_{ptos}(x_1) = \begin{cases} \frac{k_1}{k_2}x_1 & , |x_1| \leq y_l \\ \text{sgn}(x_1)(\sqrt{2b\alpha|x_1|} - \frac{\bar{u}}{k_2}) & , |x_1| > y_l \end{cases} \quad (23)$$

e os ganhos k_1 e k_2 são os ganhos proporcional e derivativo, respectivamente, decorrentes da adição do controlador PD.

A presença do fator de aceleração $0 < \alpha < 1$ permite acomodar incertezas na planta. Porém, o custo do aumento do fator de aceleração é o aumento do tempo de resposta.

Além disto, a continuidade da lei de controle é garantida somente quando

$$\begin{aligned} f_{ptos}^-(x_1) &= f_{ptos}^+(x_1) \\ \frac{df_{ptos}^-(x_1)}{dt} &= \frac{df_{ptos}^+(x_1)}{dt} \end{aligned}$$

onde $f_{ptos}^-(x_1)$ e $f_{ptos}^+(x_1)$ são as linhas superior e inferior, respectivamente, da lei mostrada na equação 23. Estas condições resultam nas restrições:

$$y_l = \frac{\bar{u}}{k_1} \quad (24)$$

$$k_2 = \sqrt{\frac{2k_1}{b}} \quad (25)$$

3 Modelo

O modelo matemático que descreve o sistema Omni-Direcional será apresentado neste capítulo. Seu desenvolvimento se dará primeiramente em coordenadas locais e será então estendido para coordenadas globais. Será apresentada ainda a identificação dos parâmetros físicos intrínsecos do sistema para alcançar uma representação em espaço de estados com sete variáveis que descrevem seu comportamento.

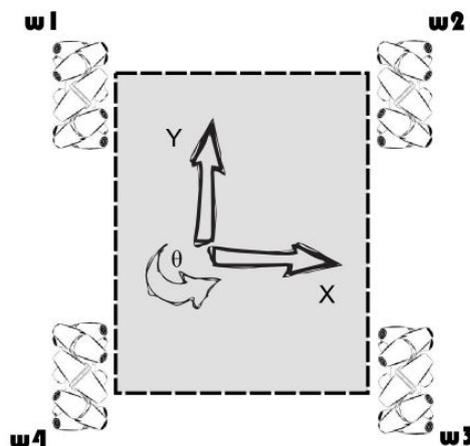
3.1 Coordenadas Locais

O veículo Omni-Direcional é composto de um chassi retangular com quatro rodas Mecanum conforme descrito na seção 2.1. Exemplificado aqui no esquema da Fig. 14, ele é descrito com suas rodas $w_1...w_4$ e o sistema de coordenadas fixo no seu corpo conforme apresentado.

Do esquema da Fig. 14 ainda é possível concluir que o sistema possui somente três graus de liberdade dentro do espaço Euclidiano, visto que ele não possui deslocamento ao longo do eixo Z e tampouco pode rotacionar em quaisquer um dos eixos X ou Y .

Assim, é possível estabelecer as primeiras três equações que descrevem as acelerações ao longo dos três graus de liberdade do sistema, sendo elas as acelerações em x , y e θ . De acordo com a equação da cinemática (1) apresentada na seção 2.1.1, e levando em

Figura 14 – Rodas e Sistema de Coordenadas Local do Veículo.



Fonte: o Autor.

consideração o ângulo α de cada uma das rodas, as equações serão:

$$\begin{aligned}\ddot{x}' &= k_x(-u_1 \cos \alpha + u_2 \cos \alpha - u_3 \cos \alpha + u_4 \cos \alpha) \\ \ddot{y}' &= k_y(u_1 \sin \alpha + u_2 \sin \alpha + u_3 \sin \alpha + u_4 \sin \alpha) \\ \ddot{\theta}' &= k_\theta(u_1 d - u_2 d - u_3 d + u_4 d)\end{aligned}\quad (26)$$

onde $u_1 \dots u_4$ são as entradas de controle em cada uma das rodas, e d é a distância euclidiana entre o comprimento e a largura do veículo.

Baseado neste conjunto de equações e definindo $c := \cos \alpha$ e $s := \sin \alpha$ é possível definir a matriz \bar{B} , incluindo ainda os parâmetros k_i , $i = x, y, \theta$, que relacionam as entradas $u_1 \dots u_4$ com as acelerações causadas pelos motores. Esse parâmetro será identificado experimentalmente nas próximas seções.

$$\bar{B} := \begin{bmatrix} k_x & 0 & 0 \\ 0 & k_y & 0 \\ 0 & 0 & k_\theta \end{bmatrix} \begin{bmatrix} -c & c & -c & c \\ s & s & s & s \\ d & -d & -d & d \end{bmatrix}\quad (27)$$

Considerando a matriz \bar{B} montada e incluindo os fatores de atrito c_i em cada um dos eixos, o espaço de estados que representa o sistema em coordenadas locais será:

$$\begin{bmatrix} \dot{x}' \\ \dot{y}' \\ \dot{\theta}' \end{bmatrix} = \begin{bmatrix} c_x \dot{x} \\ c_y \dot{y} \\ c_\theta \dot{\theta} \end{bmatrix} + \bar{B} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix}\quad (28)$$

Dado que esse modelo de equações apresentado não relaciona o estado de orientação com os estados de translação, nota-se que é impossível manter registro de sua posição completa em relação a uma origem, já que uma alteração na orientação do sistema altera o sentido dos eixos de translação. Para isso, é necessário uma influência do estado de orientação nos estados de translação. Essa ação pode ser alcançada com o uso de uma matriz de rotação que o sentido dos eixos de translação à sua origem.

3.2 Coordenadas Globais

Como o próprio nome propõe, um sistema em coordenadas globais representa todo o ambiente envolvido no estudo. Dentro de um sistema global, é possível existirem inúmeros sistemas locais, cada um com seu próprio eixo de coordenadas, porém é preciso sempre haver uma relação capaz de levar esse sistema local de volta à origem do sistema global. Para isso é utilizada uma matriz de rotação que leva o sistema local até as coordenadas globais.

O modelo apresentado na equação (28) é primeiramente rotacionado utilizando a matriz de rotação em quatérnions Q , usada para representar o sistema em coordenadas globais. Ela é definida pela matriz da equação (11) apresentada na seção 2.2.2.4, porém supondo uma rotação apenas no eixo Z . Supondo ainda um ponto $[x', y', \theta]$, essa matriz rotaciona θ° o ponto levando-o até $[x, y, \theta]$, conforme mostrado na equação (3.2).

$$Q = \begin{bmatrix} q_0^2 - q_3^2 & -2q_0q_3 & 0 \\ 2q_0q_3 & q_0^2 - q_3^2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} = Q \begin{bmatrix} x' \\ y' \\ \theta \end{bmatrix} \quad (29)$$

Nota-se que, como o sistema possui rotação apenas no eixo perpendicular ao plano onde o sistema se desloca, os termos do quatérnion q_1 e q_2 - que representam rotação nos eixos x e y respectivamente - são nulos.

Com essas matrizes, é possível representar as acelerações ao longo dos eixos x , y e θ . Levando ainda em consideração os fatores de atrito c_i , o equacionamento será:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{\theta} \end{bmatrix} = Q \left(\begin{bmatrix} c_x \dot{x} \\ c_y \dot{y} \\ c_\theta \dot{\theta} \end{bmatrix} + \bar{B} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \right) \quad (30)$$

Por fim, o espaço de estados da equação (31) é definido com as posições nos eixo x , y e θ e suas respectivas velocidades. Como mostrado em 2.2.2.4, a rotação θ pode ser substituída por dois estados que a representam em quatérnions. Ela ainda pode ser recuperada conforme demonstrado pela equação (32).

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} := \begin{bmatrix} x \\ \dot{x} \\ y \\ \dot{y} \\ \theta \\ \dot{\theta} \\ q_0 \\ q_3 \end{bmatrix} \quad (31)$$

$$\theta = \operatorname{atan} \left(\frac{2q_0q_3}{1 - 2q_3^2} \right) = \operatorname{atan} \left(\frac{2x_6x_7}{1 - 2x_7^2} \right) \quad (32)$$

O conjunto de equações completo que define o modelo em coordenadas globais e a inserção dos parâmetros físicos do sistema pode ser visto na equação (33). Essas equações relacionam as quatro entradas $u_1 \dots u_4$ do sistema com as velocidades e posições referentes ao ambiente em que o veículo se encontra.

$$\begin{aligned}
 \dot{x}_1 &= x_2 \\
 \dot{x}_2 &= k_x c (x_6^2 - x_7^2) (-u_1 + u_2 - u_3 + u_4) - 2k_x s x_6 x_7 (u_1 + u_2 + u_3 + u_4) + c_x x_2 \\
 \dot{x}_3 &= x_4 \\
 \dot{x}_4 &= k_y s (x_6^2 - x_7^2) (u_1 + u_2 + u_3 + u_4) + 2k_y c x_6 x_7 (-u_1 + u_2 - u_3 + u_4) + c_y x_4 \quad (33) \\
 \dot{x}_5 &= k_\theta du_1 - k_\theta du_2 - k_\theta du_3 + k_\theta du_4 \\
 \dot{x}_6 &= -\frac{1}{2} x_5 x_7 \\
 \dot{x}_7 &= \frac{1}{2} x_5 x_6
 \end{aligned}$$

4 Sensoriamento e Fusão de Sensores

Após estabelecer o modelo matemático que rege o veículo, é necessário preparar o ambiente - espaço de trabalho (do inglês, *workspace*) - para adquirir os dados de posição de velocidade dele, a fim de controlar propriamente sua trajetória. Para isso, dois tipos de sensores serão utilizados baseados no comentários das seções 1 e 2.1.4.

O primeiro tipo de sensor será um de posicionamento global. Ele é composto por uma *webcam* localizada acima do veículo, como apresentado em 2.1.5, a fim de captar a imagem de todo seu espaço de trabalho com o veículo incluso.

A servovisão será utilizada para demonstrar a flexibilidade deste tipo de sensor. Podendo captar não somente o objeto alvo, mas também com a possibilidade de estender a quantidade de alvos sem limites impostos pelo sensor e também com a perspectiva de observar obstáculos que interagem com o veículo.

Neste trabalho será demonstrado o caso mais simples, com um veículo e sem a inclusão de obstáculos, isso pode ser feito sem a necessidade de *hardware* adicional, porém agregando custo computacional.

O segundo tipo de sensor utilizado será um sensor local do tipo IMU. Por tratar-se de um sensor capaz de medir apenas acelerações e velocidades angulares, obter informações de posição referentes a um ponto de origem usando somente este sensor é um tema complexo, sendo praticamente inviável seu uso desacompanhado de outros dispositivos para posicionamento absoluto.

A vantagem de usar este sensor é que por ser sensível a acelerações sua precisão é grande, isso faz com que ele seja muito utilizado em conjunto com GPS para obter informações precisas. Uma grande desvantagem do seu uso para navegação é que a IMU tipicamente sofre com a acumulação do seu erro inerente. Dado que ela mede aceleração, e deve somar continuamente essa informação para obter a posição do sensor, a cada parcela que contenha um erro intrínseco, este erro também será somado, desviando a informação da posição verdadeira, esse fenômeno é conhecido como *drifting*.

O sensoriamento proposto estende a idéia da junção de um sensor de posicionamento global com o sensor de medidas inerciais. Contudo, por estar operando em ambiente fechado a câmera será utilizada como sensor de posicionamento global para a extração das informações.

4.1 Servovisão

O sensoriamento de posição do veículo através do sensor visual denomina-se Servovisão. Nesta seção, será discutido o método utilizado para extrair a posição do veículo através da câmera.

Primeiramente, uma configuração inicial da *webcam* com o software Matlab R2011b é necessária, ajustando todos os parâmetros de operação do dispositivo. Igualmente, uma janela de atuação é estabelecida manualmente a fim de obter a melhor imagem para facilitar o cálculo computacional. Após adquirida a imagem pelo software, ela passa por um pré-processamento a fim de diminuir o custo computacional. A função já existente no pacote de processamento de imagens do Matlab *regionprops(.)* é então usada para identificar os três elementos que ditarão a posição do veículo na sua área de trabalho. Assim, a posição em *pixels* do veículo é passada adiante ao programa para a continuação deste.

Essa sequência é demonstrada na Fig. ?? e será abordada em todos seus detalhes a seguir.

4.1.1 Configuração

A configuração inicial da câmera são parâmetros ajustados de acordo com o ambiente em que ela se encontra. Esse passo é necessário ser feito a cada vez que o sistema mudar o ambiente, por isso nessa configuração, ele é adequado a ambientes invariáveis onde o veículo se moverá apenas em um espaço restrito. Para aplicações onde o ambiente não possui fronteiras, o uso do GPS como sensor é mais adequado, ainda existe a opção da configuração *eye-in-hand* para servovisão, como comentado na seção 2.3. Porém, mesmo assim o ambiente ainda necessita possuir elementos reconhecíveis para o sensor de visão.

As configurações para o ambiente onde o protótipo opera estão aqui demonstradas, sendo elas a determinação da velocidade de captura (fps), o sistema de cores utilizado - aqui será utilizado escala de cinza para diminuir o tamanho da imagem e facilitar futuro processamento -, a quantidade de imagens obtidas a cada laço do programa, o modo que esses disparos serão feitos e finalmente, as configurações de imagem, como brilho, contraste e gama. O ajuste da janela de operação é de extrema importância para enquadrar devidamente o espaço de trabalho do robô, visto que caso haja uma movimentação acidental da câmera esse espaço possa ser redimensionado.

```

%% Configuracoes
% Camera
vid = videoinput('winvideo',2,'YUY2_672x480'); % Cria o objeto video
src = getselectedsource(vid);
src.FrameRate = '30.0000'; % Frame Rate
vid.FramesPerTrigger = frame; % Numero de Frames
vid.ReturnedColorspace = 'grayscale'; % Escala de Cinza
vid.TriggerRepeat = Inf; % Numero de Triggers
triggerconfig(vid, 'manual'); % Trigger Manual
src.Brightness = 0; % Brilho
src.Contrast = 255; % Contrase
src.Gamma = 1; % Gamma
%
vid.ROIPosition = [14 16 580 426]; % Ajuste da Janela de Operacao
%
```

4.1.2 Pré-Processamento

Após configurada, a câmera irá captar uma imagem de 672×480 *pixels* de tamanho e essa imagem deve ser pré-processada a fim de aumentar a chance de sucesso no reconhecimento da posição do veículo (ROHS, 2005).

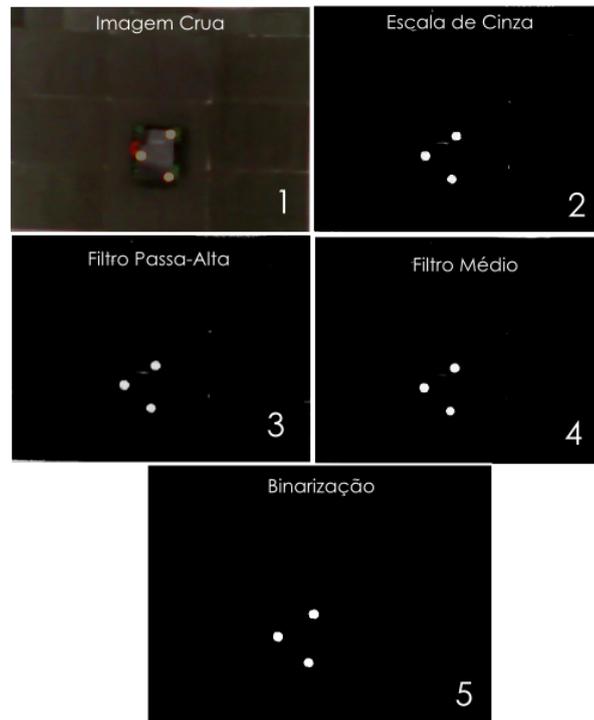
Originalmente em escala de cinza para otimização computacional a informação é capturada com muito ruído, a falta de nitidez contribui negativamente para a identificação do elemento na imagem. Um filtro passa-alta é então desenvolvido para combater as frequências baixas de natureza da imagem. Mesmo que não houvessem essas características, o filtro passa-alta enfatiza a transição do preto para o branco na foto, facilitando assim a tarefa da binarização. O filtro passa-alta que tratará a foto pode ser visto na equação (34).

$$F_{pa} = \begin{bmatrix} 0 & -5 & 0 \\ -5 & 21 & -5 \\ 0 & -5 & 0 \end{bmatrix} \quad (34)$$

Porém, como o filtro passa-alta amplifica demais as altas frequências, um filtro médio também é aplicado à Fig. para suavizar a imagem na tentativa de manter sua fidelidade. A combinação do filtro passa-alta seguido por um filtro médio preserva a qualidade da imagem e facilita a tarefa que vem a seguir. O filtro médio é demonstrado na equação (35).

$$F_m = \begin{bmatrix} \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & \frac{1}{10} & \frac{1}{10} \\ \frac{1}{10} & -\frac{1}{10} & \frac{1}{10} \end{bmatrix} \quad (35)$$

Figura 15 – Etapas de Pré-processamento da Imagem.



Fonte: o Autor.

Para transformar a imagem de escala de cinza para preto e branco, é necessário binarizá-la. A binarização ocorre através de um limiar estabelecido para o valor de cinza em que ocorre a troca do branco para o preto. A imagem então é computada para possuir somente dois níveis, zero e um. A binarização se dá através da equação (36):

$$Pix_{(x,y)} = (Pix_{(x,y)} \geq (th * max(Pix))) \quad (36)$$

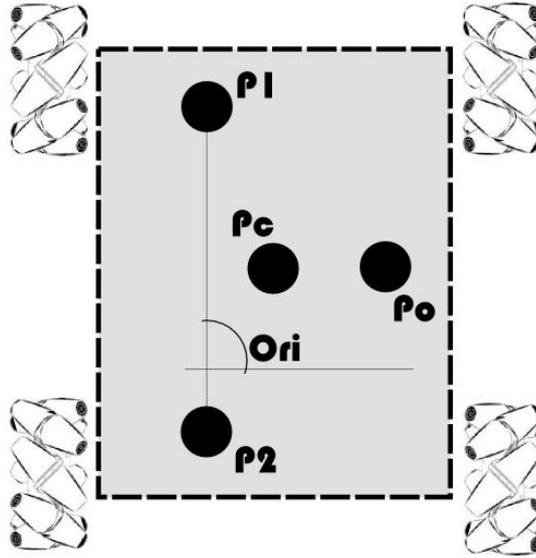
Onde $Pix_{(x,y)}$ é o pixel da coordenada (x, y) que é logicamente comparado ao maior valor da imagem multiplicado pelo limiar entre branco e preto, recebendo valor zero caso seja menor que o limiar ou um quando é maior que o limiar.

Na Fig. 15 são mostradas todas as etapas do pré processamento até o momento em que os três pontos estão prontos para serem reconhecidos pelo computador.

4.1.3 Identificação dos Elementos e Estimação da Posição

Com a imagem tratada, ela pode ser submetida a uma varredura a fim de procurar agrupamentos de *pixels* brancos através da função *regionprops*(·). Essa varredura identificará três grandes grupos que representam a marca de identificação do robô. As coordenadas centrais em *pixels* de cada um desses círculos são chamadas de centróides. Na Fig. 16 é possível ver um exemplo do padrão utilizado.

Figura 16 – Marca a ser captada pelo sensor de visão.



Fonte: o Autor.

Devido ao posicionamento dos pontos, eles são identificados como P_1 , P_2 e P_o conforme a Fig. 16, sendo o ponto P_o o que tem menor distância entre os outros dois pontos.

4.1.3.1 Posição do Veículo

A posição do veículo é facilmente obtida através da média aritmética das coordenadas dos pontos P_1 , P_2 e P_o , conforme a equação 37 demonstra.

Sendo cada ponto $P_i = [x, y]$ com sua posição em *pixels*, a posição do veículo P_c é dada por:

$$P_c = \frac{P_1 + P_2 + P_o}{3} \quad (37)$$

4.1.3.2 Orientação do Veículo

Para calcular a orientação do veículo Ori deve-se encontrar o ângulo formado entre a reta existente entre os pontos P_1 e P_2 e a horizontal. Para isso a função $atan2d(\cdot)$ calcula o arco tangente desses pontos. Porém, existe a possibilidade de o veículo estar exatamente a 180° da posição calculada, para isso, é feito o determinante da matriz mostrada na equação (38) que dependendo do valor, indica o sentido em que o veículo se encontra.

$$D = \begin{bmatrix} x_{P_1} & y_{P_1} & 1 \\ x_{P_2} & y_{P_2} & 1 \\ x_{P_o} & y_{P_o} & 1 \end{bmatrix} \quad (38)$$

Se $\det(D) < 0$, $Ori = Ori + 180^\circ$

4.2 Inertial Measurement Unit

O sensor utilizado para medir as acelerações e velocidades em que o veículo está sujeito foi a ArduIMU v3. Desenvolvido pela DIYDrones, ela utiliza o sensor MPU-6000 que conta com acelerômetro de três eixos e giroscópio de três eixos. Ainda possui um processador ATmega328 para comunicação com os sensores e processamento dos dados.

Este processador será usado para calibração dos sensores, e enviará ao sistema central de processamento do veículo a informação sobre aceleração nos eixos X e Y e a velocidade angular do eixo θ . Esses dados serão enviados diretamente ao computador central, para através do filtro Kalman, realizar a fusão das informações com os dados já obtidos através da servovisão. Isso garante uma informação mais confiável e reduz falhas decorrentes de um sistema com sensoriamento através de um único sensor (WANG, 2006).

4.3 O Filtro de Kalman

O Filtro de Kalman é um algoritmo que usa uma série de medidas que contém algum ruído ou outras imprecisões e produz estimativas de variáveis desconhecidas. Ele funciona com um processo de dois passos: no primeiro passo, a predição, o filtro produz uma estimativa futura do estado atual da variável a ser filtrada levando em consideração suas incertezas. Uma vez estimada, essa variável é novamente medida e também levando em consideração as incertezas do sensor, a estimativa é corrigida com o valor medido.

Este filtro é muito utilizado em navegação integrada, onde o uso de GPS e IMU são combinados (HUTCHINSON, 1984), (QI; MOORE, 2002). Na navegação com sensores globais e locais existe a necessidade de combinar a informação proveniente desses dois para obter dados que combinem as qualidades deles (GELB, 1974). Além do mais, essa redundância permite a detecção de informações errôneas dos sensores, fazendo com que elas sejam rejeitadas. Tal integridade da fusão aumenta a confiabilidade da solução para navegação significativamente.

Resumidamente, o filtro é encarregado de fazer a fusão de um modelo (*a priori*) com os sensores (*a posteriori*) cujos estados deseja-se estimar. Para fazer a predição das variáveis, o filtro se baseia num modelo do sistema em que opera. A equação (39) exemplifica a operação do Filtro de Kalman.

$$\begin{cases} x_k = A_k x_{k-1} + B_k u_k + \epsilon_k \\ z_k = C_k x_k + \delta_k \end{cases} \quad (39)$$

Onde x_k são os estados do modelo e u_k é a entrada do sistema. A variável z_k representa as medidas dos estados do modelo e as matrizes A_k , B_x e C_k são as matrizes do modelo, podendo ser variantes no tempo. O termo δ_k é a incerteza proveniente dos sensores enquanto ϵ_k é a incerteza inerente ao modelo.

O algoritmo de implementação do filtro é visto a seguir:

```
function [x,P] = Fkalmán(x,P,A,B,C,T,R,Q,z,w,ok)

    x = A*x+ T*B*w;
    P = A*P*A'+R;

    if ok

        e = z-C*x;
        K = P*C'*inv(C*P*C'+Q);

        x = x + K*e;
        P = (eye(4)-K*C)*P;
    end

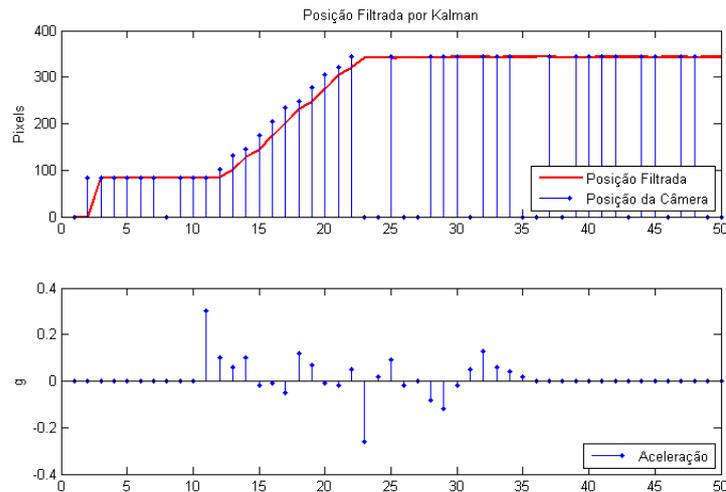
end
```

Onde as matrizes P e Q representam as incertezas do modelo e do sensor respectivamente. Nota-se ainda que o segundo passo do filtro, o passo de medição dos sensores pode ser evitado caso tenha acontecido algum erro de medida, no código representado pelo ok , fazendo com que a estimativa do filtro caso haja erro na medição do sensor ($ok = 0$) seja somente dada pelo modelo. Na Fig. 17 pode ser vista a filtragem para um dos graus de liberdade do veículo. Neste caso, erros propositalmente foram induzidos no sensor de visão para testar a estimativa do filtro, que com sucesso, conseguiu responder.

4.4 O Filtro de Kalman Extendido

Conforme demonstrado na equação (39), o Filtro de Kalman baseia-se num modelo linear para fazer sua predição, contendo as matrizes A e B do sistema. No entanto, transições de estado e de medidas raramente são lineares na prática. Tomando como exemplo um robô que se mova com velocidades de translação e rotação constantes, ele estará se movendo em uma trajetória circular que não pode ser descrita com transições de estados lineares. Essa observação torna o Filtro de Kalman discutido até agora inaplicável para grande parte dos casos, a não ser os casos mais simplistas, como por exemplo movimentação em apenas um eixo.

Figura 17 – Posição no eixo Y da câmera e filtrada.



Fonte: o Autor.

O Filtro de Kalman Extendido (do inglês, *Extended Kalman Filter* - EKF), permite relaxar a suposição de que o sistema necessita ser linear para seu correto funcionamento. Nele, as equações de predição e correção dos estados são governadas por funções não lineares, sendo elas g e h , respectivamente. O EKF pode ser visto então na equação (40), mantendo o mesmo formato do Filtro de Kalman, com a primeira equação sendo a estimativa dos estados através do modelo, e a segunda a correção dos estados através dos sensores.

$$\begin{cases} x_k = g(u_k, x_{k-1}) + \epsilon_k \\ z_k = h(x_k) + \delta_k \end{cases} \quad (40)$$

A ideia por trás do EKF é a linearização via Expansão de Taylor, o modelo não linear é linearizado e utilizado iterativamente, para realizar as estimativas dos estados através da função g . O mesmo conceito pode ser aplicado para as medições dos sensores, caso elas também contenham não linearidades intrínsecas ao seu processo, originando a função h .

O algoritmo do Filtro de Kalman Extendido pode ser visto a seguir, onde dx é o modelo do sistema, a matriz G é a aproximação via Expansão de Taylor do mesmo, e z são as medições de posição do sistema.

```
function [x, P] = EKF(x, P, R, Q, C, z, u_x, u_y, u_t, a, d, kI, cI)
    dx = model(x, u_x, u_y, u_t, a, d, kI, cI);
    G = createG(dx);
```

```

P = G*P*G'+R;

e = z-C*dx;
K = P*C'*1/(C*P*C'+Q);

x = dx + K*e;
P = (eye(7)-K*C)*P;

```

end

O Filtro de Kalman foi implementado e testado para funcionamento em um eixo do sistema, que pode ser tratado como um integrador duplo. Porém para a implementação completa, o modelo não linear deve ser considerado, surgindo aí a necessidade de utilização do Filtro de Kalman Extendido no lugar.

A implementação da etapa de predição do EFK consiste na estimativa dos estados futuros baseado no modelo não linear com a medida das entradas aplicadas ao sistema, para realizar a etapa de correção, a função g deve ser linearizada, e essa linearização é representada na matriz G . Dado que a aquisição dos sensores se dá de maneira linear, a função h não necessita ser implementada, bastando utilizar a matriz C como saídas do sistema.

Dado o modelo visto na equação (33), a matriz G baseada nele pode ser vista na equação (41).

$$G = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_x & 0 & 0 & 0 & G_{26} & G_{27} \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & c_y & 0 & 0 & 0 & G_{46} & G_{47} \\ 0 & 0 & 0 & 0 & c_\theta & 0 & 0 \\ 0 & 0 & 0 & 0 & -\frac{1}{2}dx_7 & 0 & -\frac{1}{2}dx_5 \\ 0 & 0 & 0 & 0 & \frac{1}{2}dx_6 & \frac{1}{2}dx_5 & 0 \end{bmatrix} \quad (41)$$

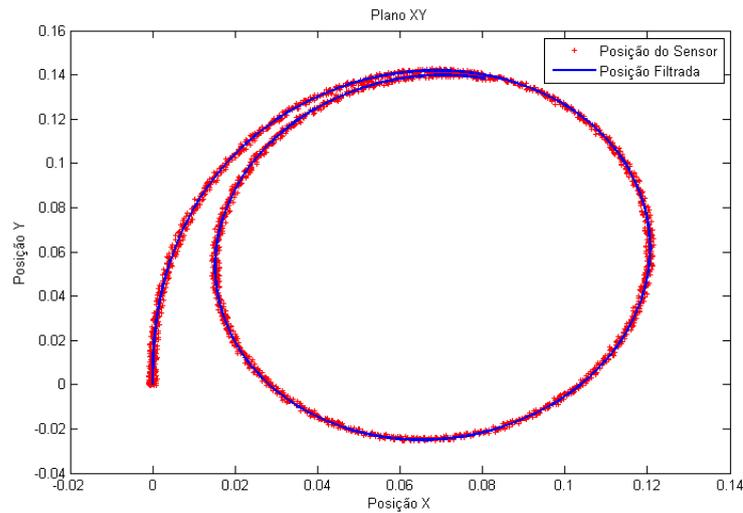
Sendo,

$$\begin{aligned}
G_{26} &= 2k_x c dx_6 (-u_1 + u_2 - u_3 + u_4) - 2k_x s x_7 (u_1 + u_2 + u_3 + u_4) \\
G_{27} &= -2k_x c dx_7 (-u_1 + u_2 - u_3 + u_4) - 2k_x s x_6 (u_1 + u_2 + u_3 + u_4) \\
G_{46} &= 2k_y s x_6 (u_1 + u_2 + u_3 + u_4) + 2k_y c x_7 (-u_1 + u_2 - u_3 + u_4) \\
G_{47} &= -2k_y s x_7 * (u_1 + u_2 + u_3 + u_4) + 2k_y c x_6 (-u_1 + u_2 - u_3 + u_4)
\end{aligned}$$

Na figura 18 é simulado o exemplo do veículo andando em uma trajetória circular, em malha aberta, com o Filtro de Kalman Extendido aplicado para estimar sua trajetória.

O EKF em várias maneiras é semelhante ao Filtro de Kalman, suas diferenças estão representadas na tabela 2.

Figura 18 – Filtro de Kalman Extendido aplicado ao modelo.



Fonte: o Autor.

Tabela 1 – Diferenças entre o Filtro de Kalman e o EKF

	Filtro de Kalman	EKF
predição	$A_k x_{k-1} + B_k u_k$	$g(u_k, x_{k-1})$
correção	$C_k x_k$	$h(x_k)$

Fonte: o Autor

Isto é, as aproximações lineares feitas pelo Filtro de Kalman são generalizadas no EKF, que utiliza as próprias equações não lineares do sistema. A correção, do mesmo modo, também pode considerar medidas não lineares dos sensores, elas serão expressas na função h .

5 Implementação e Controle do Sistema

Neste capítulo serão vistas as aplicações práticas do modelo e sensoriamento. Na primeira parte, o modelo desenvolvido em no capítulo 3 será identificado, para obter os parâmetros reais do protótipo utilizado. Após, para esse sistema serão implementados diferentes controladores a fim de comparar sua performance e implementá-los no protótipo.

5.1 Identificação do Sistema

O modelo completo do sistema Omni-Direcional é complexo, contendo quatro entradas - uma para cada roda - e uma relação não linear entre elas e as saídas. Porém, uma outra abordagem ao problema (SA; CORKE, 2012) permite tratar cada um dos graus de liberdade do veículo como um sistema isolado do tipo integrador duplo contendo ainda um ganho k_i proporcional e um pólo relacionado ao atrito do sistema.

Ao abordar o problema desta forma, existirão três sistemas isolados e menos complexos, possibilitando uma identificação dos parâmetros de cada um desses sistemas para posteriormente incluir os parâmetros no modelo original.

5.1.1 Discretização dos Sistemas

Existirão então três sistemas – um para cada grau de liberdade – como o da equação (42) sendo $i = x, y$ e θ . Onde k_i é o ganho de cada grau e c_i é o atrito relacionado a cada um dos graus.

$$\frac{Y_i(s)}{U_i(s)} = \frac{k_i}{s(s + c_i)} \quad (42)$$

Por se tratar de um sistema cuja atuação, sensoriamento e controle são feitos digitalmente, é necessário discretizar o sistema da equação (42), já que este representa um sistema contínuo no domínio do tempo. Para isso, o método de discretização por Euler Forward é utilizado, onde $s \approx \frac{z-1}{T}$, gerando um sistema da forma vista na equação (43). Sendo T o tempo de amostragem do sistema.

$$\frac{Y_i(z)}{U_i(z)} = \frac{k_i T^2}{z^2 + z(c_i T - 1) - c_i T} \quad (43)$$

5.1.2 Determinação do Tempo de Amostragem

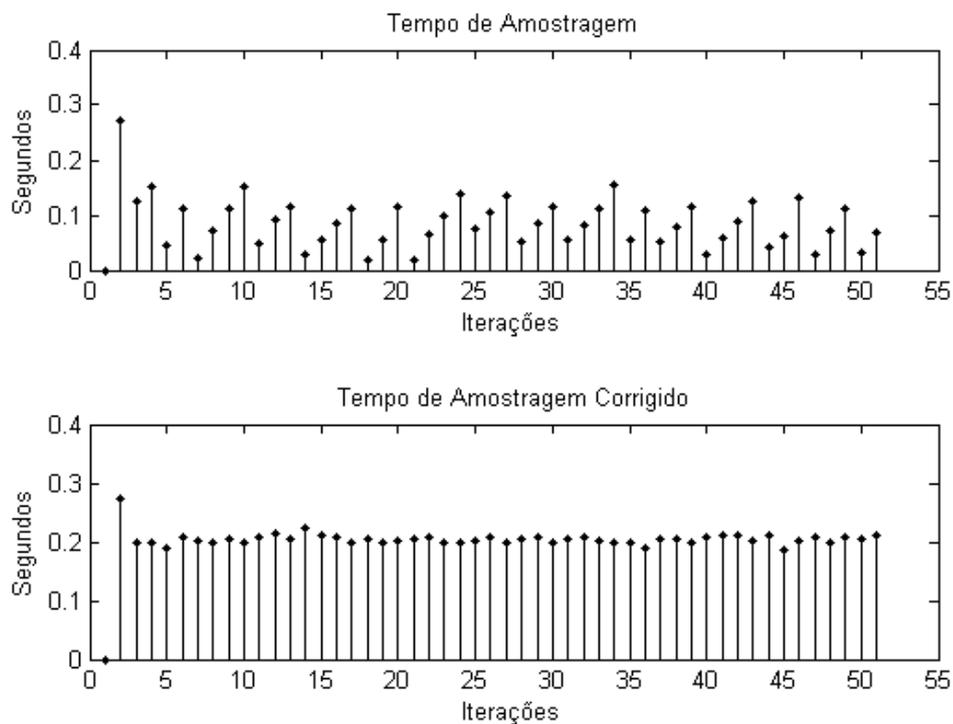
O sistema como um todo trabalha em volta da comunicação do veículo com o computador central. Levando em consideração que o software que controla o veículo somente

completa um laço após comunicar-se com o computador, e este depende da velocidade do sensor visual, que trabalha a 30 fps (do inglês, *frames per second*) e ainda dos cálculos de sensoriamento e controle, foram feitas simulações para estimar o tempo de amostragem do sistema como um todo.

A Fig. 19 mostra o resultado dessa simulação, dela é possível notar que o sistema varia no seu tempo de execução. Isso é devido ao fato de que o sensor de visão nem sempre trabalha na sua velocidade máxima - 30 fps - e que cada imagem captada por ele exige um tempo distinto para ser processada. Ainda é possível notar um pico inicial devido a calibração da imagem nesta simulação.

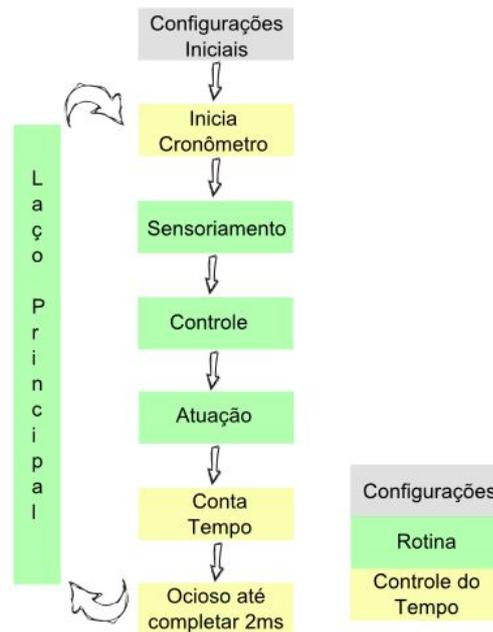
Para garantir um tempo de amostragem constante, uma função é criada de maneira que um tempo fixo T seja imposto. Caso o programa se adiante a este tempo, ele ficará ocioso até atingir o tempo limite. Isto é exemplificado no algoritmo da Fig. 20, que demonstra este processo. Com isso, o tempo de amostragem de segurança será $T = 20ms$, como pode ser visto na Fig. 19. Devido ao pico presente no começo da simulação, a movimentação e controle do veículo acontecem somente após o tempo de execução se estabilizar.

Figura 19 – Tempo de execução do sistema antes e após ser corrigido.



Fonte: o Autor.

Figura 20 – Algoritmo de ajuste do tempo de amostragem.



Fonte: o Autor.

5.1.3 Aquisição de Dados para Identificação

Tendo cada um dos modelos discretos feitos, o passo seguinte é excitar o veículo real e captar seus dados de posição. Já que a identificação será feita separadamente em cada um dos eixos, o veículo é movimentado isoladamente em cada um dos lados do eixo X , o mesmo será feito para o eixo Y e finalmente ele será rotacionado tanto no sentido horário como anti-horário para obter os dados do grau de rotação.

Uma entrada do tipo degrau é aplicada a cada um dos eixos e sua posição - em pixels - é medida para captar os dados para identificação. Visto que a entrada do sistema é separada em cada uma das rodas, a equação (1) é utilizada para calcular o sinal de cada roda conforme a tabela 2.

Tabela 2 – Tabela de Relação dos sinais de cada roda.

Rodas	Eixo X	Eixo Y	Eixo θ
w_1	-u	u	u
w_2	u	u	-u
w_3	-u	u	-u
w_4	u	u	u

Fonte: o Autor.

Esses dados foram captados para, através do método dos Mínimos Quadrados,

para realizar a identificação das funções integradores duplos, vistas na equação (42) de cada um dos eixos. Assim, será possível conhecer os parâmetros k_i e c_i de cada um dos eixos.

5.1.4 Método dos Mínimos Quadrados

A partir da equação (43) é montado o modelo para estimar os parâmetros k_i e c_i do sistema. Fazendo

$$Y_i(z)(z^2 + z(c_i T - 1) - c_i T) = U_i(z)(k_i T^2) \quad (44)$$

e aplicando a transformada \mathcal{Z} na equação (44) se obtém:

$$y(k+2) + c_i T(y(k+1) - y(k)) - y(k+1) = u(k)T^2 \quad (45)$$

Essa equação ainda pode ser levada à forma matricial $y_1(k) = \Psi_1 \Theta$, sendo que no vetor coluna Θ localizam-se os parâmetros a serem identificados. O resultado será:

$$\begin{bmatrix} y(k+2) - y(k-1) \end{bmatrix} = \begin{bmatrix} -T(y(k+1) - y(k)) & T^2 u(k) \end{bmatrix} \begin{bmatrix} c_i \\ k_i \end{bmatrix} \quad (46)$$

Assim, ao realizar n amostras do modelo, é possível criar um sistema $y_n(k) = \Psi_n \Theta$ sendo n o número de linhas das matrizes $y_n(k)$ e Ψ_n . Para conhecer o valor dos parâmetros k_i e c_i basta resolver a equação (47), visto que ela pode ser resolvida para qualquer n , notando que o vetor Θ não altera seu tamanho.

$$\Theta = (\Psi^t \Psi)^{-1} \Psi^t y \quad (47)$$

Ao realizar esse procedimento para cada um dos seis ensaios realizados e transformando de volta os sistemas discretos em sistemas contínuos, serão obtidos os parâmetros k_i^+ , k_i^- , c_i^+ e c_i^- sendo os índices $+$ e $-$ as representações de cada um dos sentidos dos ensaios. A média de cada um destes servirá como base para o modelo simulado, a fim de controlá-lo e em seguida, aplicar o controle no sistema real.

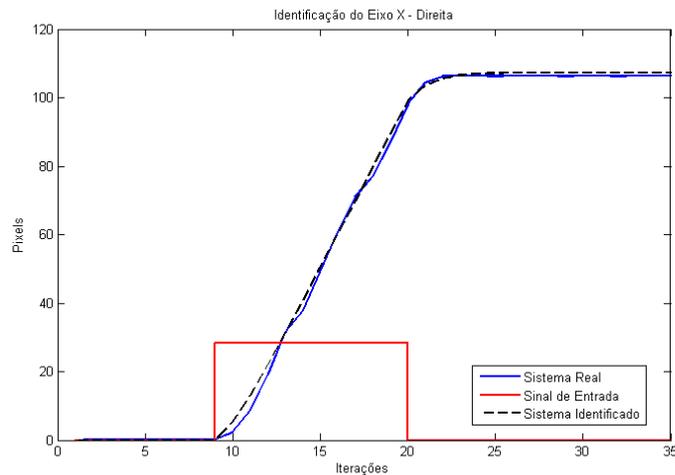
5.2 Resultados da Identificação

A identificação do sistema no eixo X , ou seja, na posição horizontal em relação a imagem obtida pela câmera foi feita em ambos sentidos, uma vez levando o veículo da esquerda para a direita, e em seguida no sentido oposto. Seus sinais de entrada e saída foram captados e eles podem ser vistos juntamente com a função identificada para o eixo X em cada um dos sentidos ensaiados nas Figs 21a e 21b. A partir dos dois sentidos, duas

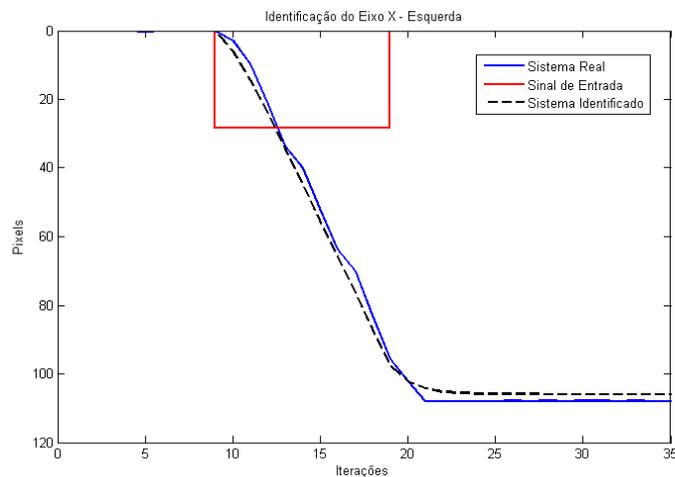
funções do tipo visto na equação 42 foram identificadas e aproximadas, dando origem aos parâmetros k_x e c_x .

Figura 21 – Identificação do Eixo X

(a) Movimento para a Direita



(b) Movimento para a Esquerda



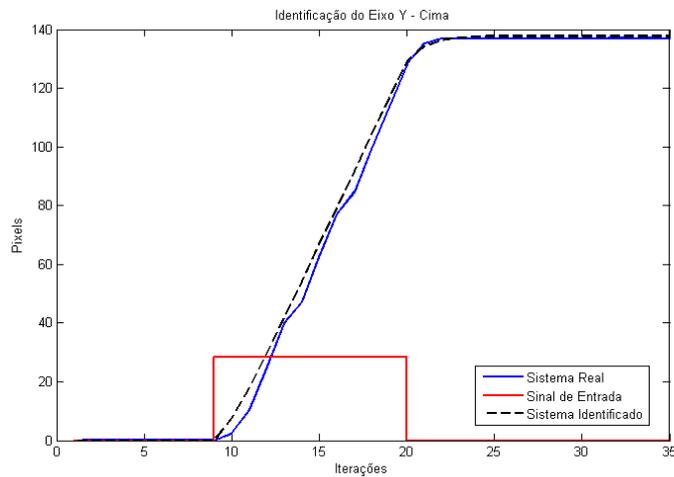
Fonte: o Autor.

Os sistemas identificados para o eixo Y , sentido vertical na imagem, foram feitos de maneira análoga ao eixo anterior. Aqui, o veículo foi movimentado para frente e para trás, a fim de captar seus dados de entrada de controle e posição para conhecer os parâmetros k_y e c_y . Ambos ensaios podem ser vistos na Fig. 22, onde a Fig. 22a representa o ensaio com o veículo andando para frente (subindo em relação a imagem da câmera) e a Fig. 22b representa o veículo no sentido oposto. O sistema identificado também pode ser visto para comparação com os dados captados.

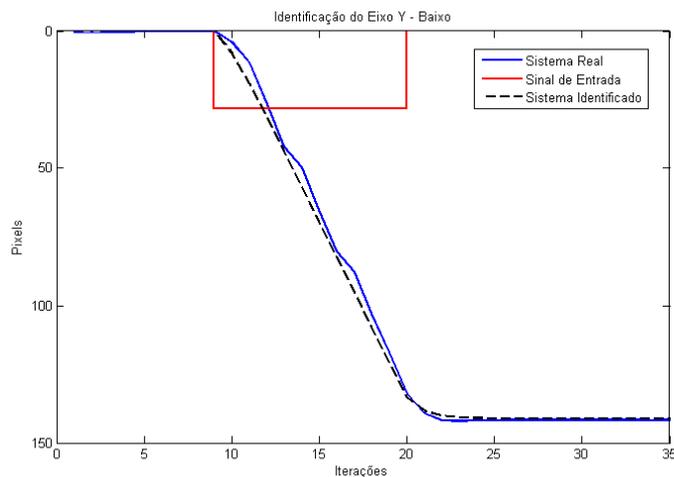
O eixo de rotação foi identificado do mesmo modo que os eixos de translação, agora com sua posição medida em graus. Ele foi primeiramente rotacionado no sentido horário

Figura 22 – Identificação do Eixo Y

(a) Movimento para Cima



(b) Movimento para Baixo



Fonte: o Autor.

e em seguida no sentido anti-horário. Seus dados de entrada e saída podem ser vistos nas Figs. 23a e 23b. Ainda pode ser visto a simulação do sistema identificado juntamente na imagem, a fim de compará-lo com o sistema real. Os parâmetros k_θ e c_θ foram finalmente identificados e juntamente com os outros quatro, serviram de base para montar o modelo a ser controlado a fim de desenvolver o sistema de controle para o protótipo.

Tabela 3 – Tabela dos Parâmetros Físicos do Sistema.

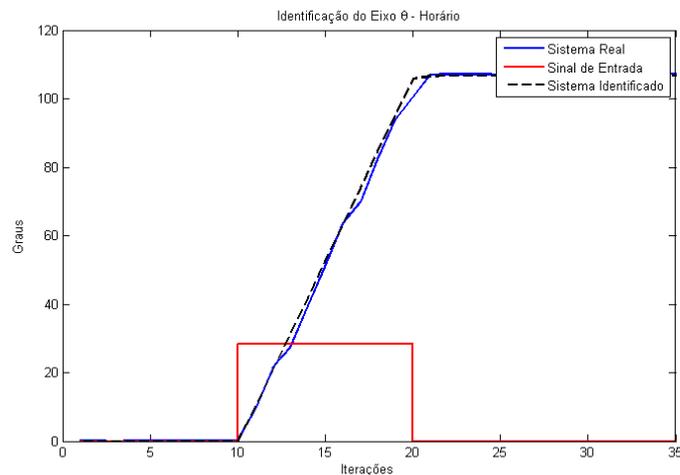
Parâmetros	Eixo X	Eixo Y	Eixo θ
k_i	4.97	6.86	6.50
c_i	-2.23	-1.93	-1.09

Fonte: o Autor

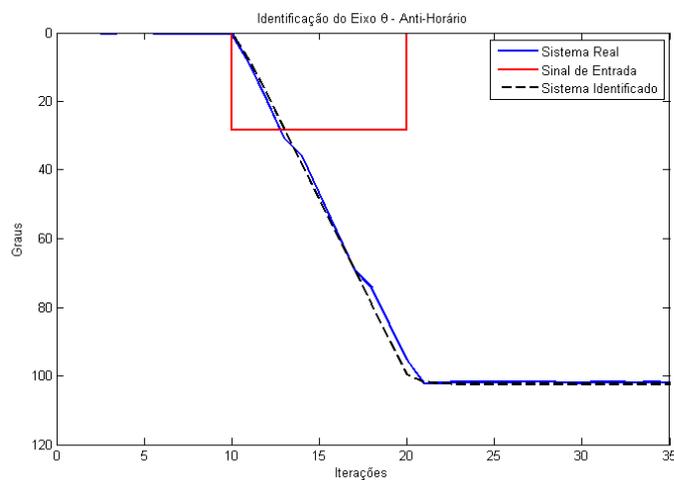
A tabela 3 mostra o resultado com todos os valores usados como referência no modelo desenvolvido.

Figura 23 – Identificação do Eixo θ

(a) Sentido Horário



(b) Sentido Horário



Fonte: o Autor.

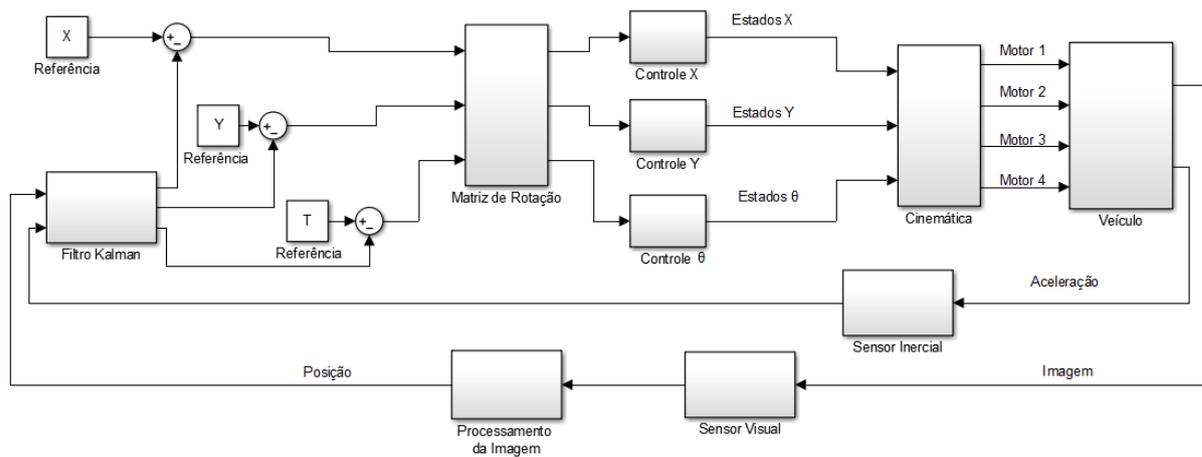
5.3 Controle

O sistema de controle é um componente essencial para o projeto atingir seu objetivo de movimentar-se sem a ajuda externa, ou seja, de forma autônoma. Ele é o responsável por tomar a referência - ponto que o veículo deve atingir - e gerar um sinal para cada roda de maneira que o veículo, não importando onde esteja, alcance-a.

Para isso, o sistema de controle será dividido em três, um para cada grau de liberdade. A referência toma a forma de $r = [x, y, \theta]$, sendo a primeira posição do vetor o valor em *pixels* no eixo X , a segunda o valor em *pixels* no eixo Y e a terceira será o

valor da orientação do veículo em graus. Assim, será calculado um erro referente a cada um dos três graus de liberdade que devem ser rotacionados para manter o sistema em suas coordenadas globais. Esse erro rotacionado servirá para os controladores calcularem o sinal que após passar pela cinemática que transforma os sinais de controle em cada eixo para o sinal de cada roda, será enviado ao veículo. O esquemático explicando esse sistema pode ser visto na Fig. 24.

Figura 24 – Esquemático do sistema completo.



Fonte: o Autor.

5.3.1 Tipos de Controladores

Para validar o modelo criado no capítulo 3 e implementar o sistema de controle, foi primeiramente criado um ambiente virtual de simulação no software Matlab R2011b. O modelo com seus sete estados foi transcrito no ambiente e um controlador do tipo Proporcional foi inicialmente elaborado.

Devido a característica do sistema de não possuir acoplamento entre os eixos, ou seja, cada grau de liberdade pode variar independentemente dos outros dois, um controlador para cada eixo foi implementado separadamente. A equação (48) é a implementação do controlador para os eixos $i = x, y, \theta$, sendo e_i o erro entre a posição atual no eixo e a referência do sistema.

$$u_i = e_i P_i \quad (48)$$

O ganho do controlador foi manualmente ajustado para que ele atingisse uma performance satisfatória. Após a verificação do modelo em malha fechada, outros dois controladores P foram implementados para completar a movimentação do veículo no plano XY

Para melhorar a performance, um controlador do tipo PD foi posteriormente utilizado, sua implementação ocorre de acordo com a equação (49), sendo v_i a velocidade do sistema. Novamente ele é iniciado somente em um grau de liberdade e em seguida estendendo-se para dois graus.

$$u_i = e_i P_i - v_i D_i \quad (49)$$

Por fim, o controlador PTOS, visto na seção 2.4.3 foi implementado, primeiramente simulado somente no eixo X . Após sua verificação a simulação foi estendida para o plano XY e em seguida para uma manobra realizada nos três graus de liberdade simultaneamente.

5.3.2 Implementação dos Controladores

Por fim, cada sistema de controle implementado foi simulado e testado no protótipo. Como o controle é aplicado sobre as posições do veículo, ele deve manter as referências de um sistema global para não perder informações. Isso é alcançado através do uso da matriz de rotação Q (equação (3.2)) para rotacionar os erros de referência e'_x e e'_y dos eixos X e Y em relação a orientação atual do veículo, evidenciado na equação (50). Os sistemas de controle são então aplicados levando em consideração o erro em cada grau de liberdade e o sinal de controle para cada grau é calculado. O sinal de cada eixo é combinado de acordo com a equação (1) da cinemática do sistema para obter o sinal de cada uma das rodas do veículo.

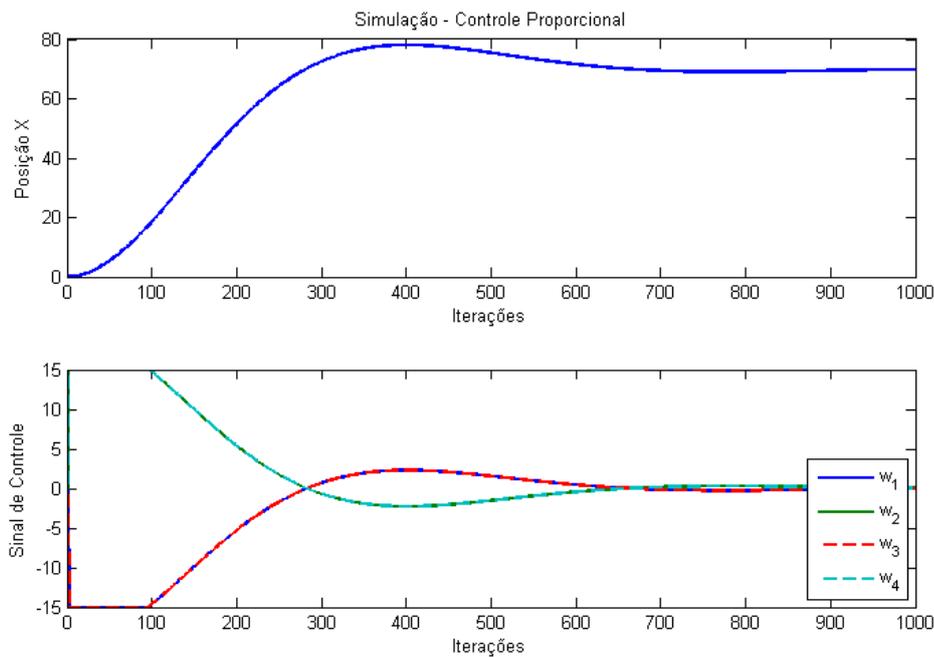
$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} q_0^2 - q_3^2 & -2q_0q_3 & 0 \\ 2q_0q_3 & q_0^2 - q_3^2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} e'_x \\ e'_y \\ e'_\theta \end{bmatrix} \quad (50)$$

6 Resultados e Discussões

Aqui serão apresentadas as simulações feitas no ambiente virtual com os controladores do tipo P, PD e PTOS e simulações feitas tanto para o eixo X isoladamente, como para os eixos XY em conjunto.

A Fig. 25 demonstra o sistema simulado somente no eixo X e controlado pelo controlador mais simples, o Proporcional. Nota-se um pequeno sobressinal no sistema na tentativa de alcançar a referência $r = 70$ em um tempo semelhante aos outros controladores. Porém isso demonstra que o veículo pode atingir a referência desejada em um tempo melhor, sem a necessidade de passar pela referência e ter que voltar.

Figura 25 – Simulação do modelo no eixo X e com controle Proporcional.

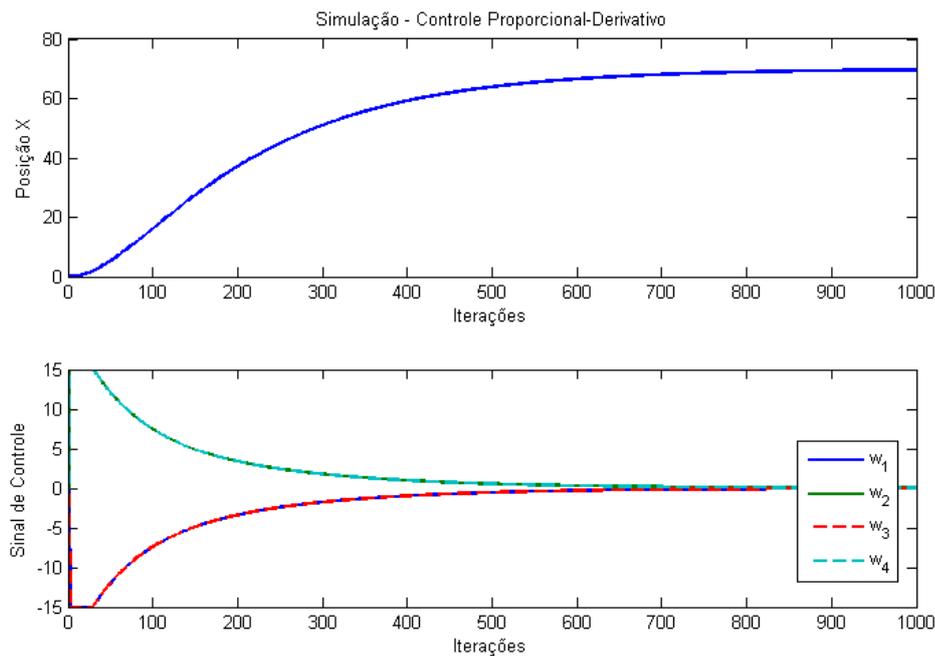


Fonte: o Autor.

O controlador Proporcional não obtém uma trajetória de tempo ótimo e devido a saturação no sinal de controle não é possível obter uma resposta mais agressiva do mesmo sem prejudicar o sistema. Para isso, é necessário o uso de um sistema de controle mais complexo. A inclusão de um sinal referente a derivada do erro de posição faz com que o controle obtenha uma resposta melhor, já que leva em consideração a velocidade do veículo para desacelerá-lo com mais eficiência. O controlador Proporcional-Derivativo é então implementado visando obter um sinal de controle mais agressivo, eliminando o sobressinal apresentado.

O resultado para a simulação do modelo controlado pelo PD e com mesma referência pode ser visto na Fig. 26, nota-se uma melhora na performance tanto em tempo de estabilização, eliminando também o sobressinal e diminuindo a energia gasta pelo atuador. Este controlador apresenta um desempenho mais próximo ao PTOS, porém com a mudança da referência ele não mantém a performance, enquanto o PTOS mantém o mesmo sinal de controle para qualquer referência utilizada.

Figura 26 – Simulação do modelo no eixo X e com controle Proporcional-Derivativo.



Fonte: o Autor.

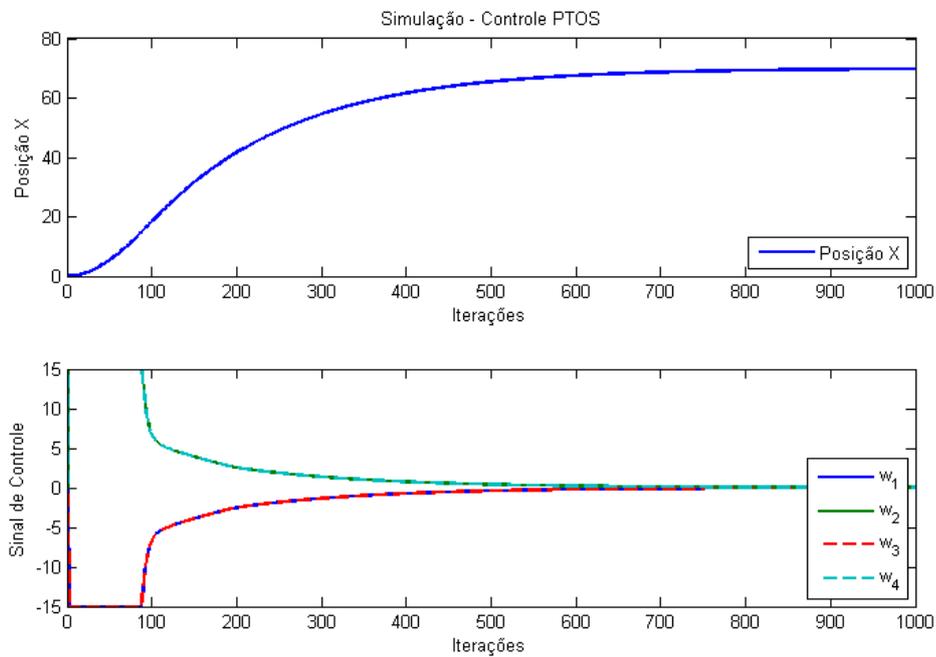
O melhor desempenho é o apresentado pelo controlador PTOS, na Fig. 27, visto que este é um controlador específico. Por levar a saturação em consideração na sua construção, ele aproveita melhor este fato para gerar um sinal de controle mais agressivo no primeiro momento, fazendo com que o sistema estabilize em um tempo menor que os controladores vistos anteriormente.

A simulação em apenas um eixo é feita para ajustar os parâmetros dos controladores da melhor maneira possível e verificar sua performance atuando em mais de um eixo ao mesmo tempo. A seguir, as simulações foram estendidas para uma movimentação do veículo no plano XY , fazendo um deslocamento diagonal.

Na Fig. 28 pode-se notar a interação entre os dois graus de liberdade, porém com o controlador P, o desempenho é considerado baixo. A diferença dos parâmetros para os graus de liberdade fazem com que os controles não consigam atingir a trajetória ótima para a referência. Visto que por mais que cada um esteja parametrizado de acordo com seu grau de liberdade, a combinação dos sinais de controle nas quatro rodas cria um desvio

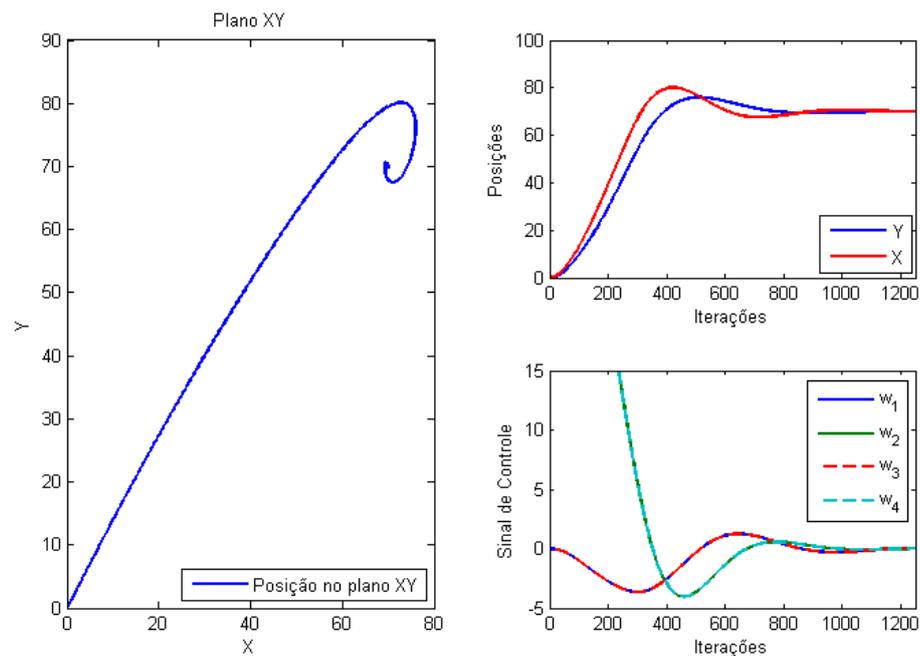
indesejado, fazendo com que o veículo não atinja uma trajetória ideal para o movimento diagonal.

Figura 27 – Simulação do modelo no eixo X e com controle PTOS.



Fonte: o Autor.

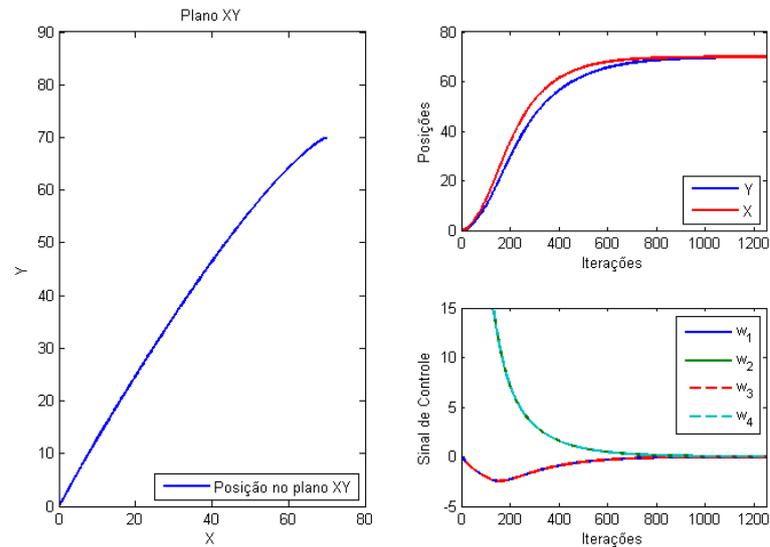
Figura 28 – Simulação do modelo no plano XY e com controle Proporcional.



Fonte: o Autor.

O controlador PD também foi simulado operando simultaneamente em duas dimensões e apresentou resultados satisfatórios, demonstrados na Fig. 29, porém novamente a trajetória não é ótima e o tempo de estabilização ainda pode ser melhorado.

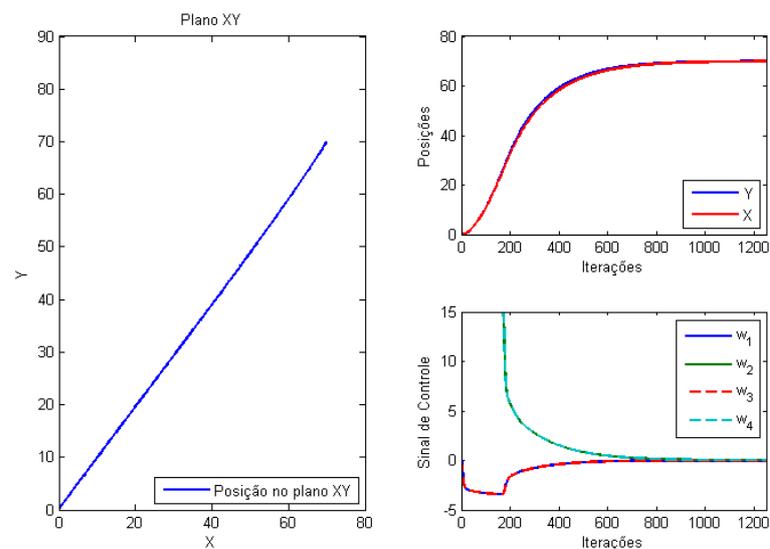
Figura 29 – Simulação do modelo no plano XY e com controle Proporcional-Derivativo.



Fonte: o Autor.

O controlador PTOS atinge uma trajetória ótima, visto na Fig. 30, para o sistema se movendo em duas dimensões simultaneamente, seu sinal de controle consegue equilibrar o movimento entre os eixos gerando um deslocamento coordenado no sistema.

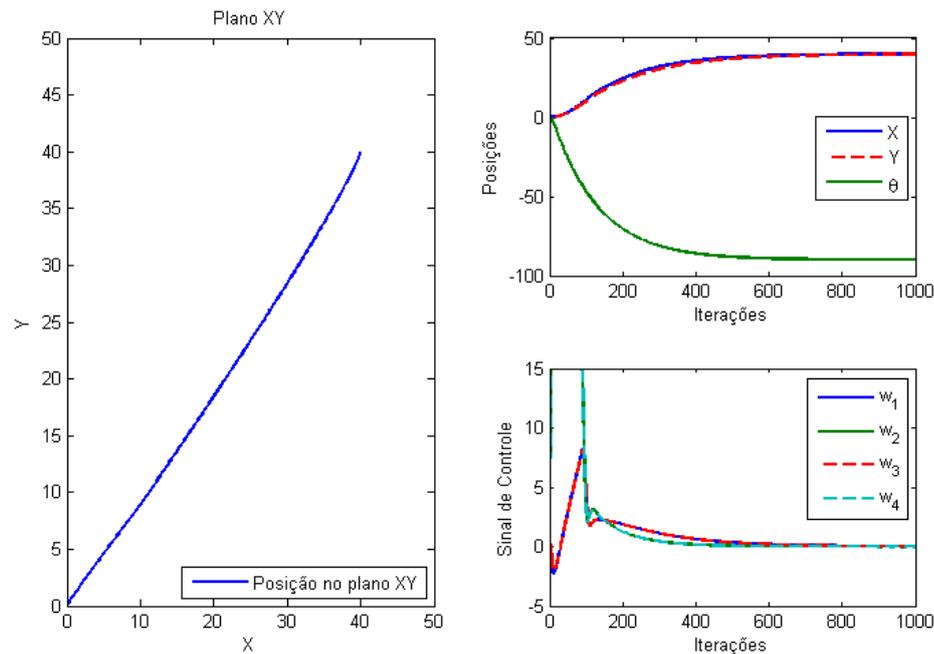
Figura 30 – Simulação do modelo no plano XY e com controle PTOS.



Fonte: o Autor.

O movimento simultâneo entre os três eixos - fazendo um deslocamento diagonal enquanto gira - é um movimento complicado fazendo com que os controladores P e PD não foram capazes de estabilizar o sistema em tempo aceitável, porém novamente o controlador PTOS foi capaz de coordenar os três graus de liberdade ao mesmo tempo, fazendo com que o sistema se deslocasse de maneira ótima mesmo realizando uma manobra complexa. Este é demonstrado na Fig. 31.

Figura 31 – Simulação do modelo no plano XY, incluindo giro e com controle PTOS.

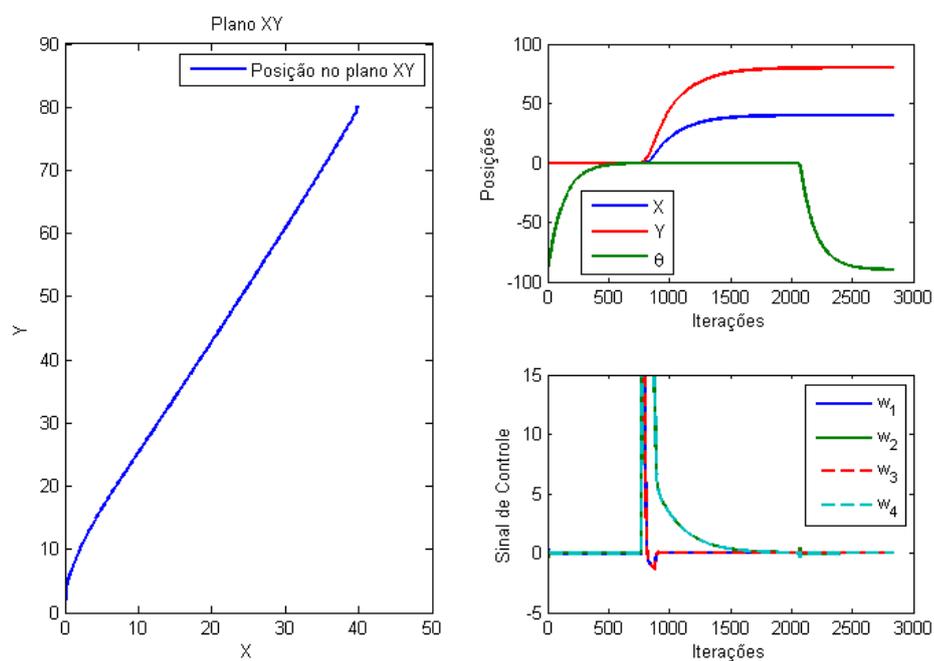


Fonte: o Autor.

Outro ensaio feito com o controlador PTOS foi a movimentação sucessiva do giro com uma referência diferente para X e Y . Na Fig. 32, vemos as referências $r_x = 40$ e $r_y = 80$ seguidos de um giro de 90° do veículo.

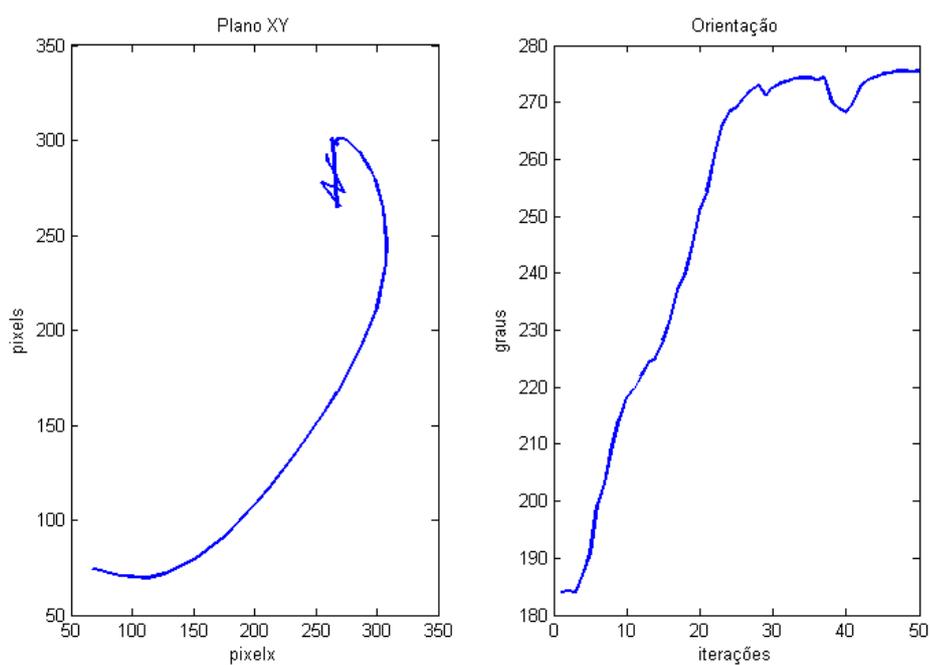
Todas as simulações foram repetidas com o sistema real, que apresentou resultados semelhantes porém com maior ruído devido a imperfeição dos sensores. Na Fig. 33 pode ser vista a manobra mais complexa, que consiste no deslocamento diagonal enquanto o veículo faz um giro. Essa manobra é trivial em veículos com eixo frontal que permite a mudança de direção, como um carro comum, porém para um veículo que não possui essa habilidade é necessária uma combinação complexa de velocidades nas rodas para atingir o mesmo movimento. Esse movimento é análogo ao de um carro estacionando - deslocando-se em um L - e a imagem quadro a quadro desse movimento pode ser vista na Fig. 34.

Figura 32 – Movimentos nos três eixos com referências distintas.



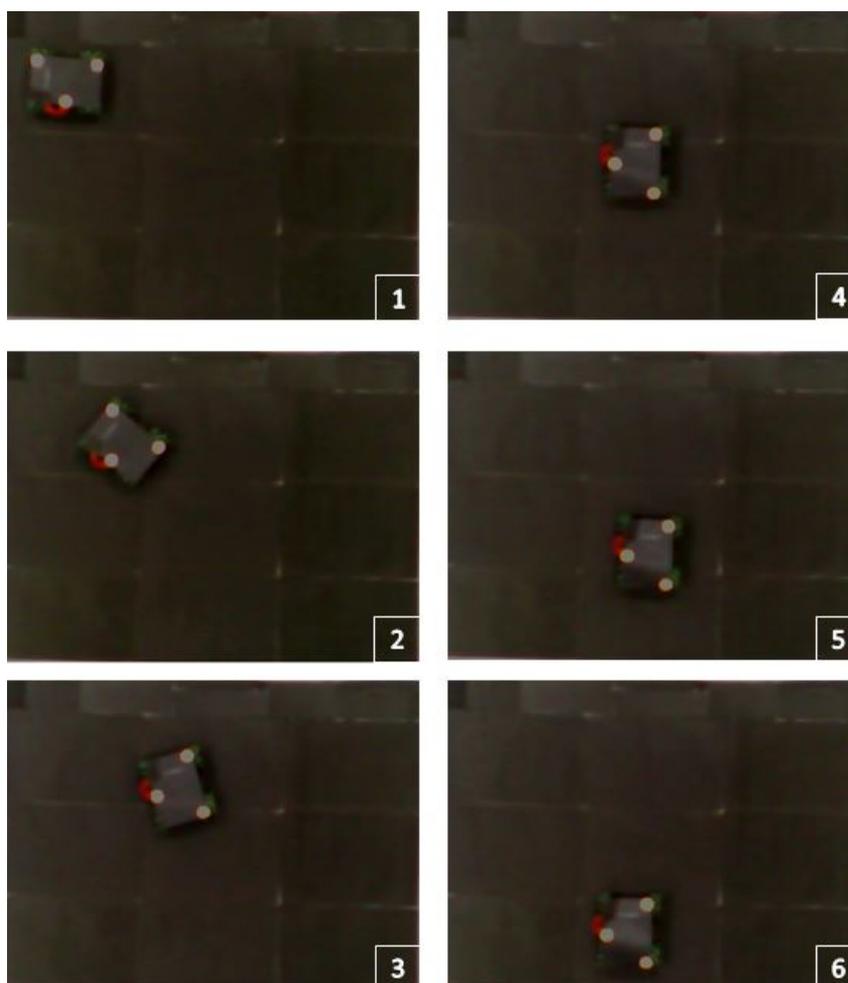
Fonte: o Autor.

Figura 33 – Manobra no plano XY com giro com controlador PTOS.



Fonte: o Autor.

Figura 34 – Quadro a quadro da manobra em "L".



Fonte: o Autor.

7 Conclusão

Com a grande evolução da tecnologia e a crescente exposição de novos estudos o desenvolvimento se expande em todas as áreas. Robôs dos tipos mais diversos surgiram para executar as mais variadas tarefas. Dentro deste campo, os robôs autônomos são aqueles que executam suas funções sem o auxílio de um humano para interferir na sua tomada de decisão. Eles devem ser capazes de ler e interpretar o ambiente a sua volta e decidir pelo melhor caminho, ou pela melhor maneira de cumprir sua missão.

Este trabalho propôs o desenvolvimento de um veículo autônomo capaz de manobrar e se movimentar dentro de um espaço limitado sem interferência externa. O objetivo foi cumprido dividindo a proposta em tarefas menores e mais específicas para serem solucionadas uma a uma, porém sempre pensando na integração do sistema como um todo.

A primeira problemática foi definir o tipo de veículo a ser utilizado. O veículo Omni-Direcional foi escolhido devido a sua facilidade de executar manobras e devido ao seu desacoplamento entre seus graus de liberdade, fazendo com que a tarefa de controle fosse facilitada. O protótipo utilizado esteve apto a demonstrar as vantagens deste tipo de veículo, validando igualmente a aplicação do sistema de controle no seu modelo matemático.

O modelo desenvolvido se mostrou fiel ao comportamento do sistema, além disso foi criado de modo a abranger qualquer tamanho e massa de veículos Omni-Direcionais devido ao método utilizado para a identificação dos parâmetros intrínsecos ao sistema. Estes que se demonstraram condizentes com o sistema real quando verificados em simulação, controlados e permitindo a aplicação do mesmo sistema de controle no veículo real.

O sensoriamento se demonstrou adequado ao sistema proposto pois cobre com eficiência todo o espaço de trabalho e localiza com precisão o posicionamento do veículo. A fusão do sensor de visão com sensores inerciais é uma abordagem análoga ao uso do GPS com sensores inerciais, todavia sua aplicabilidade apresenta vantagens tanto em precisão como funcionamento em ambientes fechados, onde o GPS não apresenta uma performance adequada.

O sistema de controle PTOS apresentou o melhor funcionamento para a problemática proposta devido ao tratamento do sistema como três integradores duplos com atrito. Este controlador foi desenvolvido a fim de obter o melhor desempenho de um sistema integrador duplo, seu uso em conjunto com os três graus de liberdade e mesmo com a presença do atrito ainda demonstraram resultados superiores ao de controladores mais genéricos.

Por fim, o sistema como um todo demonstrou com sucesso sua aplicabilidade tanto em indústrias, atuando como um AGV transportador de cargas como em qualquer outro ambiente onde alta manobrabilidade, agilidade e flexibilidade quanto ao número de veículos e suas trajetórias seja requerido.

Como sugestão para trabalhos futuros existe ainda a possibilidade de incorporar o sensor de visão no próprio veículo (*eye-in-hand*) sem a perda de qualidade na informação. Outros tipos de controladores também podem ser implementados porém será necessária a linearização do modelo ou uma mudança na representação do mesmo para uma Representação Algébrico-Diferencial (DAR) (COUTINHO et al., 2008) ou Representação Linear Fracional (LFR). Essas representações permitirão a síntese de um controlador ainda mais específico para o sistema.

Referências

- CHAUMETTE, F.; HUTCHINSON, S. Visual servo control. i. basic approaches. *Robotics Automation Magazine, IEEE*, v. 13, n. 4, p. 82–90, 2006. ISSN 1070-9932. Citado na página 29.
- COOKE, J. M. et al. Npsnet: Flight simulation dynamic modeling using quaternions. *Presence*, v. 1, p. 404–420, 1992. Citado na página 29.
- COUTINHO, D. et al. 2-gain analysis and control of uncertain nonlinear systems with bounded disturbance inputs. *International Journal of Robust and Nonlinear Control*, John Wiley & Sons, Ltd., v. 18, n. 1, p. 88–110, 2008. ISSN 1099-1239. Disponível em: <<http://dx.doi.org/10.1002/rnc.1207>>. Citado na página 67.
- DUROLA, S. et al. Rational systems and matrix inequalities to the multicriteria analysis of visual servos. In: *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*. [S.l.: s.n.], 2009. p. 1504–1509. ISSN 1050-4729. Citado na página 15.
- GELB, A. *Applied Optimal Estimation*. MIT Press, 1974. ISBN 9780262570480. Disponível em: <<http://books.google.com.br/books?id=KlFrn8lpPP0C>>. Citado na página 45.
- GUSTAFSSON, T. On the design and implementation of a rotary crane controller. *European Journal of Control*, v. 2, n. 3, p. 166 – 175, 1996. ISSN 0947-3580. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0947358096700420>>. Citado na página 30.
- HATA, A. et al. Terrain mapping and classification in outdoor environments using neural networks. *International Journal of u-and e-Service, Science and Technology*, v. 2, p. 51–61, 2009. Citado na página 14.
- HUGHES, P. C. *Spacecraft attitude dynamics*. [S.l.]: Courier Dover Publications, 2012. Citado na página 30.
- HUTCHINSON, C. E. The kalman filter applied to aerospace and electronic systems. *Aerospace and Electronic Systems, IEEE Transactions on*, AES-20, n. 4, p. 500–504, July 1984. ISSN 0018-9251. Citado na página 45.
- ILON, B. *Wheels for a course stable selfpropelling vehicle movable in any desired direction on the ground or some other base*. Google Patents, 1975. US Patent 3,876,255. Disponível em: <<http://www.google.com/patents/US3876255>>. Citado na página 17.
- KIM, K. et al. Teleoperated cleaning robots for use in a highly radioactive environment of the ddf. In: *SICE-ICASE, 2006. International Joint Conference*. [S.l.: s.n.], 2006. p. 3094–3099. Citado na página 13.
- KOENIG, T.; DONG, Y.; DESOUZA, G. Image-based visual servoing of a real robot using a quaternion formulation. In: *Robotics, Automation and Mechatronics, 2008 IEEE Conference on*. [S.l.: s.n.], 2008. p. 216–221. Citado na página 26.

- LIMA, D. A. de. *Navegação segura de um carro autônomo utilizando Campos Vetoriais e o Método da Janela Dinâmica*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais (UFMG), 2010. Citado na página 14.
- LUETTEL, T.; HIMMELSBACH, M.; WUENSCH, H. J. Autonomous ground vehicles - concepts and a path to the future. *Proceedings of the IEEE*, v. 100, n. Special Centennial Issue, p. 1831–1839, May 2012. ISSN 0018-9219. Citado na página 14.
- NAYAR, S. K.; NENE, S. A.; MURASE, H. *Subspace Methods for Robot Vision*. 1995. Citado na página 30.
- PSARROS, D. et al. Frc: A low-cost service robot for subsea flexible risers. In: *Advanced Robotics, 2009. ICAR 2009. International Conference on*. [S.l.: s.n.], 2009. p. 1–6. Citado na página 13.
- QI, H.; MOORE, J. Direct kalman filtering approach for gps/ins integration. *Aerospace and Electronic Systems, IEEE Transactions on*, v. 38, n. 2, p. 687–693, Apr 2002. ISSN 0018-9251. Citado na página 45.
- ROHS, M. Real-world interaction with camera phones. Springer Berlin Heidelberg, v. 3598, p. 74–89, 2005. Citado na página 42.
- SA, I.; CORKE, P. System identification, estimation and control for a cost effective open-source quadcopter. In: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. [S.l.: s.n.], 2012. p. 2202–2209. ISSN 1050-4729. Citado 2 vezes nas páginas 16 e 50.
- SAITO, P. T. M.; SABATINE, R. J. An analysis of parallel approaches for a mobile robotic self-localization algorithm. 2009. Citado na página 14.
- SALTON, A. et al. Preview control of dual-stage actuator systems for superfast transition time. *IEEE/ASME Transactions on Mechatronics*, v. 16, n. 4, p. 758–763, aug. 2011. Citado na página 31.
- SANTOS, M. M. *Desenvolvimento de um Sistema de Localização para um Veículo Terrestre*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais (UFMG), 2009. Citado na página 14.
- SICILIANO, B.; KHATIB, O. *Springer Handbook of Robotics*. [S.l.]: Springer, 2008. (Gale virtual reference library). ISBN 9783540239574. Citado 3 vezes nas páginas 13, 15 e 18.
- UTKIN, V.; LEE, H. Chattering problem in sliding mode control systems. In: *Variable Structure Systems, 2006. VSS'06. International Workshop on*. [S.l.: s.n.], 2006. p. 346–350. Citado na página 32.
- WANG, M. Data fusion of mems imu/gps integrated system for autonomous land vehicle. In: *Information Acquisition, 2006 IEEE International Conference on*. [S.l.: s.n.], 2006. p. 80–84. Citado 2 vezes nas páginas 15 e 45.
- WELCH, R.; LIMONADI, D.; MANNING, R. Systems engineering the curiosity rover: A retrospective. In: *System of Systems Engineering (SoSE), 2013 8th International Conference on*. [S.l.: s.n.], 2013. p. 70–75. Citado na página 13.

- WORKMAN, M. L.; KOSUT, R.; FRANKLIN, G. Adaptive proximate time-optimal servomechanisms: Continuous time case. In: *American Control Conference, 1987*. [S.l.: s.n.], 1987. p. 589–594. Citado na página 32.
- ZHANG, L. et al. Study and realization of vehicle guidance system based on gis and gps. In: *Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on*. [S.l.: s.n.], 2008. v. 1, p. 546–549. Citado na página 15.
- ZHENG, J.; FU, M. Nonlinear feedback control of a dual-stage actuator system for reduced settling time. *Control Systems Technology, IEEE Transactions on*, v. 16, n. 4, p. 717–725, 2008. ISSN 1063-6536. Citado na página 31.
- ZHOU, J.; BOLANDHEMMAT, H. Integrated ins/gps system for an autonomous mobile vehicle. In: *Mechatronics and Automation, 2007. ICMA 2007. International Conference on*. [S.l.: s.n.], 2007. p. 694–699. Citado na página 26.