

Antonio Benjamin Leão de Medeiros

**Controlador PID para posicionamento de um  
modelo veicular elétrico baseado em um sensor  
óptico**

**Manaus  
2014**

Antonio Benjamin Leão de Medeiros

## **Controlador PID para posicionamento de um modelo veicular elétrico baseado em um sensor óptico**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de engenharia de Controle e Automação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Universidade do Estado do Amazonas – UEA  
Escola Superior de Tecnologia – EST  
Programa de Graduação em Engenharia de Controle e Automação

Orientador M.Sc. Charles Luiz Silva de Melo

Manaus  
2014

Antonio Benjamin Leão de Medeiros

## **Controlador PID para posicionamento de um modelo veicular elétrico baseado em um sensor óptico**

Trabalho de Conclusão de Curso apresentado à banca avaliadora do Curso de engenharia de Controle e Automação, da Escola Superior de Tecnologia, da Universidade do Estado do Amazonas, como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Trabalho aprovado. Manaus, 5 de Dezembro de 2014:

---

**M.Sc. Charles Luiz Silva de Melo**  
Orientador

---

**Dr. Walter Andres Vermehren  
Valenzuela**  
Professor

---

**M.Sc. Moisés Pereira Bastos**  
Convidado 1

---

**Dr. Israel Francisco Benitez Pina**  
Convidado 2

Manaus  
2014



# Agradecimentos

Agradeço primeiramente a Deus, ser superior e único, o qual me presenteou com a vida e colocou ao meu lado pessoas maravilhosas, que torceram e lutaram juntas comigo nesta jornada. Aos meus queridos pais, Antonio Benjamin e Maria Lúcia, que são minha força e inspiração para prosseguir sorrindo e feliz, ainda que em meio a adversidades. Exemplos de caráter, amor e dedicação, jamais deixaram de me amparar e me apoiar, sempre me levantando quando caio, caminhando ao meu lado e orando por mim. A Bruna Chacon, pelo apoio e compreensão nos momentos de dificuldades que antecederam o fim deste trabalho, estando ao meu lado e sempre torcendo pelo sucesso deste projeto. A todos meus amigos de graduação, em especial a Lucas Bonetti, Anderson Hiago, Talita Vilela, Erica Kido, Renan Baima, Filipe Máximo e Wellington di Tommaso, Isabelle Stoco, Rafael Goulart, Dilermando Ferreira, que juntos sorrimos e choramos, numa luta diária pelo alcance dos nossos objetivos, deixando de lado quaisquer individualidades, trabalhando como irmãos e parceiros. Aos companheiros Luis Jr., Cristiano Coimbra, Cassio Rodrigo e Rodrigo Santa Rita pela ajuda em diversos momentos, orientando tanto na parte técnica como na formatação deste trabalho. A todos que de forma direta ou indireta me ajudaram, com uma palavra de apoio, um conselho ou uma oração. Obrigado a Coordenação do Curso de Engenharia de Controle e Automação e seus professores, em especial ao meu orientador Charles Melo, que através da Universidade do Estado do Amazonas, deu subsídios para a realização deste projeto. Serei eternamente grato a todos.

# Resumo

Os processos em geral que utilizam veículos automaticamente guiados (AGV - *Automatic Guided Vehicle*) em sua arquitetura sofrem com a falta de flexibilidade de alguns destes dispositivos. Como são os meios mais comuns de transporte em células produtivas, buscam-se maneiras para garantir a flexibilidades destes. Porém, muitas das soluções disponíveis são de alto custo ou de difícil adaptação dos veículos já existentes, seja por incompatibilidade entre os componentes ou da planta. Utilizando sensores ópticos com processamento digital de imagens, será apresentado uma proposta de um AGV cuja dependência em relação ao *layout* onde se encontra seja mínima, bastando apenas modificações na programação para adaptá-lo a outras plantas. As etapas de um projeto de controle serão realizadas, desde a modelagem do sistema até a implementação do controlador PID utilizado, com a análise e discussão dos resultados obtidos.

**Palavras-chaves:** AGV. Sensor Óptico. Controle.

# Abstract

The processes generally using automated guided vehicles (AGV) in its architecture suffer from lack of flexibility of some of these devices . What are the most common means of transport in productive cells are sought ways to ensure these flexibilities . However, many of the available solutions are expensive or difficult to adapt existing vehicles , either incompatibility between components or plant . Using optical sensors and digital image processing , a proposal of an AGV will be presented whose dependence on the layout where is minimal , requiring only changes in the schedule to adapt it to other plants . The steps of a control project will be carried out , since the modeling of the system to implementation PID controller used with the analysis and discussion of results.

**Key-words:**AGV. Optical Sensor. Control.

# Lista de ilustrações

Figura 1 – AGV de trajetória fixa . . . . .	17
Figura 2 – Triangulação a <i>laser</i> . . . . .	17
Figura 3 – Modelo do robô uniclo . . . . .	18
Figura 4 – Controle PID aplicado a uma planta . . . . .	21
Figura 5 – Curva de resposta ao degrau unitário . . . . .	22
Figura 6 – Controle PID digital . . . . .	24
Figura 7 – Robô Pololu 3pi . . . . .	24
Figura 8 – Arduino MEGA 2560 . . . . .	25
Figura 9 – IDE Arduino . . . . .	26
Figura 10 – Topologia com dois escravos . . . . .	27
Figura 11 – Diagrama fase <i>versus</i> polaridade . . . . .	28
Figura 12 – Montagem ADNS 5020 . . . . .	29
Figura 13 – IDE Matlab® . . . . .	31
Figura 14 – Biblioteca Simulink® . . . . .	31
Figura 15 – Diagrama de blocos do controlador . . . . .	32
Figura 16 – Trajetória seguida pelo robô a partir da origem para a circunferência . . . . .	33
Figura 17 – Erro nas coordenadas x e y do robô para trajetória circular . . . . .	33
Figura 18 – Trajetória seguida pelo robô a partir da origem para a leminiscata . . . . .	34
Figura 19 – . . . . .	34
Figura 20 – Diagrama geral do sistema de controle proposto . . . . .	35
Figura 21 – Controlador dinâmico: trajetória leminiscata. (a) Trajetória percorrida pelo robô antes e depois da adaptação de parâmetros. (b) erro da distância com os parâmetros exatos . . . . .	36
Figura 22 – Centro de Curvatura Instantânea no ponto . . . . .	37
Figura 23 – Estrutura de controle do robô móvel . . . . .	37
Figura 24 – Esquema de controle . . . . .	38
Figura 25 – Etapas . . . . .	39
Figura 26 – Gerenciamento da comunicação . . . . .	40
Figura 27 – Pinagem ADNS 5020 . . . . .	40
Figura 28 – Modos de operação SPI . . . . .	42
Figura 29 – Operações de leitura e escrita . . . . .	42
Figura 30 – Diagrama de Blocos . . . . .	43
Figura 31 – Funções de leitura e escrita . . . . .	44
Figura 32 – Registrador <i>Motion</i> . . . . .	44
Figura 33 – Registradores Delta_X e Delta_Y . . . . .	45
Figura 34 – Registrador <i>Mouse_control</i> e parâmetros de resolução. . . . .	45

Figura 35 – Pinagem Pololu . . . . .	46
Figura 36 – Comunicação com o Matlab® . . . . .	47
Figura 37 – Módulo XBee® . . . . .	48
Figura 38 – Resolução aproximada <i>versus</i> distância das lentes ao plano de referência . . . . .	48
Figura 39 – Sentido do movimento . . . . .	49
Figura 40 – Leitura dos registradores de movimento . . . . .	50
Figura 41 – Leitura e cálculo das posições . . . . .	50
Figura 42 – Distância do plano das lentes ao plano de deslocamento . . . . .	51
Figura 43 – Acoplamento . . . . .	51
Figura 44 – Modelo de teste montado . . . . .	52
Figura 45 – Gerador do gráfico . . . . .	53
Figura 46 – Gráfico da trajetória: movimento manual . . . . .	54
Figura 47 – Diagrama de blocos do sistema . . . . .	54
Figura 48 – Entradas . . . . .	55
Figura 49 – Distância ao ponto de interesse . . . . .	55
Figura 50 – Interações . . . . .	55
Figura 51 – Saídas . . . . .	56
Figura 52 – Deslocamento . . . . .	57
Figura 53 – Sistema controlado . . . . .	57
Figura 54 – Respostas para ganho proporcional . . . . .	58
Figura 55 – Resposta para $K=1,3$ . . . . .	58
Figura 56 – Posição $x$ <i>versus</i> tempo . . . . .	59
Figura 57 – Posição $y$ <i>versus</i> tempo . . . . .	60
Figura 58 – Trajetória <i>versus</i> referência . . . . .	61
Figura 59 – Erros de posição $x$ . . . . .	61
Figura 60 – Erro de posição $y$ . . . . .	62
Figura 61 – Velocidade angular <i>versus</i> tempo . . . . .	62
Figura 62 – Velocidade linear <i>versus</i> tempo . . . . .	63
Figura 63 – Respostas do sistema de controle PID do posicionamento $x$ . . . . .	63
Figura 64 – Respostas do sistema de controle PID do posicionamento $y$ . . . . .	64
Figura 65 – Pulsos de distúrbio . . . . .	64
Figura 66 – Velocidade angular com aplicação de distúrbios . . . . .	65
Figura 67 – Trajetória com distúrbios . . . . .	66

# Lista de tabelas

Tabela 1 – Regra de sintonia de Ziegler-Nichols . . . . .	21
Tabela 2 – Características Arduino Mega 2560 . . . . .	26
Tabela 3 – Pinagem ADNS 5020 e descrições . . . . .	41
Tabela 4 – Conexão ADNS 5020 e Arduino . . . . .	42
Tabela 5 – Registradores . . . . .	43
Tabela 6 – Conexão Plolu e Arduino Mega . . . . .	46
Tabela 7 – Especificações do regime transitório . . . . .	60

# Sumário

	<b>Lista de ilustrações</b>	<b>7</b>
	<b>Lista de tabelas</b>	<b>9</b>
	<b>Sumário</b>	<b>10</b>
<b>1</b>	<b>INTRODUÇÃO</b>	<b>12</b>
1.1	Problemática	12
1.2	Motivação	13
1.3	Justificativa	13
1.4	Objetivo geral	14
1.5	Objetivos específicos	14
1.6	Metodologia	14
1.7	Estrutura do trabalho	15
<b>2</b>	<b>AGV'S E A ROBÓTICA MÓVEL</b>	<b>16</b>
2.1	Métodos de navegação	16
2.1.1	Trajectoria fixa	16
2.1.2	Trajectoria dinâmica	17
2.2	Robôs Uniciclos	18
2.3	Controlador PID	20
2.3.1	Sintonia pelo método de Ziegler-Nichols	21
2.3.2	Especificações da resposta transitória	21
2.3.3	Algoritmo de Controle	23
2.4	Robô Pololu 3pi	24
2.5	Microcontrolador	25
2.6	Comunicação SPI	27
2.7	Sensor ADNS 5020	28
2.7.1	Resolução	30
2.8	Matlab®	30
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>32</b>
3.1	Controle não linear de robô móvel uniciclo utilizando linearização entrada-saída por realimentação	32
3.2	Modelagem e compensação da dinâmica de robôs móveis e sua aplicação em controle de formação	35

---

<b>3.3</b>	<b>Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida . . . . .</b>	<b>36</b>
<b>3.4</b>	<b>Controle de Trajetória de Robôs Móveis Omni-direcionais: Uma Abordagem Multivariável . . . . .</b>	<b>37</b>
<b>4</b>	<b>MATERIAIS E MÉTODOS . . . . .</b>	<b>39</b>
<b>4.1</b>	<b>Comunicação . . . . .</b>	<b>39</b>
4.1.1	ADNS 5020 - Arduino . . . . .	40
4.1.1.1	Registradores . . . . .	43
4.1.2	Arduino - Pololu . . . . .	46
4.1.3	Arduino - PC . . . . .	46
<b>4.2</b>	<b>Cálculo do posicionamento . . . . .</b>	<b>47</b>
<b>4.3</b>	<b>Montagem e acoplamento . . . . .</b>	<b>50</b>
<b>5</b>	<b>SIMULAÇÕES E ANÁLISE DOS RESULTADOS . . . . .</b>	<b>53</b>
<b>5.1</b>	<b>Aquisição de Dados . . . . .</b>	<b>53</b>
<b>5.2</b>	<b>Descrição do modelo . . . . .</b>	<b>53</b>
<b>5.3</b>	<b>Sistema de controle proposto . . . . .</b>	<b>56</b>
5.3.1	Determinação dos parâmetros dos controladores . . . . .	57
<b>5.4</b>	<b>Aplicação do controle . . . . .</b>	<b>59</b>
5.4.1	Distúrbios . . . . .	64
<b>6</b>	<b>CONCLUSÕES . . . . .</b>	<b>67</b>
<b>6.1</b>	<b>Dificuldades Encontradas . . . . .</b>	<b>68</b>
<b>6.2</b>	<b>Trabalhos Futuros . . . . .</b>	<b>68</b>
	<b>Referências . . . . .</b>	<b>69</b>

# 1 Introdução

Nos últimos anos, o interesse no desenvolvimento de tecnologias para veículos guiados automaticamente (AGV - *Automatic Guided Vehicle*) vêm gerando novos métodos e dispositivos de controle embarcados para a automação industrial. Desde simples aplicações de translação de materiais até implementações que envolvam lógicas mais complexas, as melhorias realizadas têm diminuído riscos de acidentes e aumentado a eficiência quanto ao consumo de energia elétrica e tempos de traslado (CANO, 2006).

Neste contexto há uma necessidade de sistemas de produção flexíveis e de caráter antropocêntrico e, ainda, com ferramentas de diagnóstico de fácil interação com os operadores das células produtivas. A redução dos custos dos investimentos em automação para viabilizar projetos que necessitam quase integralmente de importação de tecnologias, também tem sido um desafio para projetistas e engenheiros que tentam atender a essa necessidade (BARROS, 2006).

Embora os estudos para a melhoria desses processos sejam constantes, ainda há uma dependência clara destes dispositivos em relação ao *layout* dos processos fabris onde estão instalados.

Um forte aliado dessas tecnologias são os dispositivos de controle, executando as funções lógicas e de predição dos erros de posicionamento que porventura ocorram durante as operações destes sistemas.

Segundo Ogata (2010), o controle automático é essencial em qualquer campo da engenharia e da ciência; é um componente importante e intrínseco em sistemas de veículos espaciais, sistemas robóticos, modernos sistemas de manufatura e quaisquer operações industriais que envolvam o controle de grandezas, como velocidade e posição.

O controle de sistemas complexos exige um esforço maior na execução do projeto, pois para compreender e controlar esses sistemas, deve-se obter modelos matemáticos quantitativos destes. Torna-se necessário, por conseguinte, analisar as relações entre as variáveis do sistema e obter um modelo matemático (DORF, 1998).

Assim, unindo todos os elementos envolvidos, tem-se a definição de um sistema embarcado, como a capacidade computacional dentro de um circuito integrado, equipamento ou sistema, sendo assim um sistema completo e independente, mas preparado para realizar uma determinada tarefa (CUNHA, 2007).

## 1.1 Problemática

Seja por meio de trilhos ou sensores posicionados na planta, mudanças físicas no ambiente devem ser feitas previamente para a adequação dos AGV's. Tais mudanças

tornam o sistema pouco flexível, acarretando complicações nas mudanças de *layout* que venham ocorrer nos processos de produção.

De acordo com Goes (2013), a aplicação de robôs móveis requer uma abordagem robusta e confiável que desempenhe atividades pré-determinadas, com segurança e desempenho satisfatórios. Esta definição se completa com o conceito de flexibilidade, pois não basta ser apenas seguro e confiável, mas também deve ser flexível ao ponto de serem adaptados a novos percursos.

Com base no descrito acima, este trabalho visa à implementação de um sistema auto guiado para AGV's que independa de meios físicos fixos na planta de trabalho, garantindo não só a robustez e a segurança, como também a flexibilidade em mudanças de processo.

## 1.2 Motivação

A crescente utilização de robôs autônomos em diversas aplicações, principalmente na exploração de locais onde há risco de vida, faz com que muitas pesquisas e competições surjam no intuito de incentivar estudantes de engenharia e áreas afins a desenvolver conhecimento sobre o tema (E.SILVA, 2006).

Dentro deste contexto, o maior exemplo de aplicação e utilização é o seguidor de linha, que usa sensores fotossensíveis para se localizar e percorrer um caminho traçado (DIAS, 2012).

Esses robôs surgiram inicialmente em aplicações de suporte em atividades acadêmicas, mas hoje já se mostram bem úteis em aplicações industriais, lazer e entretenimento.

Um robô seguidor de linha é um exemplo prático e compacto de um AGV industrial, pois segue uma referência fixa na superfície de deslocamento e apresenta alta confiabilidade com algoritmos que conseguem manter parâmetros de percurso, estabilidade e oscilação em limites aceitáveis. Logo, adaptar um robô desta natureza para um método mais independente de referência serve de modelo para implementações em escalas maiores.

## 1.3 Justificativa

Segundo Oliveira (2011), o sistema guia nos AGV's utiliza métodos pelos quais seus caminhos são definidos e controlados para seguirem tais caminhos. As três tecnologias mais utilizadas nesses sistemas são:

- Cabo guiado;
- Rastreamento de caminhos desenhados no chão;
- Veículos autoguiados.

Os dispositivos mais modernos utilizam o método autoguiado para deslocamento, onde não há uma dependência fixa com relação ao meio físico. Esse tipo de tecnologia apresenta uma grande vantagem em relação às demais por serem flexíveis (BORENSTEIN, 1995) no que diz respeito às mudanças de *layout*.

Contudo, é esperado que a exatidão do posicionamento do veículo seja afetada quando as distâncias percorridas são maiores. Assim, a localização do AGV deve ser periodicamente verificada, comparando as posições calculadas com as conhecidas (OLIVEIRA, 2011).

Apresentar-se-á neste trabalho uma proposta de um modelo de AGV utilizando sensor óptico com leitura constante de posicionamento, propondo a independência em relação à planta onde está instalado e a aplicação de um controlador PID para o seu posicionamento, bastando apenas mudanças nos parâmetros de programação para adaptar-se a novos *layouts*.

## 1.4 Objetivo geral

Implementar um sistema utilizando um modelo veicular elétrico associado a um sensor óptico com processamento digital de imagens e controle PID digital microcontrolado, realizando simulações e testes que serão feitos em superfície lisa e rota pré-definida.

## 1.5 Objetivos específicos

- Acoplar o sensor ao robô e gerar as condições de funcionamento;
- Obter as leis de controle e a modelagem matemática de caráter cinemático do modelo;
- Garantir a comunicação do *hardware* envolvido (sensor, microcontrolador, robô e computador);
- Projetar controlador e gerar o algoritmo de controle;
- Realizar simulações e testes para validar modelo;

## 1.6 Metodologia

- Levantamento bibliográfico nas bases literárias, sobre trabalhos relacionados ao tema proposto;
- Estudo das técnicas de modelagem do sistema utilizado;
- Obtenção dos dados relacionados aos dispositivos utilizados no modelo de teste;

- Projetar ambiente de teste;
- Análise do desempenho do sistema com e sem controle;

## 1.7 Estrutura do trabalho

A estrutura do trabalho está composta em seis capítulos dispostos da seguinte maneira:

No capítulo 1 é exposto uma introdução do trabalho, suas justificativas, motivações, objetivo a ser alcançado e a metodologia a ser utilizada. No fim do mesmo há uma visão organizacional do projeto.

No capítulo 2 o referencial teórico é explorado, com a contextualização do problema e das teorias a serem utilizadas no desenvolvimento, bem como a descrição das tecnologias utilizadas.

No capítulo 3 serão apresentados quatro trabalhos que estão relacionados com o projeto e os pontos importantes para a realização do mesmo. É descrito os métodos utilizados pelos autores e os resultados por eles obtidos.

No capítulo 4 são apresentados os materiais utilizados e como os mesmos se relacionam, abordando as comunicações e os algoritmos implementados para garantir o funcionamento do sistema.

No capítulo 5 são realizadas simulações e testes e o estudo das respostas. Parâmetros de desempenho são inclusos nas análises para a verificação da efetividade do modelo empregado.

No capítulo 6 são feitas as conclusões do projeto referentes aos dados obtidos, além de expor as dificuldades encontradas e as sugestões para melhorias em trabalhos futuros.

## 2 AGV's e a Robótica Móvel

Para Nascimento (2009), robótica móvel é a área da robótica que agrupa os robôs que não possuem base fixa, ou seja, que podem se locomover dentro de um espaço limitado ou não. É por meio dessa locomoção que o robô deve se reconhecer como objeto móvel, armazenar sua posição dentro desse espaço, compreender os limites físicos deste espaço e se adaptar prevendo caminhos e objetos fixos, ou móveis, que podem ser obstáculos a serem desviados.

Um veículo guiado automaticamente (AGV - *Automated Guided Vehicle*) é um robô móvel com uma série de sensores e circuitos de comunicação que o torna autônomo, ou seja, capaz de executar sua tarefa independente, sem a supervisão humana (SIDNEI, 2013).

Os benefícios teóricos de utilizar robôs em uma indústria são numerosos e vão desde o aumento da produtividade, a melhoria e a consistência na qualidade final do produto, a menor demanda de concentração de mão de obra especializada, que é difícil de encontrar, a confiabilidade no processo, a facilidade na programação e uso de robôs, a operação em ambientes difíceis e perigosos ou em tarefas desagradáveis e repetitivas para o ser humano e, finalmente, a capacidade de trabalho sem descanso por longos períodos (FILHO, 1999).

Assim, além dos benefícios, o crescente interesse por robôs autônomos se deve à grande diversidade de tarefas que podem ser por eles realizadas. O problema de controle de robôs móveis exhibe dois subproblemas principais: navegação, que designa a determinação de posição e orientação do veículo em um dado instante de tempo e guiagem, que se refere ao controle da trajetória (SANDI, 1998).

### 2.1 Métodos de navegação

Atualmente nas plantas industriais, é comum a utilização de sistemas que utilizam algum tipo de guia para o deslocamento dos robôs pelo chão-de-fábrica, como canaletas, fitas refletoras nas paredes e fitas magnéticas ou coloridas no piso (SIDNEI, 2013).

Santos (2013) define dois tipos elementares de AGV's no que se refere ao método de navegação: os que apresentam trajetória fixa e os que possuem trajetória dinâmica.

#### 2.1.1 Trajetória fixa

Estes tipos de AGV's apresentam a desvantagem de serem pouco flexíveis a mudanças estruturais no meio em que se encontram, pois precisam de um método fixo de referência, seja por fitas magnéticas ou linhas pintadas no chão (faixas) ou por contatos elétricos (filoguiados), sendo o primeiro mais flexível, pois basta apenas a mudança de faixas para

mudar um *layout* específico. O método filoguiado, exige uma mudança na posição dos contatos e terminais elétricos da planta, e pode ser visto na figura 1.



Figura 1 – AGV de trajetória fixa

Fonte: <http://www.logismarket.es/ip/>

### 2.1.2 Trajetória dinâmica

Ao contrário dos AGV's de trajetória fixa, esses sistemas apresentam possibilidade de serem adaptados a novas rotas, sendo os mais flexíveis e confiáveis, mas também os mais caros. Como por exemplo, tem-se os sistemas de triangulação a *laser*, conforme exibido na figura 2, utilizam refletores espalhados pelo ambiente de locomoção do AGV, onde receptores recebem e memorizam as posições. São necessários três pontos para determinar a posição do veículo.

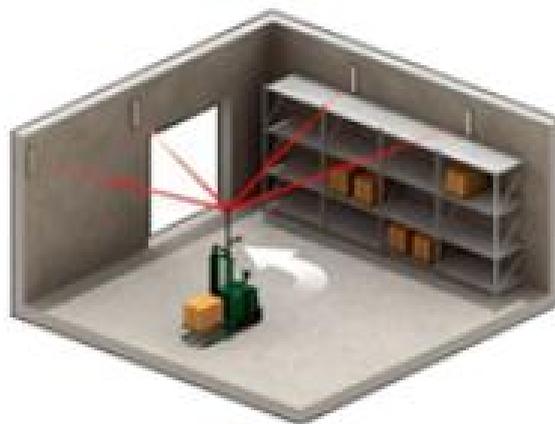


Figura 2 – Triangulação a *laser*

Fonte: Santos (2013)

## 2.2 Robôs Uniciclos

Segundo Ferreira, Cerqueira e Costa (2012), para movimentar-se em um ambiente e, por conseguinte, ser caracterizado como móvel, um robô deve ter seu movimento controlado. Assim ele poderá seguir trajetórias ou mover-se com exatidão para um ponto determinado. Para tanto, deve-se projetar um controlador que, agindo sobre os atuadores, ajuste uma ou mais variáveis de estado do sistema para um valor desejado. O conhecimento do modelo matemático da planta é importante para a seleção da técnica de projeto de controlador a ser usada.

Robôs uniciclos são definidos por apresentarem apenas duas rodas de tração e um ponto fixo de apoio para o deslocamento. O movimento é determinado pela diferença de velocidade entre as rodas. A figura 3 é a representação desse tipo do robô, onde  $u$  e  $\omega$  são, respectivamente, as velocidades linear e angular desenvolvidas pelo robô,  $G$  é seu centro de massa,  $C$  é a posição de sua roda castor,  $E$  é a posição de uma ferramenta a bordo do robô (que aqui não é considerada),  $h$  é o ponto de interesse (de coordenadas  $x$  e  $y$  no plano  $XY$ ),  $\psi$  é a orientação do robô, e  $a$  é a distancia entre o ponto de interesse e o ponto central do eixo virtual que conecta as rodas de tração (ponto  $B$ ) Martins (2009) .

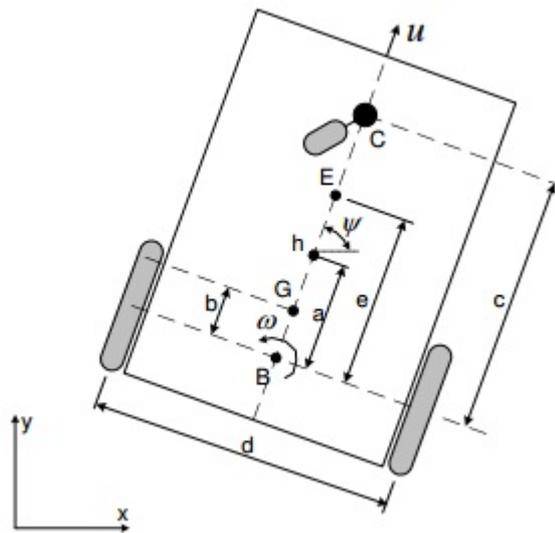


Figura 3 – Modelo do robô uniciclo

Fonte: Martins (2009)

Para executar a modelagem matemática de robôs, tanto as características dinâmicas quanto as cinemáticas devem ser consideradas. Na primeira, as propriedades físicas de interação com o meio são abordadas, como centro de massa, inércia, atrito, etc. Já na segunda, apenas as características geométricas do robô são consideradas.

O modelo abaixo representa o sistema completo de um robô móvel, com sua dinâmica e cinemática, descrito por CRUZ (2006):

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \\ \dot{u} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} u \cos \psi - a\omega \sin \psi \\ u \sin \psi + a\omega \cos \psi \\ \omega \\ \frac{\theta_3^0}{\theta_1^0} r^2 \omega^2 - 2 \frac{\theta_4^0}{\theta_1^0} u \\ -2 \frac{\theta_3^0}{\theta_2^0} r^2 u \omega - \frac{\theta_4^0}{\theta_2^0} d^2 \omega \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{2r}{\theta_1^0} & 0 \\ 0 & \frac{2rd}{\theta_2^0} \end{bmatrix} \begin{bmatrix} v_u \\ v_\omega \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \\ \bar{\delta}_u \\ \bar{\delta}_\omega \end{bmatrix} \quad (2.1)$$

onde  $\delta = [\delta_x \ \delta_y \ 0 \ \bar{\delta}_u \ \bar{\delta}_\omega]^T$  é o vetor de incertezas associado ao robô móvel relacionado às características dinâmicas (velocidades de deslizamento e de orientação, parâmetros físicos como inércia, massa, diâmetros das rodas e pneus, parâmetros dos servomotores, forças das rodas, etc.).

$v_u$  e  $v_\omega$  são as velocidades linear e angular desenvolvidas pelo robô em termos de tensões aplicadas nas rodas:

$v_u = \frac{v_l + v_r}{2}$  e  $v_\omega = \frac{v_l - v_r}{2}$ , com  $v_r$  e  $v_l$  tensões nas rodas direita e esquerda, respectivamente.

Os parâmetros  $\theta$  são encontrados por meio da identificação do sistema e dependem das características físicas do robô, e sua aplicação é voltada para uma análise dinâmica do sistema:

$$\theta_1^0 = \frac{Ra}{ka} (mr^2 + 2I_e),$$

$$\theta_2^0 = \frac{Ra}{ka} (I_e d^2 + 2r^2 (I_z + mb^2)),$$

$$\theta_3^0 = \frac{Ra}{ka} mb,$$

$$\theta_4^0 = \frac{Ra}{ka} \left( \frac{kakb}{Ra} + Be \right),$$

onde  $Ra$  é a resistência elétrica dos motores,  $kb$  é sua constante eletromotriz,  $ka$  é sua constante de torque,  $Be$  é o coeficiente de atrito,  $m$  é a massa do robô,  $I_z$  é seu momento de inércia no ponto  $G$ ,  $I_e$  é o momento de inércia de cada grupo motorredução-roda,  $r$  é o raio de suas rodas e  $b$  e  $d$  são as distâncias indicadas na figura 3.

A abordagem desses parâmetros se torna eficiente quando se tem o controle dos motores pela tensão aplicada em suas bobinas. Contudo, nem sempre se tem essa liberdade.

Como uma representação alternativa deste modelo, os parâmetros  $\theta$  podem ser representados em função das características de controle dos motores, comumente encontrado em robôs comerciais, onde os valores de tensão aplicados nos motores não podem ser alterados diretamente. Neste caso, os valores de tensão são denotados em função de ganhos de controladores PD (Proporcional-Derivativo) utilizados pelos fabricantes, em que CRUZ (2006) define com ganhos proporcionais  $k_{PT} > 0$  e  $k_{PT} > 0$ , e derivativos  $k_{DT} \geq 0$  e  $k_{DR} \geq 0$ .

O modelo pode ser desmembrado e abordado de forma separada, assim tem-se o modelo cinemático na equação 2.2 :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \end{bmatrix} \quad (2.2)$$

onde

$$u = \frac{1}{2}[r(\omega_r + \omega_l)] \quad (2.3)$$

e

$$\omega = \frac{1}{d}[r(\omega_r - \omega_l)] \quad (2.4)$$

, sendo  $\omega_l$  e  $\omega_r$  as velocidades angulares das rodas esquerda e direita, respectivamente.

## 2.3 Controlador PID

Um controlador proporcional-integral-derivativo (PID) combina as vantagens do controlador proporcional-derivativo (PD) e do proporcional-integrativo (PI). Ou seja, a parte integral está relacionada diretamente com a precisão do sistema e a parte derivativa torna o sistema mais rápido devido a sua característica antecipatória (GOES, 2013).

Segundo Ogata (2010), a utilidade dos controladores PID está na sua aplicabilidade geral à maioria dos sistemas de controle. Em particular, quando o modelo matemático da planta não é conhecido e, portanto, métodos de projeto analítico não podem ser utilizados, controles PID se mostram os mais úteis. Na área dos sistemas de controle de processos, sabe-se que os esquemas básicos de controle PID e os controles PID modificados provaram sua utilidade conferindo um controle satisfatório, embora em muitas situações eles possam não proporcionar um controle ótimo.

A equação 2.5 descreve este tipo de controlador:

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t) dt + K_p T_d \frac{de(t)}{dt} \quad (2.5)$$

onde tem-se que:

$u(t)$ : Sinal de saída do controlador;

$e(t)$ : Sinal de erro do sistema;

$K_p$ : Constante proporcional ajustável;

$T_i$ : Tempo integrativo;

$T_d$ : Tempo derivativo.

O controlador PID é posicionado em série com a planta ou processo a ser controlado, conforme exibido na figura 4.

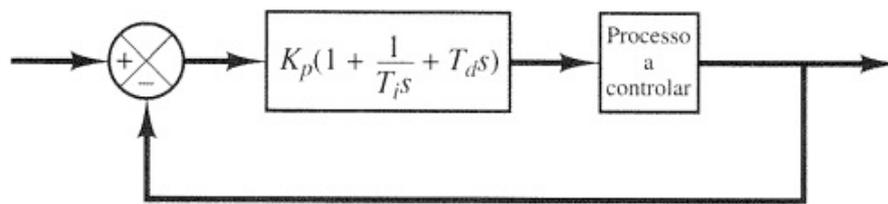


Figura 4 – Controle PID aplicado a uma planta

Fonte: Ogata (2010)

### 2.3.1 Sintonia pelo método de Ziegler-Nichols

O processo de selecionar parâmetros do controlador que garantam dada especificação de desempenho é conhecida como sintonia do controlador. Ziegler e Nichols sugeriram regras para a sintonia de controladores PID (o que significa ajustar os valores de  $K_p$ ,  $T_i$  e  $T_d$ ) baseadas na resposta experimental ao degrau ou no valor de  $K_p$  que resulta em uma estabilidade marginal, quando somente uma ação proporcional é utilizada (OGATA, 2010).

Utilizando o método do ganho crítico, apenas a ação proporcional é empregada no modelo. Segundo Ogata (2010), o valor de  $K_p$  é aumentado de zero até um valor crítico  $K_{cr}$ , no qual a saída exibe uma oscilação sustentada pela primeira vez. Caso a saída não exiba uma oscilação sustentada para qualquer valor que  $K_p$  possa assumir, este método não pode ser realizado.

O valor de  $K_{cr}$  (ganho crítico) gera uma resposta na saída do sistema com um período  $P_{cr}$ , (período crítico) e estes dados são utilizados para o cálculo dos parâmetros dos controladores. Ziegler e Nichols sugeriram escolher os valores dos parâmetros  $K_p$ ,  $T_i$  e  $T_d$  de acordo com as fórmulas da tabela 1.

Fonte: Ogata (2010)

Tipo de Controlador	$K_p$	$T_i$	$T_d$
P	$0,5K_{cr}$	$\infty$	0
PI	$0,45K_{cr}$	$\frac{1}{1,2}P_{cr}$	0
PID	$0,6K_{cr}$	$0,5P_{cr}$	$0,125P_{cr}$

Tabela 1 – Regra de sintonia de Ziegler-Nichols

### 2.3.2 Especificações da resposta transitória

A resposta de um sistema no domínio do tempo apresenta duas situações distintas: regime transitório e regime permanente. Para Ogata (2010), as características de desempenho de um sistema de controle são especificadas em termos de resposta transitória a

uma entrada em degrau unitário, já que se trata de uma entrada suficientemente brusca e gerada com facilidade.

Antes de atingir o regime permanente, a resposta transitória de um sistema de controle apresenta, frequentemente, oscilações amortecidas. Na especificação das características das respostas transitórias de um sistema de controle a uma entrada em degrau, é especificado o seguinte e exibido na figura 5:

1. Tempo de atraso,  $t_d$ : trata-se do tempo requerido para que a resposta alcance metade de seu valor final pela primeira vez;
2. Tempo de subida,  $t_r$ : é o tempo necessário para que a resposta passe de 10% a 90%, ou de 5% a 95%, ou de 0% a 100% do valor final.
3. Tempo de pico,  $t_p$ : é o tempo para que a resposta atinja o primeiro pico de sobressinal;
4. Máximo sobressinal (*overshoot*),  $M_p$ : é o valor máximo de pico da curva de resposta, medido a partir da unidade. Se o valor final da resposta em regime permanente diferir da unidade, então é comum utilizar porcentagem máxima de sobressinal, definida por:

$$\frac{c(t_p) - c(\infty)}{c(\infty)} \quad (2.6)$$

, onde  $c(t)$  é o valor da resposta do sinal em função do tempo.

5. Tempo de acomodação,  $t_s$ : é o tempo necessário para que a curva de resposta alcance valores em uma faixa (geralmente de 2% a 5%) em torno do valor final.

A resposta típica de um sistema a uma entrada em degrau é exibida na figura 5.

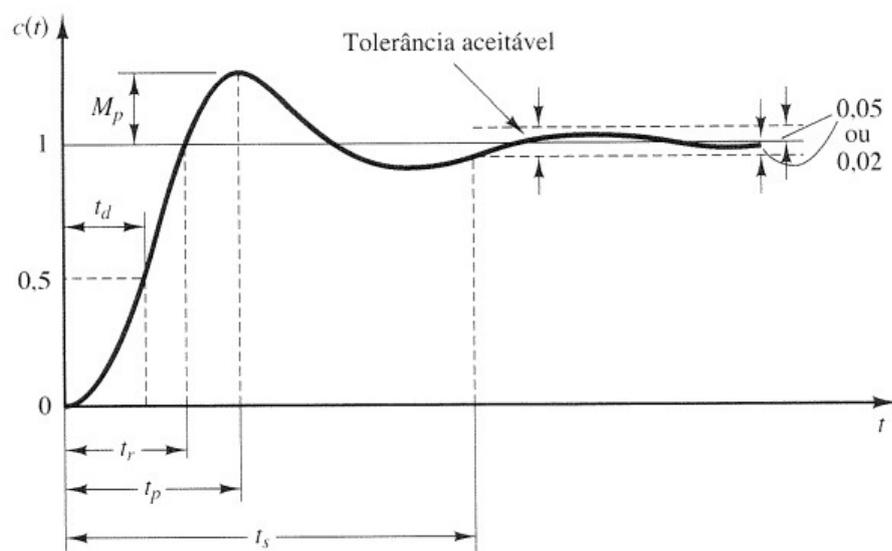


Figura 5 – Curva de resposta ao degrau unitário

Fonte: Ogata (2010)

### 2.3.3 Algoritmo de Controle

O controle PID é a soma de três componentes calculadas em paralelo com respeito à função de erro: componente proporcional, integrativa e derivativa. A componente proporcional determina a relação do controlador ao erro atual; a componente integrativa leva em conta a soma de recentes erros; e a componente derivativa atua usando a taxa de variação do erro no decorrer do tempo. Cada uma dessas componentes tem seu propósito e está associada a toda uma teoria de controle relacionada às suas características (GOMES; SANTOS; OLIVEIRA, 2010).

No domínio de Laplace, a função de transferência do controlador pode ser descrita, derivada da equação 2.5, como:

$$G(s) = K_p \left[ 1 + \frac{1}{sT_i} + sT_d \right] \quad (2.7)$$

Sabendo que:  $\mathcal{L}^{-1}\{sF(s)\} = \frac{df(t)}{dt}$  e discretizando, tem-se que:

$$\mathcal{L}^{-1}\{sF(s)\} = \frac{df(t)}{dt} \rightarrow \frac{F[k] - F[k-1]}{T_s} \quad (2.8)$$

Aplicando a transformada Z na equação 2.8

$$\frac{F[k] - F[k-1]}{T_s} \rightarrow F[k] \frac{1 - z^{-1}}{T_s} \quad (2.9)$$

Assim, define-se a equação de saída do controlador:

$$u[k] = e[k]K_p \left( 1 + \frac{T_s}{T_i(1 - z^{-1})} + T_d \frac{1 - z^{-1}}{T_s} \right) \quad (2.10)$$

Finalmente a equação 2.10 pode ser descrita para uma forma implementável em microcontroladores:

$$u[k] = u[k-1] + K_p(e[k] - e[k-1]) + K_p T_s \frac{e[k]}{T_i} + K_p \frac{T_d}{T_s} (e[k] - 2e[k-1] + e[k-2]) \quad (2.11)$$

Onde  $T_s$  é o tempo de amostragem adotado.

Na figura 6 é exibido o diagrama de blocos do controle PID digital, onde os ADC e DAC são conversores Analógico-Digital e Digital-Analógico.

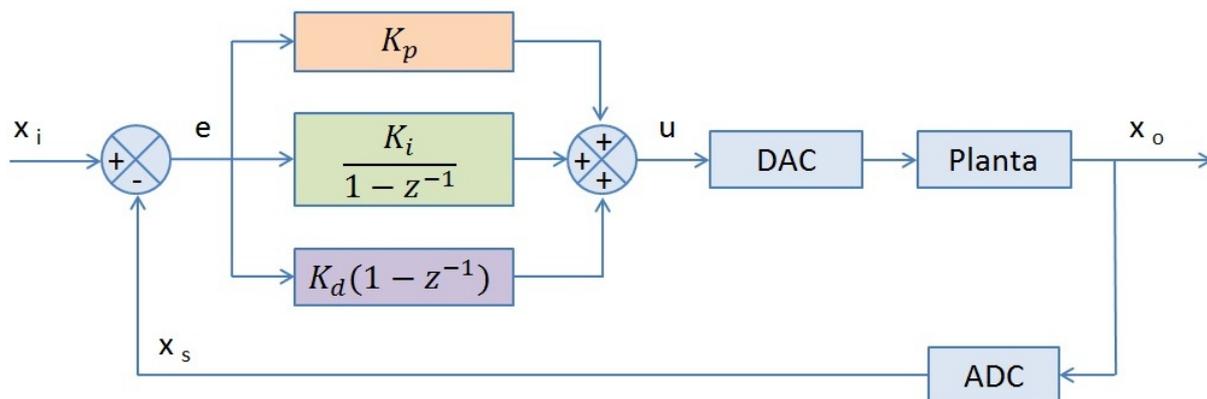


Figura 6 – Controle PID digital

Fonte: [http://www.filipelbc.org/prototype\\_chapters.php](http://www.filipelbc.org/prototype_chapters.php)

## 2.4 Robô Pololu 3pi

Um robô seguidor de linha é caracterizado pelo emprego de mecânica, transdutores que variam em número dependendo da complexidade do projeto e eletrônica. Por isso, se encontram no campo da robótica móvel cujo funcionamento se baseia no movimento, sensores e controladores que os permitem tomar decisões. Sua principal função é seguir uma linha marcada no solo (GRANJA; RODRÍGUEZ, 2012).

O robô utilizado é o Pololu 3pi, concebido para realizar funções de seguidor de linha. Seu uso foi definido pela versatilidade e características de programação, além de propositalmente demonstrar que é possível adaptar dispositivos que inicialmente utilizam um meio de referência fixa para uma orientação dinâmica. Na figura 7 é exibido a forma como o 3pi se apresenta sem modificações.

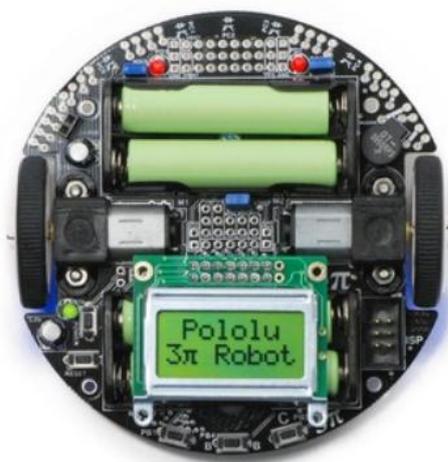


Figura 7 – Robô Pololu 3pi

Fonte: Pololu (2014)

Por apresentar características intrínsecas de um seguidor de linha, representa um modelo de um AGV industrial, principalmente pela presença de sensores fotoelétricos que indicam a presença ou não de linha demarcada na superfície de deslocamento. Alimentado por quatro baterias AAA e um único sistema de potência que fornece 9,25 V para os dois motores, o robô 3pi atinge a velocidade de 100 cm/s enquanto executa curvas e rotações com precisão, não sofrendo influência da variação de tensão das baterias. Isso resulta em alto desempenho nas atividades repetitivas, mesmo com as baterias em níveis baixos. Utiliza dois motores DC com caixa de redução, e seu processamento é feito por um microprocessador ATmega328, de 20 MHz. A memória interna é de 32KB e 1KB de memória volátil. O uso deste processador o torna compatível com a popular plataforma de programação Arduino. Pode ser programado em C/C++, e uma extensa biblioteca de funções existentes estão disponíveis de forma livre para o desenvolvimento de aplicações diversas.

## 2.5 Microcontrolador

O controle de equipamentos para acionamentos industriais é comumente realizado por intermédio de um microprocessador embarcado no produto, o qual carrega um programa armazenado composto de algoritmos dedicados à aplicação em questão [...]. O tipo de microprocessador que melhor se presta a esta categoria de aplicações costuma ser aquele que integra na mesma pastilha de silício, além da unidade central de processamento (CPU), também circuitos de memória e uma diversidade de circuitos auxiliares (periféricos) dedicados a funções de entrada e saída (E/S) específicas, tais como conversão analógico-digital (A/D) e saídas digitais moduladas por largura de pulso (PWM). Tal tipo de processador é usualmente chamado de microcontrolador (STEPHAN, 2013).

O Arduino MEGA 2560, exibido na figura 8, é um microcontrolador baseado no microprocessador ATmega2560, da Atmel. Apresenta em sua arquitetura 54 portas digitais, sendo que 15 delas podem ser usadas como PWM, 4 *hardwares* de porta serial, frequência de operação gerada por um cristal de 16MHz, conexão USB e ICSP. Pode ser alimentado pela conexão USB com o PC ou ainda por uma bateria ou adaptador AC/DC.

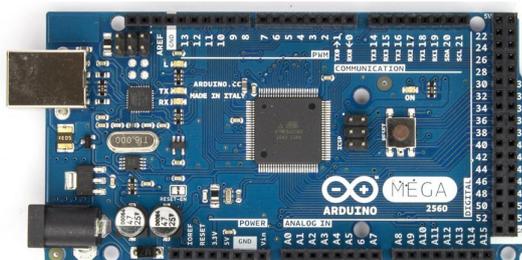


Figura 8 – Arduino MEGA 2560

Fonte: <http://arduino.cc/en>

Segue um resumo de suas características básicas na tabela 2:

Item	Característica
Microprocessador	ATmega2560
Tensão de operação	5 V
Tensão de entrada recomendada	7-12V
Limites de tensão	6-20V
I/O's Digitais	54 (15 como PWM)
Entradas analógicas	16
Corrente DC por I/O	40mA
Memória <i>Flash</i>	256KB
SRAM	8KB
EEPROM	4KB
Velocidade de <i>Clock</i>	16MHz

Tabela 2 – Características Arduino Mega 2560

A IDE (*Integrated Development Environment*) é um ambiente de programação que permite a integração do *hardware* com o *software*. A Arduino oferece um IDE na plataforma de *software* livre, podendo ser adquirida sem custos no site da fabricante. Sua tela principal é exibida por meio da figura 9.

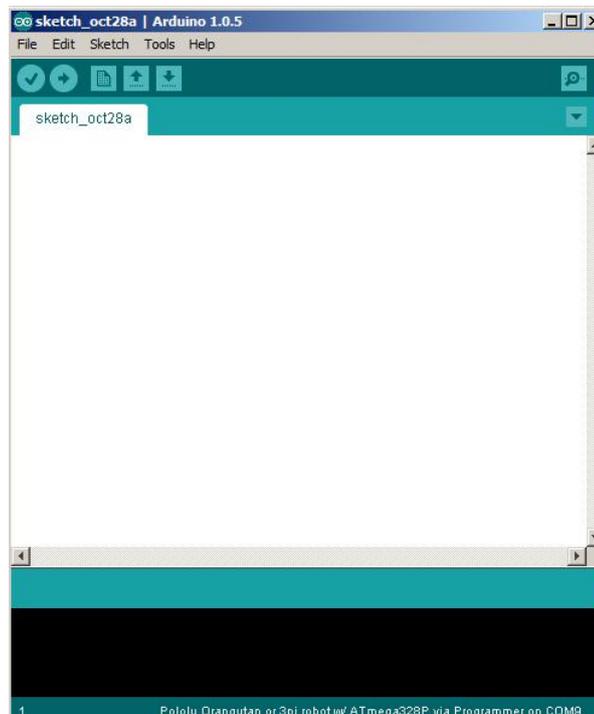


Figura 9 – IDE Arduino

Esta interface permite a configuração das portas de comunicação do *hardware*, o protocolo de gravação do programa e a inserção de bibliotecas para funções específicas. Outra função que será utilizada é o canal serial, que exhibe em tempo real os dados enviados e recebidos entre o microcontrolador e o computador.

## 2.6 Comunicação SPI

O barramento SPI (*Serial Peripheral Interface*) é um barramento de comunicação série, que permite o funcionamento no modo *full-duplex*. Este é constituído por 4 linhas, SCK (*Serial Clock*), MOSI (*Master Output Slave Input*), MISO (*Master Input Slave Output*) e SS (*Slave Select*). Por outro lado, este barramento funciona numa arquitetura mestre-escravo, em que o mestre controla todas as comunicações efetuadas (FERREIRA, 2012).

Cada uma dessas linhas apresenta uma função específica: SCK: é o sinal que é usado para sincronizar a transferência dos dados (bits) na comunicação e é gerado pelo mestre da comunicação. MOSI: é o canal por onde os dados são enviados do mestre para o escravo. MISO: é o canal que o mestre utiliza para receber os dados do escravo. SS: é utilizado para selecionar o escravo, visto que esta comunicação permite a conexão de vários escravos, conforme exibido na figura 10.

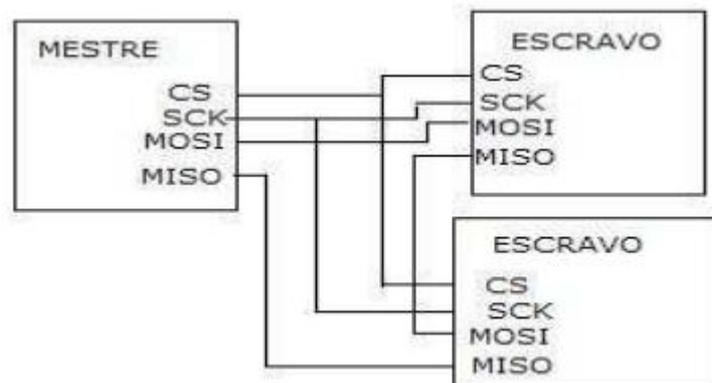


Figura 10 – Topologia com dois escravos

Fonte: Ferreira (2012)

Utilizando a arquitetura mestre-escravo, toda transmissão de dados, seja de leitura ou de escrita, é iniciada pelo mestre. Deve-se, primeiramente, configurar a velocidade da comunicação, com frequências que podem chegar a 70MHz. Então, SS deve ser colocado em nível baixo para habilitar a comunicação com o escravo. Esta seleção permite uma comunicação *full-duplex*, ou seja, comunicação nos dois sentidos de forma simultânea. A polaridade do sinal de *clock* (CPOL) e a fase (CPHA) definem em qual momento do pulso

os bits serão trafegados. Exibe-se na figura 11 a transferência dos bits com relação *clock* e fase.

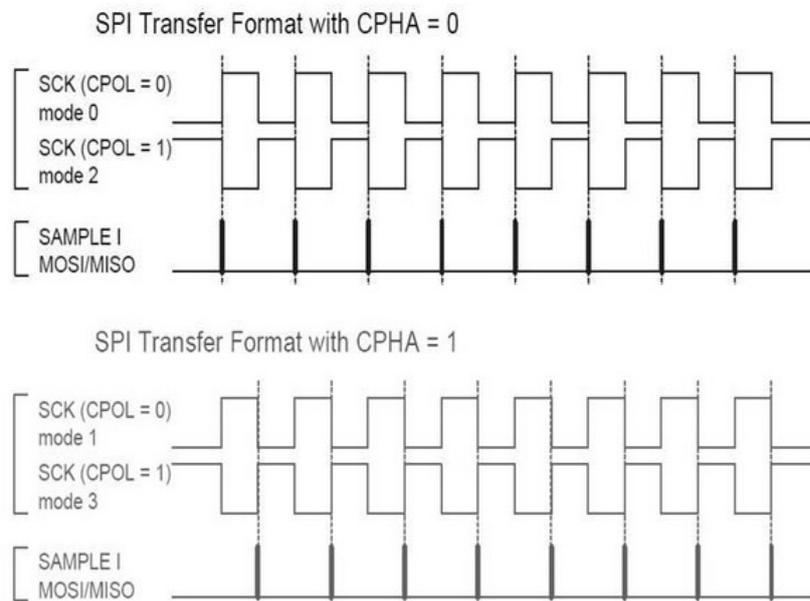


Figura 11 – Diagrama fase *versus* polaridade

Fonte: <http://www.embarcados.com.br/spi-parte-1/>

O modo de operação do periférico deve ser conhecido para definir qual polaridade será utilizada. Abaixo segue o resumo desta característica segundo (FERREIRA, 2012):

Para  $CPOL = 0$ : valor base do sinal de relógio é zero:

- $CPHA=0$  – Os dados são lidos à transição positiva do sinal de relógio e devem ser escritos na transição negativa;
- $CPHA=1$  – Os dados são lidos à transição negativa do sinal de relógio e devem ser escritos na transição positiva.

Para  $CPOL=1$ : valor base do sinal de relógio é um:

- $CPHA=0$  – Os dados são lidos à transição negativa do sinal de relógio e devem ser escritos na transição positiva;
- $CPHA=1$  – Os dados são lidos à transição positiva do sinal de relógio e devem ser escritos na transição negativa.

## 2.7 Sensor ADNS 5020

O crescente avanço da tecnologia digital, associado ao desenvolvimento de novos algoritmos, tem permitido um número de aplicações cada vez maior [...]. Na automação

industrial, montagem e inspeção de produtos, visão robótica e controle de qualidade podem ser realizados a partir de técnicas de processamento e análise de imagens. Atividades comuns incluem a verificação de falhas em circuitos impressos, separação de peças por robôs em uma linha de montagem e classificação de defeitos em soldas. Possuem inúmeras aplicações na área militar, como a navegação de veículos autônomos (PEDRINI, 2008).

Responsável pelo sensoriamento da malha de realimentação do sistema a ser controlado, este sensor tem a propriedade de identificar a posição em um sistema cartesiano de duas posições, identificando a real posição do modelo.

O ADNS 5020 é encontrado em *mouses* ópticos e é baseado nas tecnologias ópticas de navegação, as quais medem as mudanças nas posições pela aquisição sequencial de imagens (*frames*) da superfície de deslocamento e matematicamente determina a magnitude e a direção do movimento. Conforme exibido na figura 12, o sensor é montado sobre uma placa e um sistema de lentes direcionam o feixe de luz emitido por uma fonte até ele.

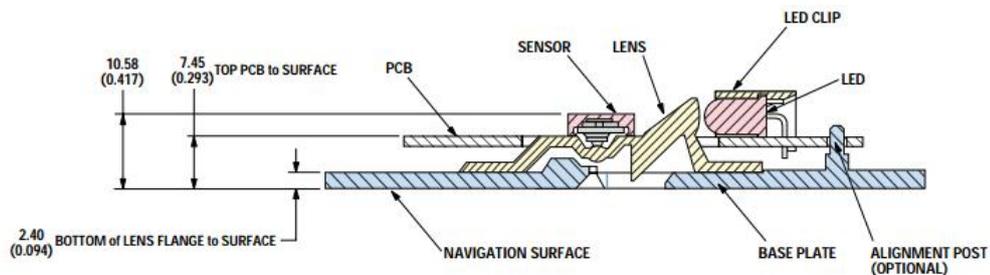


Figura 12 – Montagem ADNS 5020

Fonte: Avago (2007)

Analisando o modo de operação, executa a leitura e interpretação dos dados em dois sistemas distintos: o IAS e o DSP. O Primeiro (*Image Aquisition System*) é responsável pela captura de imagens microscópicas da superfície, através das lentes e da fonte de luz (LED). Posteriormente, o DPS (*Digital Signal Processor*) determina a direção e a distância do movimento, calculando  $\Delta x$  e  $\Delta y$  relativos do movimento. De acordo com Stephan (2013), os DSP's são processadores equipados com instruções especiais implementadas no *hardware*. As operações são também otimizadas de forma que a maioria das instruções é executada num único ciclo. Devido à sua capacidade avançada de processamento, os controladores implementados em DSP's são capazes de executar complexos algoritmos em tempo real, possibilitando, por exemplo, controles de posição ou velocidade sem sensoriamento mecânico.

A comunicação do ADNS 5020 é gerenciada pelo protocolo SPI (*Serial Peripheral Interface*), utilizado para conectar e comunicar dispositivos de forma serial a unidades mais avançadas de processamento e interpretação de dados, como os microcontroladores.

### 2.7.1 Resolução

A resolução deste tipo de sensor é definida como a capacidade de relacionar quantidade de pontos na tela com a distância física percorrida pelo sensor na superfície de deslocamento. Essa resolução pode ser mensurada por dois indicadores: CPI e DPI.

O CPI (*Counts Per Inches*), Contagens Por Polegadas, é definido pela contagem interna de uma variável por unidade de polegada no deslocamento físico em um segundo. Ou seja, quanto mais polegadas forem encontradas no intervalo de um segundo, mais preciso será o sensor.

DPI (*Dots Per Inches*), Pontos Por Polegada é a quantidade de pontos que o sensor identifica no espaço de uma polegada. Analogamente, quanto mais pontos forem identificados em uma polegada, mais preciso será o sensor.

A relação entre as duas grandezas é dada pela seguinte equação:

$$CPI = 2DPI \quad (2.12)$$

Para relacionar diretamente a leitura do sensor com a distância percorrida, deve ser utilizada a resolução em DPI, pois a mesma ignora os efeitos do tempo em sua medição.

## 2.8 Matlab®

Matlab® é uma linguagem de alto nível e um ambiente interativo de computação numérica, visualização e programação. Com os recursos do Matlab®, podem ser feitos análise de dados, desenvolvimento de algoritmos e criação de modelos e aplicações. A linguagem, as ferramentas e funções criadas no próprio *software* permitem a exploração de múltiplos recursos e solução de problemas em tempo menor se comparado com o uso de linguagens de programação tradicionais (MATHWORKS, 2014). É um software proprietário, mas permite a obtenção de uma licença estudantil fornecida pelo site da fabricante. Exibe-se na figura 13 sua tela principal.

Dentro de sua plataforma, o Simulink® é um ambiente de diagrama de blocos para simulação baseada em modelos. Neste programa podem ser feitas simulações, geração automática de códigos e verificação e testes contínuos de sistemas embarcados. Provê um editor gráfico, bibliotecas de blocos e soluções para modelagem e simulação de sistemas dinâmicos. Por ser integrado ao Matlab®, permite incorporar algoritmos nos modelos e exportar dados de resultados de simulações para outras análises. Suas bibliotecas permitem as simulações de diversos sistemas, conforme exibido na figura 14.

Por ser um ambiente muito utilizado tanto no meio acadêmico quanto no industrial, a sua utilização se faz necessária não apenas pelos recursos, mas também pela compatibilidade com outros projetos.

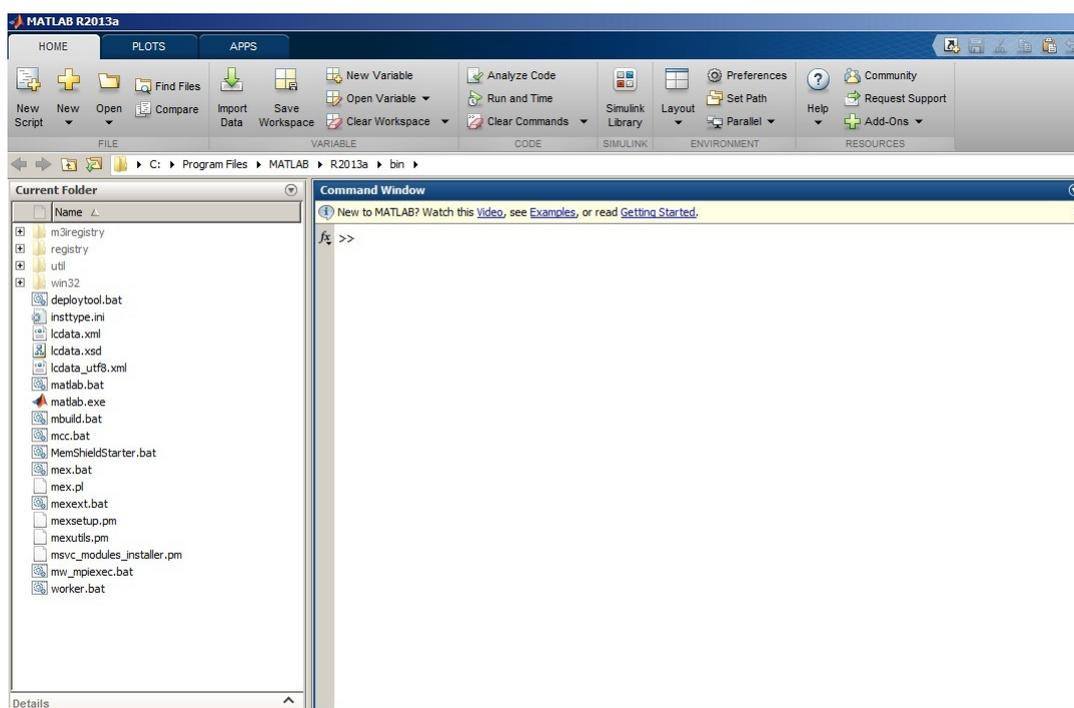


Figura 13 – IDE Matlab®

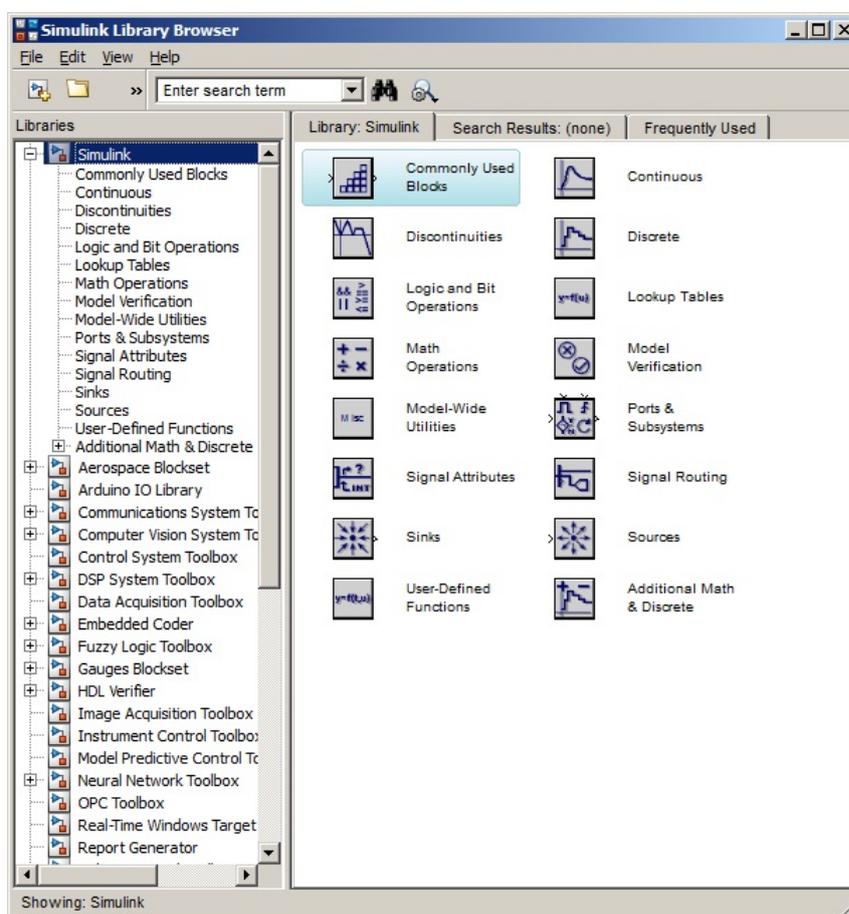


Figura 14 – Biblioteca Simulink®

## 3 Trabalhos relacionados

No campo da robótica móvel, é possível encontrar alguns trabalhos que especificamente abordem a modelagem e o controle de robôs uniciclos, seja por meio de dissertações ou publicação de artigos.

De forma geral, a análise inicia-se da modelagem das características dinâmicas e cinemáticas e posteriormente uma proposta de controle é definida.

### 3.1 Controle não linear de robô móvel uniciclo utilizando linearização entrada-saída por realimentação

Neste artigo, Ferreira, Cerqueira e Costa (2012) propõem uma estratégia de controle baseada na linearização do modelo matemático encontrado. Este tipo de controle é eficiente, contudo embarcá-lo pode muitas vezes demandar uso de *hardwares* mais avançados, ou até mesmo o auxílio de um computador com mais recursos. Como ele mesmo define, um modelo matemático dinâmico não-linear com múltiplas entradas e múltiplas saídas (ou MIMO, do inglês *Multiple Inputs, Multiple Outputs*) para o robô é apresentado, juntamente com a proposta de um controlador não-linear de linearização entrada-saída por realimentação, exibido por meio da figura 15. Todo o artigo é baseado no trabalho de Yun, X. e Yamamoto, Y, *Internal Dynamics of a Wheeled Mobile Robot*, 1993.

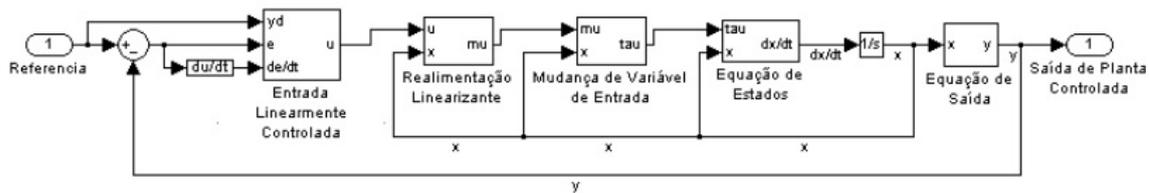


Figura 15 – Diagrama de blocos do controlador

Fonte: Ferreira, Cerqueira e Costa (2012)

Após a execução da modelagem, é apresentada uma simulação, comparando os dados do modelo e a referência a ser adotada. Os parâmetros utilizados não indicam se foram realmente coletados de um robô real ou apenas estimados para simulação.

Para uma trajetória circular, exhibe-se na figura 16 a respectiva resposta:

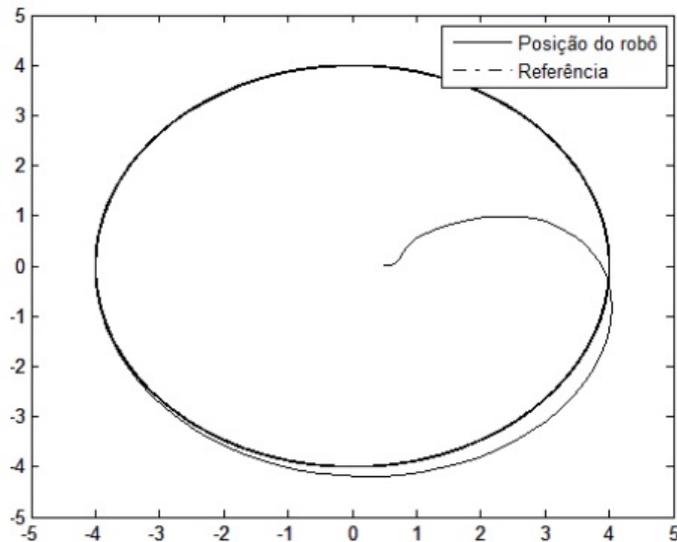


Figura 16 – Trajetória seguida pelo robô a partir da origem para a circunferência

Fonte: Ferreira, Cerqueira e Costa (2012)

Conforme exibido na figura 17, o erro mostrou-se bem aceitável. Contudo, o tempo de resposta pode não ser vantajoso para aplicações que demandem alta velocidade de resposta.

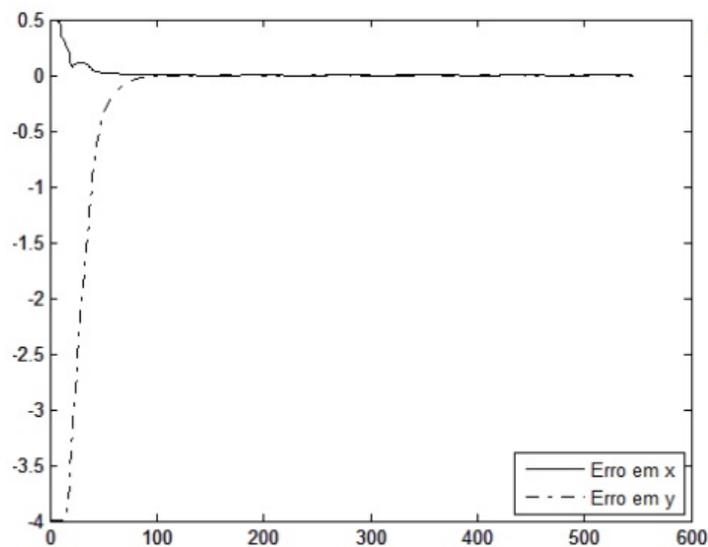


Figura 17 – Erro nas coordenadas x e y do robô para trajetória circular

Fonte: Ferreira, Cerqueira e Costa (2012)

Para uma trajetória lemniscata, as resposta não foram muito boas com relação ao erro, conforme exhibe-se nas figuras 18 e 19.

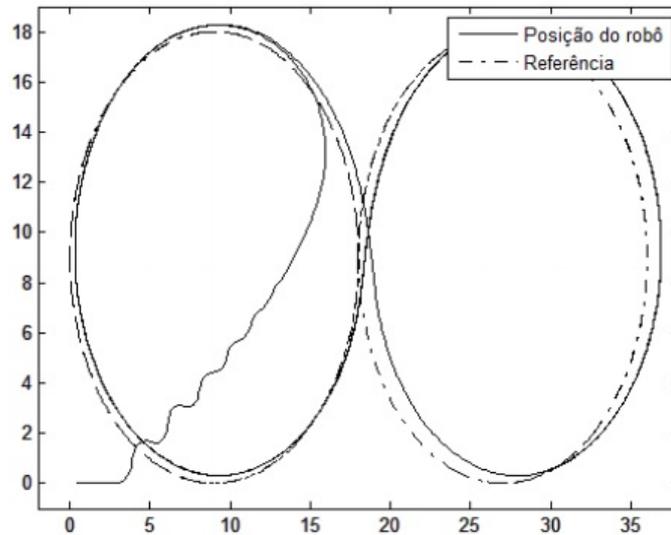


Figura 18 – Trajetória seguida pelo robô a partir da origem para a lemniscata

Fonte: Ferreira, Cerqueira e Costa (2012)

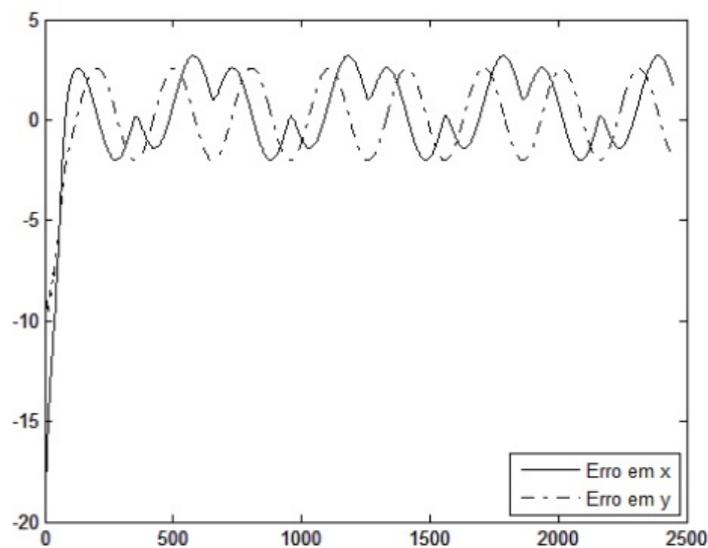


Figura 19 –

Fonte: Ferreira, Cerqueira e Costa (2012)

Para trajetórias mais complexas, o sistema sustenta a estabilidade porém com erro não nulo. De acordo com Ferreira, Cerqueira e Costa (2012), há necessidade de cautela no planejamento de trajetórias e seu compromisso com a tolerância ao erro requerida para a tarefa em execução.

### 3.2 Modelagem e compensação da dinâmica de robôs móveis e sua aplicação em controle de formação

Em sua tese de doutorado, Martins (2009) não só implementa um controlador para a dinâmica e a cinemática, mas também sugere uma compensação das características dinâmicas por um controle adaptativo, exibido por meio da figura 20.

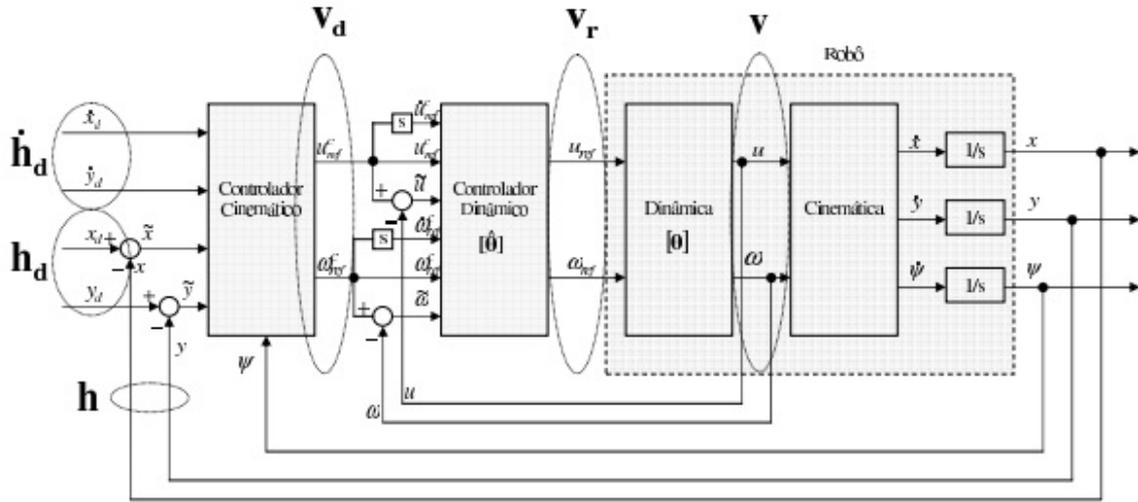


Figura 20 – Diagrama geral do sistema de controle proposto

Fonte: Martins (2009)

Basicamente, o sistema de controle é dividido em controle dinâmico e controle cinemático. As respostas às características dinâmicas (massa, inércia, etc) são usadas para alimentar as propriedades cinemáticas da planta (velocidade, deslocamento, posição, etc). Propõe a seguinte lei para o controle cinemático na equação 3.1.

$$\begin{bmatrix} u_{ref}^c \\ w_{red}^c \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\frac{1}{a}\sin\psi & \frac{1}{a}\cos\psi \end{bmatrix} \begin{bmatrix} \dot{x}_d + l_x \tanh\left(\frac{k_x}{l_x} \tilde{x}\right) \\ \dot{y}_d + l_y \tanh\left(\frac{k_y}{l_y} \tilde{y}\right) \end{bmatrix} \quad (3.1)$$

Utiliza parâmetros reais e simula o modelo proposto. As respostas são satisfatórias, e o erro com o compensador adaptativo é levado a níveis mínimos.

O comparativo do desempenho para uma trajetória lemniscata é exibido por meio da figura 21.

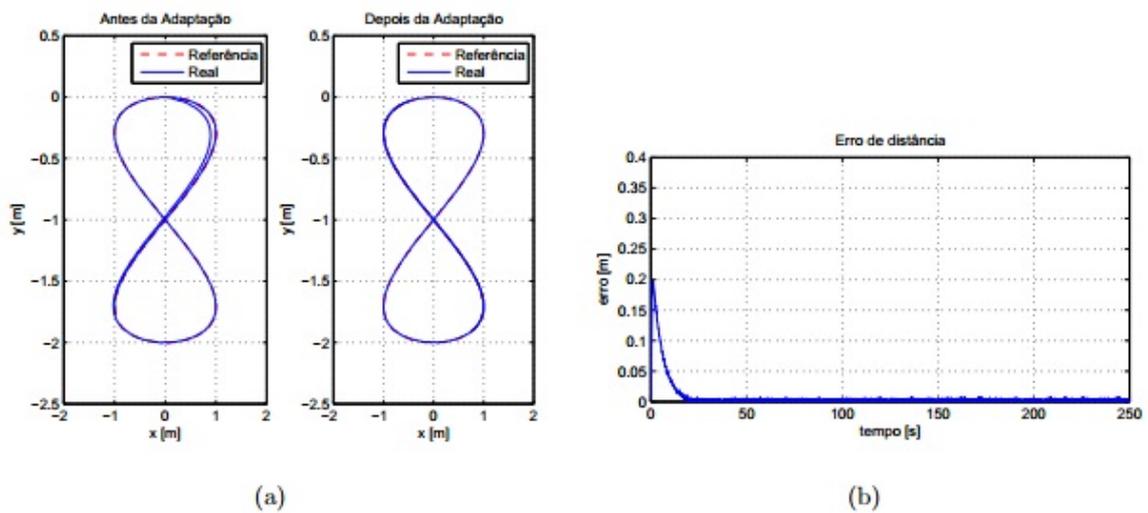


Figura 21 – Controlador dinâmico: trajetória lemniscata. (a) Trajetória percorrida pelo robô antes e depois da adaptação de parâmetros. (b) erro da distância com os parâmetros exatos

Fonte: Martins (2009)

As variáveis utilizadas para controlar o sistema são as velocidades angular e linear. Segundo o autor, a inserção do controlador dinâmico produz redução no valor do erro de velocidade, e por conseguinte, diminuição no valor do erro de seguimento.

### 3.3 Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida

Para uma aplicação didática, Melo (2007) propõe um sistema totalmente simulado, aplicando as equações de modelagem e controle PID embarcado. Utiliza em sua tese os conceitos de prototipagem rápida, bastante difundida, para obtenção de trajetórias pela manipulação de pontos no plano cartesiano.

De forma análoga aos trabalhos acima relacionados, faz a modelagem de um robô virtual, propõe um controle embarcado, mas PID, e realiza as simulações. Como diferencial, a modelagem mantém o foco no controle dos motores, e por consequência, o posicionamento é atingido. Aplica conceitos de CCI (Centro de Curvatura Instantânea) em sua modelagem, e sua representação exibe-se na figura 22.

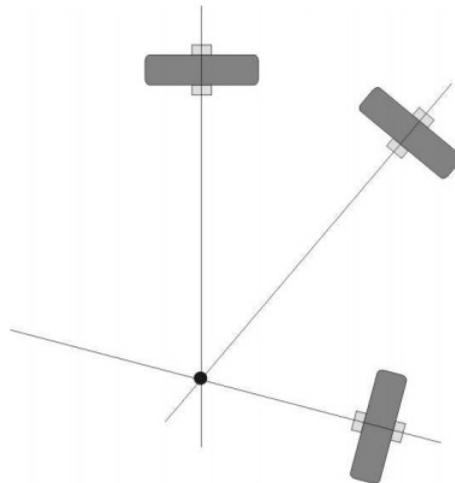


Figura 22 – Centro de Curvatura Instantânea no ponto

Fonte: Melo (2007)

Conforme exibido na figura 23, o controlador é elaborado de forma a controlar as características dinâmicas e cinemáticas, com as velocidades das rodas direita e esquerda como variáveis de controle.

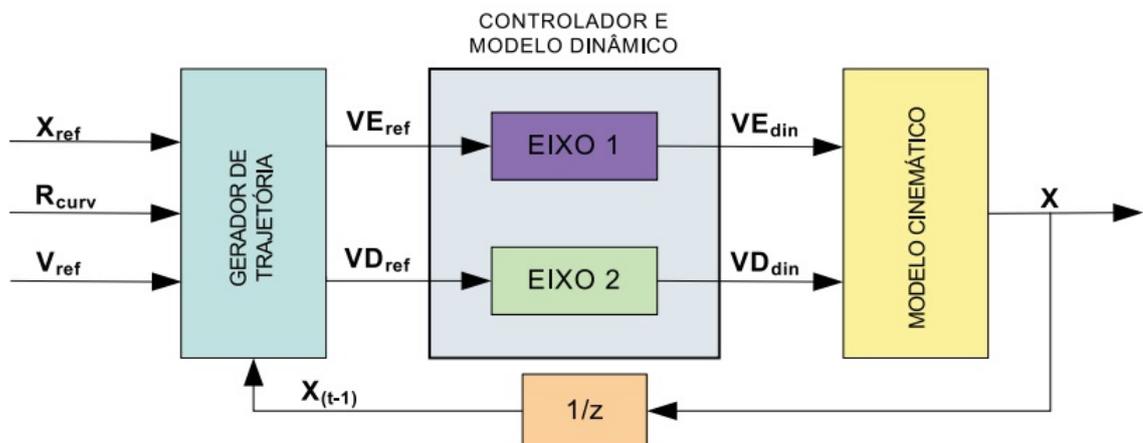


Figura 23 – Estrutura de controle do robô móvel

Fonte: Melo (2007)

### 3.4 Controle de Trajetória de Robôs Móveis Omni-direcionais: Uma Abordagem Multivariável

Nascimento (2009) aborda em sua pesquisa a aplicação do controle multivariável no controle de robôs, e cita em seu referencial a aplicação de AGV's no contexto atual. A

modelagem realizada tem um escopo diferente dos trabalhos pesquisados, pois utiliza o conceito de espaço de estados, e sua representação matricial é dada pela equação 3.2

$$\begin{bmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_{n-1} \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} x_2 \\ \vdots \\ x_n \\ f(x) + g(x)u \end{bmatrix} \quad (3.2)$$

A teoria de controle multivariável é bem explorada, e alia ao seu desenvolvimento os conceitos de estabilidade pelo método de lugar das raízes. O sistema de controle proposto é chamado de controle em cascata, mais especificamente o controle de trajetória em cascata para robôs móveis, baseado em uma malha de controle externa, aplicada à cinemática da base móvel, gerando as velocidades angulares de referência para os motores em função do erro de postura, e uma malha interna relacionada à dinâmica da base móvel. Este esquema de controle é exibido por meio da figura 24.

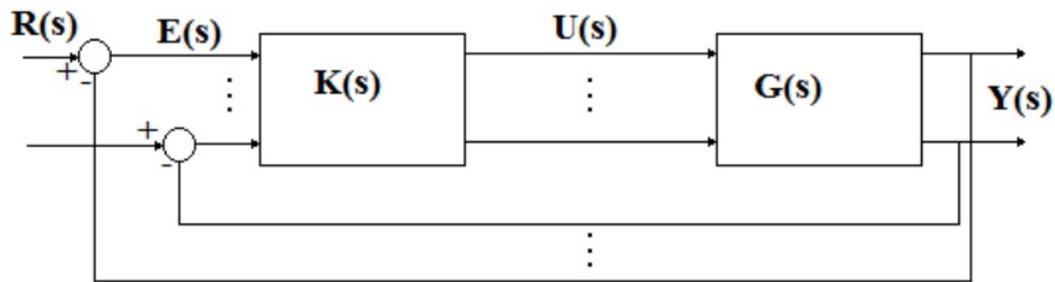


Figura 24 – Esquema de controle

Fonte: Nascimento (2009)

O sistema possui três controladores PID, e cada sistema é SISO (*Single Input / Single Output*). O modelo é acoplado, mas a lei de controle para o mesmo não é. Esta realimentação entrada-estado permite que o sistema seja controlado por qualquer controlador linear, como no caso um PID.

## 4 Materiais e Métodos

Neste capítulo, será abordada a metodologia utilizada para a conexão e utilização dos elementos envolvidos, gerando assim as condições de funcionamento do sistema. Após garantir a comunicação do *hardware*, seguem as etapas para o cálculo do posicionamento e por fim, o acoplamento físico dos elementos utilizados. Na figura 25 tem-se um diagrama de blocos com estas etapas.

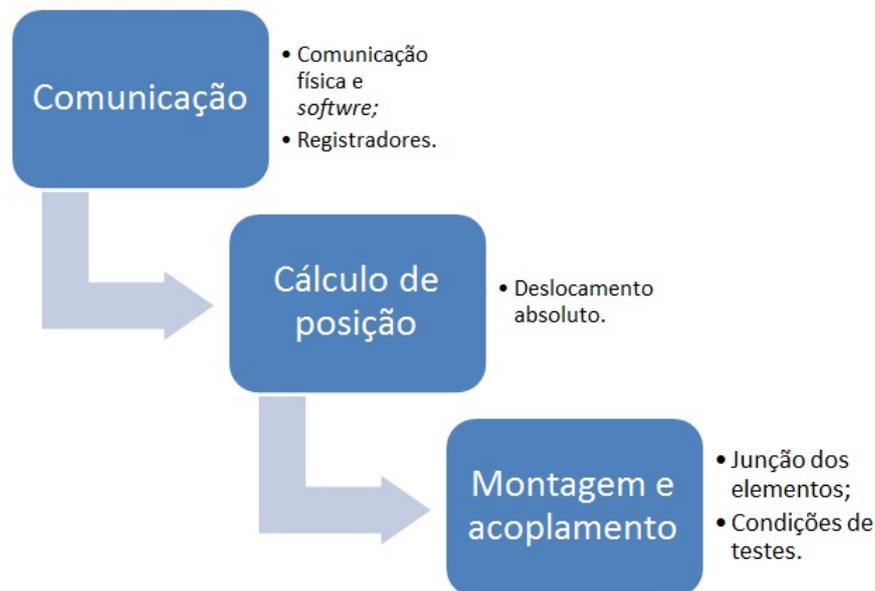


Figura 25 – Etapas

### 4.1 Comunicação

Tão importante quanto o conhecimento dos dispositivos utilizados é o conhecimento dos seus métodos de comunicação, bem como a correta conexão dos elementos. O microcontrolador aparece como elemento central da comunicação, gerenciando os dados e mantendo as conexões ativas ou não. Na figura 26 exibe-se o diagrama de comunicação do projeto:

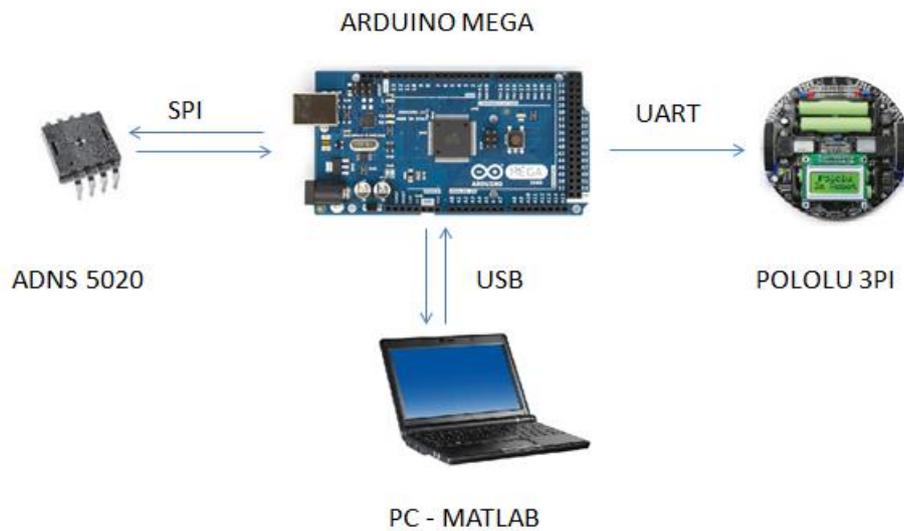


Figura 26 – Gerenciamento da comunicação

Um dos pontos mais importantes para a escolha do Arduino MEGA é a capacidade de executar múltiplas comunicações. Isso se dá pela existência de quatro canais para comunicação serial, além de pinos que permitem a comunicação SPI, utilizada pelo ADNS 5020. Essa arquitetura de rede permite que este *hardware* se comunique tanto com seus periféricos quanto com um computador, de forma sincronizada e simultânea.

#### 4.1.1 ADNS 5020 - Arduino

A comunicação SPI é utilizada para conectar o ADNS 5020 ao microcontrolador. Para que a operação e a comunicação sejam bem sucedidas, as conexões dos pinos devem ser realizadas de forma correta. Na figura 27 segue a pinagem deste sensor e suas respectivas funções estão descritas na tabela 3.

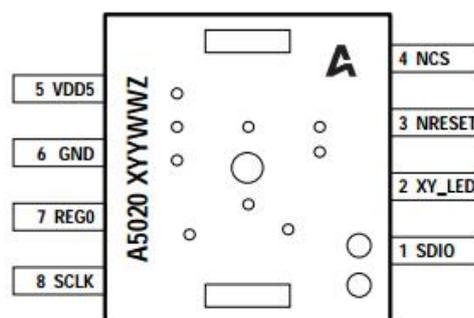


Figura 27 – Pinagem ADNS 5020

Fonte: Avago (2007)

Fonte: Avago (2007)

Pino	Nome	Descrição
1	SDIO	Entrada e saída da comunicação serial
2	XY_LED	Controle do LED
3	NRESET	Pino de RESET (Ativo em nível baixo)
4	NCS	Seleciona o Chip (Ativo em nível baixo)
5	VDD5	Tensão de Alimentação
6	GND	Terra
7	REGO	Regulador de Saída
8	SLCK	Entrada de <i>Clock</i> da Comunicação serial

Tabela 3 – Pinagem ADNS 5020 e descrições

Para esta aplicação, os pinos 2 e 7 não serão utilizados, bastando o controle em apenas seis pinos:

- SDIO – Este pino é responsável pelo tráfego das informações na comunicação serial. Ele deverá ser conectado aos pinos do microcontrolador referentes à saída e à estrada de dados.
- NRESET – O ADNS 5020 não aplica um auto *reset* quando é alimentado. Logo, a cada vez que o dispositivo é iniciado, este pino deve ser levado para nível lógico baixo para “*resetar*” o chip.
- NCS – Para cada operação de leitura ou escrita, este pino deve ser levado para nível lógico baixo. Caso contrário, os pulsos de *clock* e os dados de comunicação serão ignorados. Esta função existe pelo fato de permitir que esse dispositivo faça parte de uma rede com outros periféricos, sendo ativado apenas quando solicitado.
- VDD5 – Alimentação de 5V.
- SLCK – Entrada de sinal de *Clock* para sincronizar a transmissão de dados. É gerado pelo mestre da comunicação, neste caso o microcontrolador.

Na tabela 4 tem-se a interligação correta dos pinos do ADNS 5020 com o Arduino MEGA:

Arduino	ADNS 5020
MISO (50)	SDIO
MOSI (51)	SDIO
SCK (52)	SCLK
+Vcc	VDD5
Gnd	GND
8	NCS
9	NRESET

Tabela 4 – Conexão ADNS 5020 e Arduino

Conforme exibido por meio da figura 28, o microcontrolador disponibiliza quatro modos de operação de acordo com o especificado na seção 2.6 que aborda a comunicação SPI.

Mode	Clock Polarity (CPOL)	Clock Phase (CPHA)
SPI_MODE0	0	0
SPI_MODE1	0	1
SPI_MODE2	1	0
SPI_MODE3	1	1

Figura 28 – Modos de operação SPI

Fonte: <http://arduino.cc/en/Reference/SPI>

O modo de operação do ADNS 5020 pode ser identificado por meio da figura 29, que relaciona os dados transmitidos com o pulso e fase do sinal de *clock*.

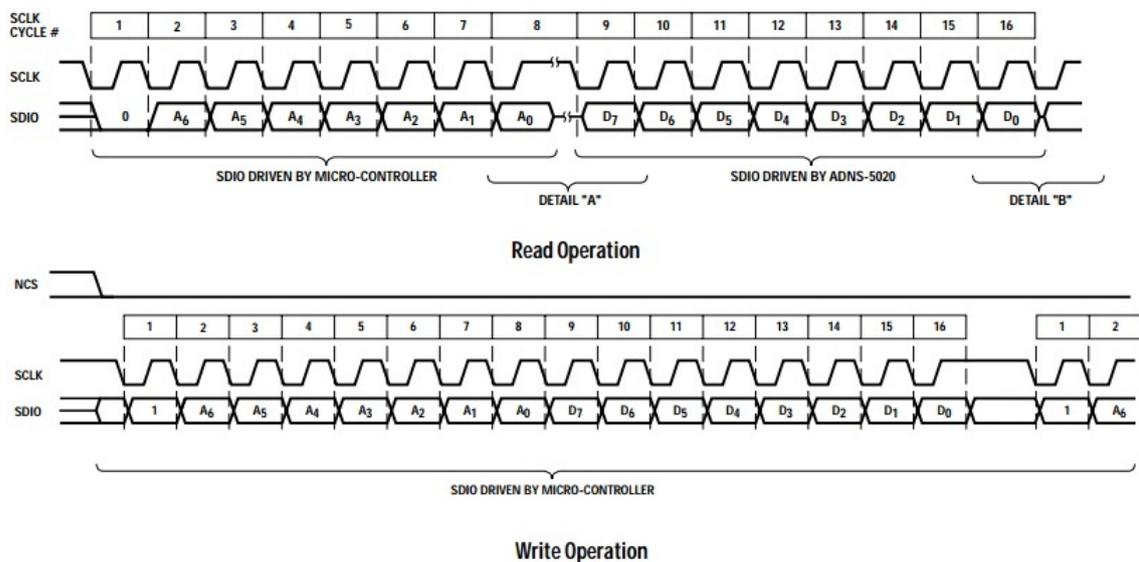


Figura 29 – Operações de leitura e escrita

Fonte: Avago (2007)

O modo de operação é identificado como 3 (CPOL=1 e CPHA =1).

#### 4.1.1.1 Registradores

Os dados gerados pelo processamento do sensor e as informações usadas para controle da operação do chip são acessados pelos registradores. Estes registradores, conforme exibido na figura 30, estão dispostos em posições fixas da memória interna do ADNS 5020, e são identificados pelos seus endereços.

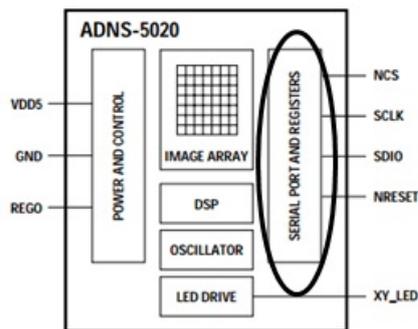


Figura 30 – Diagrama de Blocos

Fonte:Avago (2007)

O acesso a esses endereços são feitos por comandos específicos de leitura e escrita do microcontrolador, respeitando o protocolo de comunicação pela porta serial (SPI), e podem ser visualizados na tabela 5.

Fonte: Avago (2007)

Endereço	Registrador	Leitura/Escrita	Valor padrão
0x00	<i>Product_ID</i>	L	0x12
0x02	<i>Motion</i>	L	0x00
0x03	<i>Delta_X</i>	L	-
0x04	<i>Delta_Y</i>	L	-
0x0a	<i>Minimum_Pixel</i>	L	-
0x0b	<i>Pixel_Grab</i>	L/E	-
0x0d	<i>Mouse Control</i>	L/E	-
0x3a	<i>Chip_Reset</i>	E	N/A
0x3b – 0x3e	<i>Reserved</i>		
0x3f	<i>Inv_Rev_ID</i>	L	0xff
0x40 – 0x62	<i>Reserved</i>		
0x63	<i>Motion_Burst</i>	L	0x00

Tabela 5 – Registradores

Os códigos de acesso aos dados enviados pelo sensor (leitura), assim como o comando de escrita dos registradores realizados pelo microcontrolador Arduino MEGA e programado em sua IDE são exibidos por meio da figura 31.

```
// Função de Escrita
void ADNS_write(unsigned int address, unsigned int data){
  // NCS em nível baixo para selecionar o Chip:
  digitalWrite(nCS,LOW);
  // Envia o endereço dos registradores e o valor:
  SPI.transfer(address);
  SPI.transfer(data);
  // NCS em nível alto para liberar o canal:
  digitalWrite(nCS,HIGH);
}
// Função de Leitura
unsigned int ADNS_read(unsigned int address){
  unsigned int data;
  //NCS em nível baixo para selecionar o Chip:
  digitalWrite(nCS,LOW);
  //Envia o endereço via SPI
  SPI.transfer(address);

  pinMode(MOSI, INPUT);
  data = SPI.transfer(0x00);
  pinMode(MOSI, OUTPUT);
  //NCS em nível alto para liberar o canal:
  digitalWrite(nCS,HIGH);
  return(data);
}
```

Figura 31 – Funções de leitura e escrita

Alguns registradores merecem maior atenção, pois serão utilizados para o cálculo do movimento absoluto. Cada um desses registradores são formados por oito bits (um byte). Dependendo do registrador, cada bit apresenta uma informação específica.

*Motion*: este registrador detecta se houve algum movimento do sensor. Mantém o valor em “0” caso esteja parado. Em caso de movimento, o bit de número sete (MOT) assume o valor “1”, e seu valor final se torna 128. Após a detecção do movimento, as leituras dos registradores Delta\_X e Delta\_Y devem ser feitas logo em seguida.

**Motion**                      Address: 0x02  
Access: Read/Write            Reset Value: 0x00

Bit	7	6	5	4	3	2	1	0
Field	MOT	Reserved						

Figura 32 – Registrador *Motion*

Fonte:Avago (2007)

Delta\_X e Delta\_Y: Os movimentos relativos nas posições X e Y são guardados nestes registradores. Apenas é registrado o movimento entre um ciclo de leitura e outro, e o valor representa o movimento em uma escala de bits de um contador interno. É preciso trabalhar esse valor incluindo nos cálculos a resolução adotada e a relação com o deslocamento real.

<b>Delta_X</b>		Address: 0x03							
Access: Read		Reset Value: 0x00							
Bit	7	6	5	4	3	2	1	0	
Field	X <sub>7</sub>	X <sub>6</sub>	X <sub>5</sub>	X <sub>4</sub>	X <sub>3</sub>	X <sub>2</sub>	X <sub>1</sub>	X <sub>0</sub>	

<b>Delta_Y</b>		Address: 0x04							
Access: Read		Reset Value: 0x00							
Bit	7	6	5	4	3	2	1	0	
Field	Y <sub>7</sub>	Y <sub>6</sub>	Y <sub>5</sub>	Y <sub>4</sub>	Y <sub>3</sub>	Y <sub>2</sub>	Y <sub>1</sub>	Y <sub>0</sub>	

Figura 33 – Registradores Delta\_X e Delta\_Y

Fonte: Avago (2007)

Para realizar uma leitura com maior precisão, é recomendado que os registradores Delta\_X e Delta\_Y sejam lidos em sequência, logo após a detecção do movimento.

*Mouse\_control*: Os parâmetros de resolução do chip são acessados por este registrador. Como opções, este trabalha em dois modos: 500 CPI ou 1000 CPI, bastando apenas alterar o valor do bit RES para “0” ou “1”, dependendo da escolha.

<b>Mouse_control</b>		Address: 0x0d							
Access: Read/Write		Reset Value: 0x00							
Bit	7	6	5	4	3	2	1	0	
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	RES	

Data Type: Eight bit number

USAGE: Resolution and chip reset information can be accessed or to be edited by this register.

Field Name	Description
RES	Set resolution 0 = 500 cpi 1 = 1000 cpi
Reserved	Reserved

Figura 34 – Registrador *Mouse\_control* e parâmetros de resolução.

Fonte: Avago (2007)

### 4.1.2 Arduino - Pololu

Como já descrito, o robô Pololu 3pi apresenta um microprocessador ATmega328. Sua arquitetura permite a comunicação UART (*Universal Asynchronous Receiver/Transmitter*), utilizando dois pinos, denominados TX e RX, respectivamente o pino transmissor e o pino receptor. Por ser uma comunicação de fácil implementação, não exige acesso a posições de memória ou endereços de registradores, e a sincronização é feita por *software*.

2/3rds of battery voltage	analog input 6 (through jumper)	ADC6
ICSP programming lines (x3)	digital pins 11, 12, and 13	PB3, PB4, PB5
reset pushbutton	reset	PC6
UART (RX and TX)	digital pins 0 and 1	PD0 and PD1
I2C/TWI	inaccessable to user	
SPI	inaccessable to user	

Figura 35 – Pinagem Pololu

Fonte: Pololu (2014)

De acordo com a figura 35, os pinos utilizados para a comunicação serão PD0 e PD1, como RX e TX. Fica evidenciado, também, porque o processamento do sensor ADNS 5020 não é executado pelo próprio robô, visto que sua comunicação SPI não é acessível pelo usuário.

A comunicação é feita pela simples conexão do TX do Arduino com o RX do pololu, e vice-versa, conforme a tabela 6.

Pololu	Arduino
TX (PD1)	RX1(19)
RX (PD0)	TX1(18)

Tabela 6 – Conexão Pololu e Arduino Mega

O Arduino utiliza esta comunicação para enviar os parâmetros de velocidade para o robô, não havendo necessidade de retorno de informação.

### 4.1.3 Arduino - PC

Para realizar a comunicação com o Matlab®, utiliza-se o protocolo USB (*Universal Serial Bus*). De forma simplificada, este protocolo permite a comunicação entre computadores e periféricos.

As coordenadas do posicionamento são enviadas pelo Arduino para o Matlab®, que, entre outras aplicações, plota um gráfico referente ao seu deslocamento e compara os dados

reais com os simulados no modelo. Para iniciar tal comunicação, deve ser informadas a velocidade de transferência dos dados (configurada no Arduino) e a porta a ser utilizada, determinada pelo computador durante a instalação do *hardware*, que é feita na primeira conexão.

O diagrama de blocos exibido na figura 36 determina os passos a serem realizados na comunicação.



Figura 36 – Comunicação com o Matlab®

Os testes devem ser realizados com uma comunicação sem fio, pois a tração de cabos que interligam o sistema ao PC influencia diretamente o comportamento dinâmico do modelo. Para eliminar a dependência de cabos, utiliza-se a comunicação pelo protocolo zigbee, proporcionada pelo módulo XBee®, exibido por meio da figura 37.

## 4.2 Cálculo do posicionamento

O fabricante do ADNS 5020 fornece algumas informações de desempenho do dispositivo em diferentes superfícies. De acordo com a cor, a precisão aumenta e mais pontos



Figura 37 – Módulo XBee®

Fonte: <http://www.labdegaragem.org/>

podem ser lidos durante o deslocamento. Este parâmetro de desempenho é exibido na figura 38, e deve ser levado em consideração durante o cálculo da resolução.

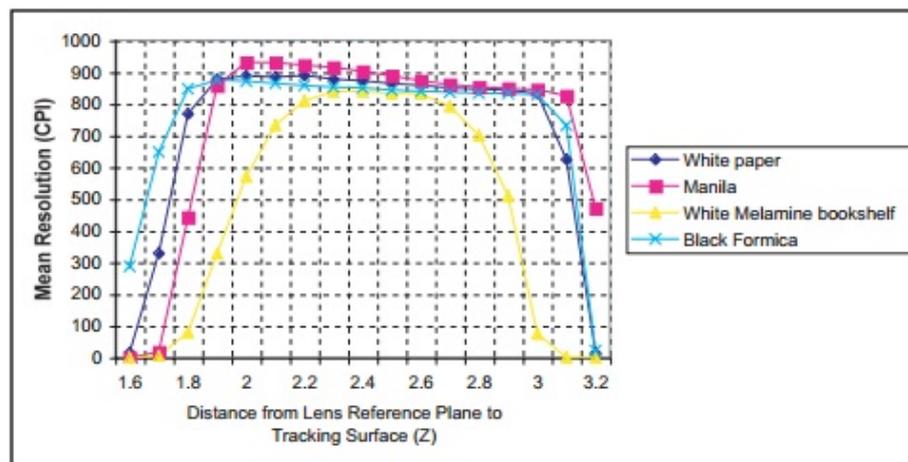


Figura 38 – Resolução aproximada *versus* distância das lentes ao plano de referência

Fonte: Avago (2007)

O tipo de cor que confere uma maior precisão é a parda (*manila*), pois mantém uma resolução o mais próximo da máxima para a distância padrão de 2,4 mm. Esta resolução é de no máximo 900 CPI

Utilizando a resolução máxima do ADNS 5020 em 900 CPI, utilizando a equação 2.9, tem-se que:

$$DPI = 450$$

Considerando a seguinte relação:

$$1 \text{ polegada} = 25,4 \text{ milímetros},$$

e serão lidos 500 pontos em uma polegada ou 25,4 mm, cada ponto lido terá a seguinte razão:

$$\Delta x = \left(\frac{25,4}{450}\right) n_x,$$

$$\Delta x = 0,0564n_x \quad (4.1)$$

onde  $n_x$  é o número de pontos lidos no deslocamento em  $x$ , com  $\Delta x$  em mm (milímetros).

De forma análoga, vale a mesma relação para  $\Delta y$ :

$$\Delta y = 0,0564n_y \quad (4.2)$$

onde  $n_y$  é o número de pontos lidos no deslocamento em  $y$ , com  $\Delta y$  em mm (milímetros).

A orientação e o sentido do movimento em relação aos eixos coordenados que referencia o sensor é exibido por meio da figura 39.

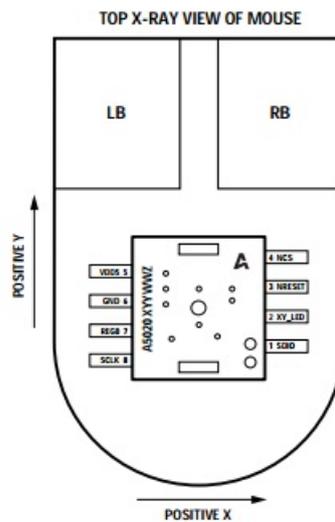


Figura 39 – Sentido do movimento

Fonte: Avago (2007)

Esses deslocamentos são acumulados e somados a fim de indicar a posição real do sensor. A quantidade de pontos lidos é indicada na figura a seguir, que é o resultado da leitura dos registradores Delta\_X e Delta\_Y, assim como o sentido do movimento, conforme exibido por meio da figura 40.

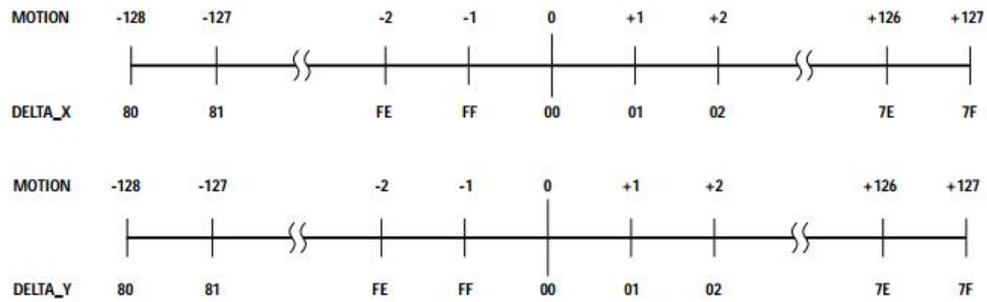


Figura 40 – Leitura dos registradores de movimento

Fonte: Avago (2007)

Analisando a sequência e o sinal de leitura, os registradores guardam valores de 0 a 255 no sistema binário. Para deslocamentos positivos, assume valores entre 0 e 127. Já para os deslocamentos negativos, guarda valores entre 128 e 255. Logo, caso o deslocamento seja negativo, o valor deverá ser subtraído de 256 e atribuído o sinal quando houver a conversão para a unidade métrica. Esse valor lido é o  $n_x$  ou  $n_y$  das equações 4.1 e 4.2.

Exibe-se na figura 41 o trecho do programa que parametriza a resolução, faz a leitura do deslocamento e calcula a posição real.

```
ADNS_write(0x0d,1); //Parametriza Resolução em 1000 CPI

if(ADNS_read(0x02)==128){
    //Ocorreu movimento
    delay(25);

    x=ADNS_read(0x03); // Leitura do deslocamento em x
    y=ADNS_read(0x04); // Leitura do deslocamento em y
    if(x>127)x=-256+x; //atribuição do sinal negativo em x
    if(y>127)y=-256+y; //atribuição do sinal negativo em y

    pos_x=(x*0.0564)+pos_x; //Cálculo da posição em x
    pos_y=(y*0.0564)+pos_y; //Cálculo da posição em y
```

Figura 41 – Leitura e cálculo das posições

### 4.3 Montagem e acoplamento

Para que o sensor ADNS 5020 funcione corretamente, sua distância em relação à superfície de deslocamento deve ser mantida em aproximadamente 2,4 mm, conforme exibe-se na figura 42.

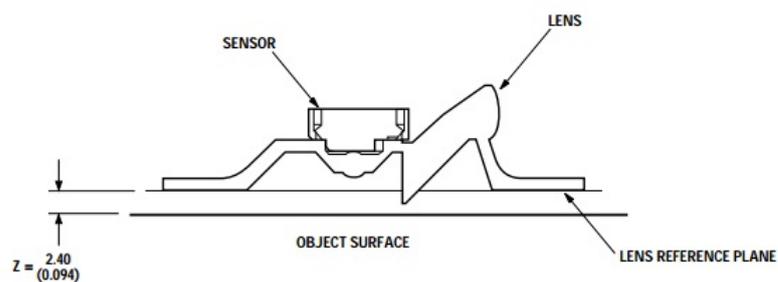


Figura 42 – Distância do plano das lentes ao plano de deslocamento

Fonte: Avago (2007)

Como a estrutura de um *mouse* já mantém esta condição de fábrica, optou-se em manter a base deste dispositivo e acoplá-la ao robô utilizado. A união foi feita por uma chapa de PVC, ajustável ao ponto de garantir que a base utilizada fique faceada com a superfície de locomoção. Exibe-se na figura o acoplamento 43 realizado.

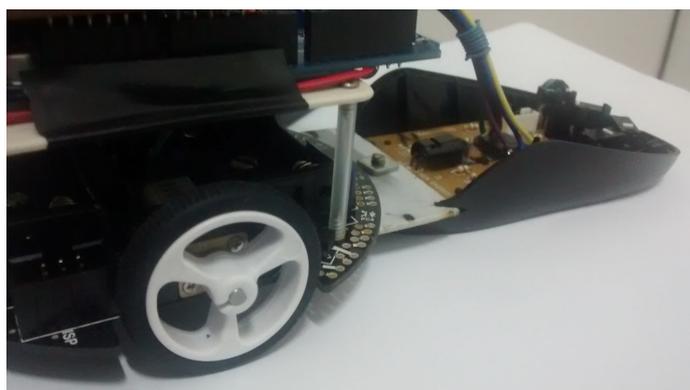


Figura 43 – Acoplamento

O modelo utilizado para os testes exibe-se na figura 44.

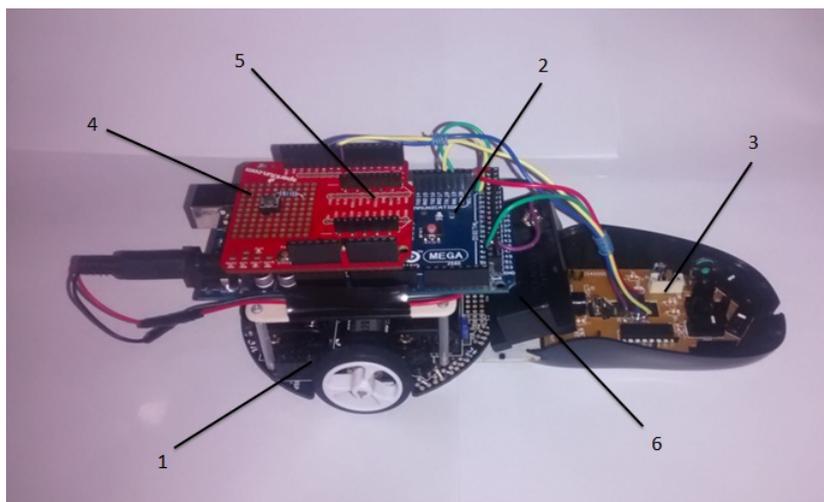


Figura 44 – Modelo de teste montado

- 1: Robô Pololu 3pi
- 2: Microcontrolador Arduino Mega
- 3: Sensor ADNS 5020
- 4: Botão para zerar posicionamento
- 5: *Shield* para módulo xBee
- 6: *Case* para bateria 9 V

## 5 Simulações e Análise dos Resultados

### 5.1 Aquisição de Dados

Como método para monitoramento do desempenho do sistema, faz-se a aquisição dos dados direto para o Matlab®. Dessa forma, esses dados podem ser comparados com as referências de deslocamento.

Após configurados os parâmetros de comunicação, o trecho de programa exibido por meio da figura 45 faz a leitura do deslocamento real e o plota em um gráfico.

```

i=1;
clear x
clear y;

while s.BytesAvailable>0

a=fscanf(s);
a=str2double(a);
y(i)=a;
a=fscanf(s);
a=str2double(a);
x(i)=a;
i=i+1;
end

figure(1)
clf
plot(x,y)
title('Gráfico da trajetória');
xlabel('x(mm)');
ylabel('y(mm)');

```

Figura 45 – Gerador do gráfico

Cada trajetória é lida até o final. Um *buffer* armazena os dados, e após a leitura de todas as posições o gráfico é gerado. Exibe-se na figura 46 o resultado de um movimento realizado manualmente.

### 5.2 Descrição do modelo

A equação 2.2 do modelo cinemático, serve de base para a simulação do sistema de controle. Nela, nota-se que há duas entradas e duas saídas, configurando um sistema multivariável com interações.

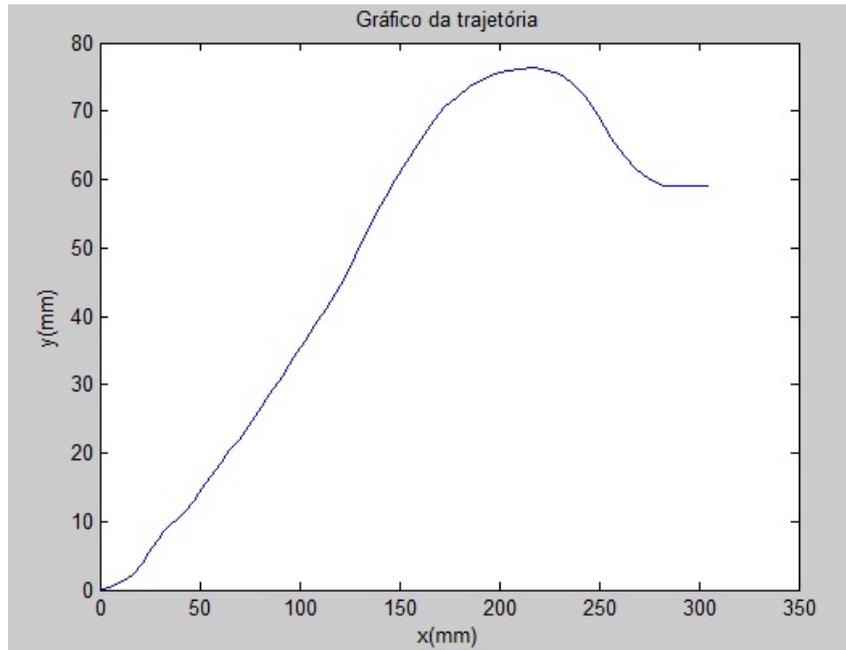


Figura 46 – Gráfico da trajetória: movimento manual

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -a \sin \psi \\ \sin \psi & a \cos \psi \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ \omega \end{bmatrix} + \begin{bmatrix} \delta_x \\ \delta_y \\ 0 \end{bmatrix}$$

As variáveis controladas são as coordenadas x e y, e as variáveis manipuladas são as velocidades linear e angular, respectivamente  $u$  e  $\omega$ .

A equação 2.2 pode ser representada num modelo de diagrama de blocos para simulação, conforme exibido na figura 47.

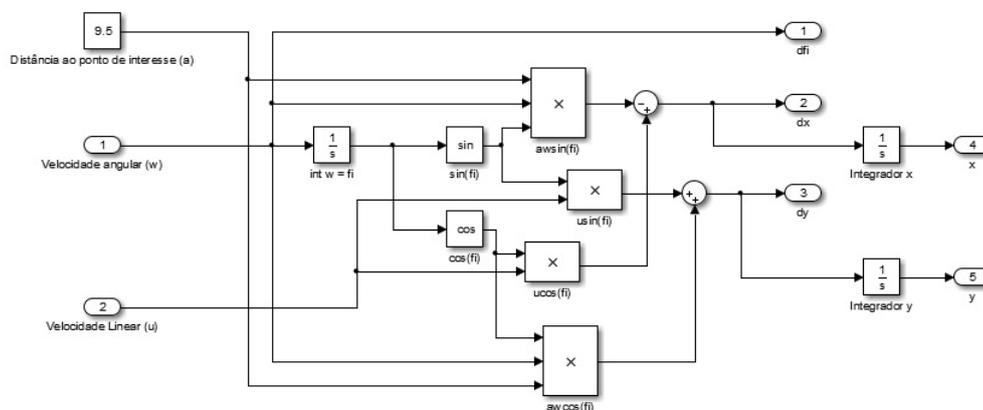


Figura 47 – Diagrama de blocos do sistema

O modelo apresenta quatro blocos principais e cada um destes realiza uma função específica na dinâmica do sistema:

- Entradas: as entradas do sistema, exibidas por meio da figura 48, são responsáveis por excitar o mesmo, e o sinal de controle atua diretamente sobre elas. O sistema apresenta duas entradas distintas, sendo estas as velocidades linear ( $u$ ) e angular ( $\omega$ ), definidas respectivamente pelas equações 2.3 e 2.4.

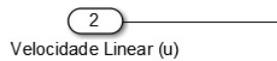


Figura 48 – Entradas

- Distância ao ponto de interesse: esta constante representa a distância em mm do centro do eixo virtual que une as duas rodas do robô até o ponto onde é feita a leitura das posições x e y pelo sensor, e foi coletada por medição realizada após o acoplamento do sensor, resultando em 95 mm, conforme a figura 49.

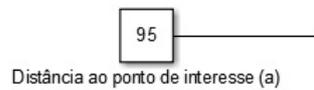


Figura 49 – Distância ao ponto de interesse

- Interações: são as relações matemáticas que definem as interações entre as entradas, resultando em valores de posições nas saídas, conforme exibido na figura 50.

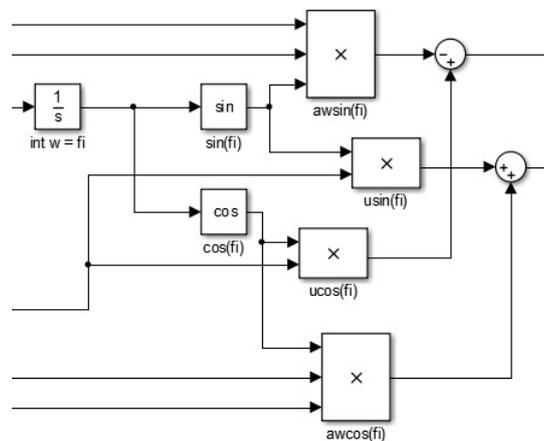


Figura 50 – Interações

- Saídas: são os valores das posições  $x$  e  $y$  influenciadas por toda a dinâmica do sistema e, neste caso, são consideradas em mm. Exibe-se na figura 51 as saídas precedidas dos seus respectivos integradores, pois os valores de referência serão comparados com a soma dos valores infinitesimais resultantes das interações. A orientação do modelo não foi considerada no projeto, sendo assim desnecessário o uso de integrador nesta saída.

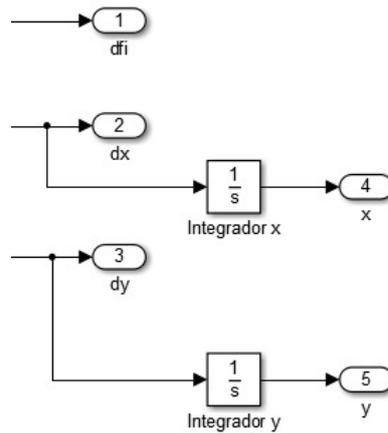


Figura 51 – Saídas

### 5.3 Sistema de controle proposto

Para fechar a malha de realimentação do sistema de controle, deve-se determinar antecipadamente a variável de controle que terá mais influência na variável de saída. Para a implementação, será considerada uma trajetória retilínea, partindo da origem em direção a uma reta paralela ao eixo  $y$ .

Logo, seguindo tal princípio, variações em  $x$  deverão influenciar de forma mais determinante os deslocamentos angulares do veículo, enquanto variações em  $y$ , os lineares. Exibe-se na figura 52 um sistema veicular que, partindo da origem, deve seguir uma trajetória retilínea em  $x=X_p$ , paralela ao eixo  $y$ .

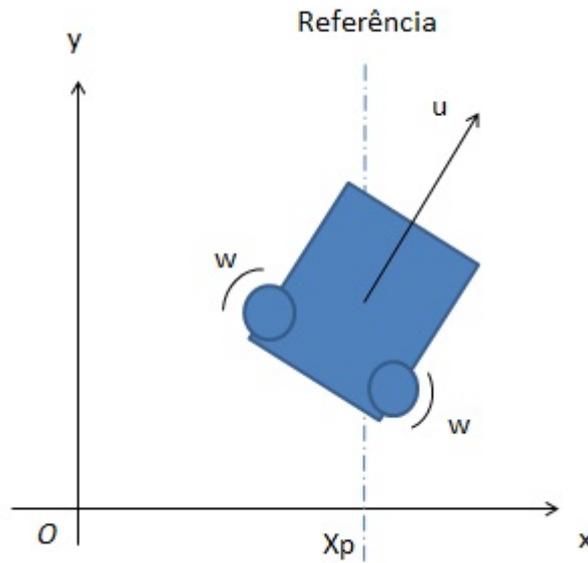


Figura 52 – Deslocamento

O sistema de controle proposto, conforme exibido na figura 53, apresenta dois controladores PID, um em cada malha de controle, destinados aos controles das velocidades lineares e angulares, cujo sinal de erro provém da diferença entre as posições  $x$  e  $y$  de referência e as atuais do modelo.

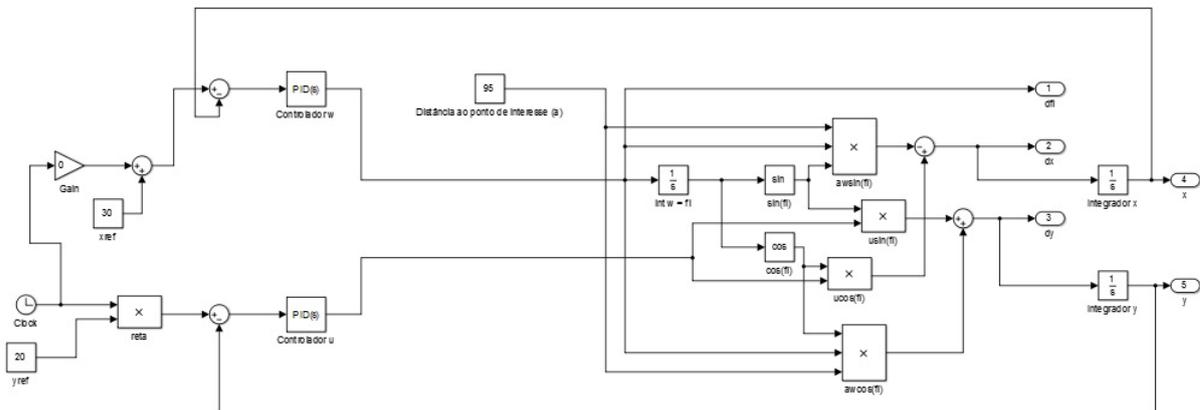


Figura 53 – Sistema controlado

### 5.3.1 Determinação dos parâmetros dos controladores

Para obter os valores do ganho crítico  $K_{cr}$  e do período crítico  $P_{cr}$ , o método utilizado foi o experimental por meio de simulação. O modelo simulado foi excitado por entradas do tipo degrau amplificadas por um ganho  $K$ , variando de 0 a um  $K_{cr}$  que resulta em uma saída instável sustentada pela primeira vez.

Para o controlador que atua sobre a velocidade angular para ajuste da posição  $x$ , as respostas exibidas por meio da figura 54 são produtos da variação do ganho e indicam que as respostas obtidas apresentam valores estáveis para diferentes valores de  $K$ , e assim permitem que este método seja utilizado.

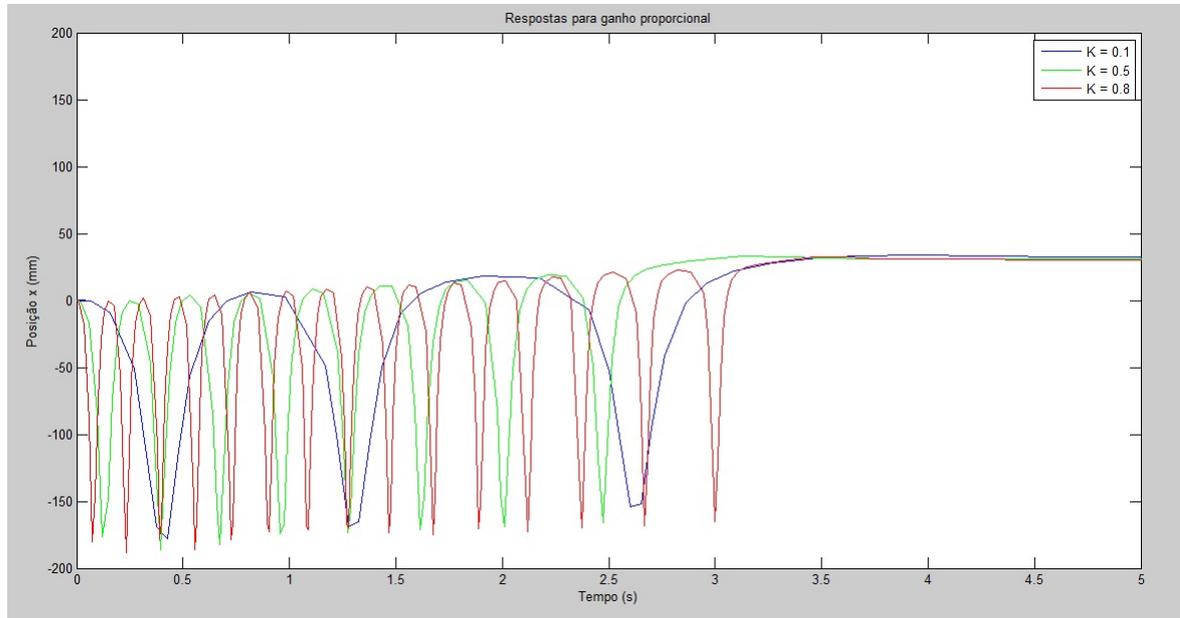


Figura 54 – Respostas para ganho proporcional

O ganho foi aumentado, e quando atingiu o valor  $K = 1,3$ , a oscilação permaneceu sustentada para  $t \rightarrow \infty$ , conforme exibe-se na figura 55.

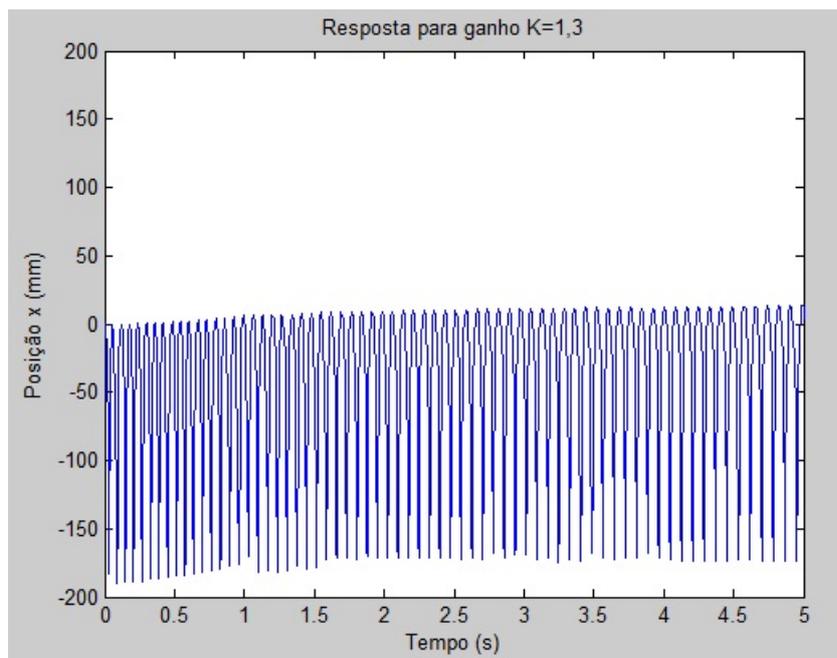


Figura 55 – Resposta para  $K=1,3$

Os dados obtidos para o controlador da velocidade angular são:

$$K_{cr} = 1,3$$

$$P_{cr} = 0,06$$

Os parâmetros então são obtidos pelas fórmulas da tabela 1:

$$K_p = 0,78;$$

$$T_i = 0,03;$$

$$T_d = 0,075;$$

Sendo  $K_p$ ,  $T_i$  e  $T_d$  definidos pela equação 2.5.

Os mesmos testes foram realizados para a malha de controle da velocidade linear, obtendo-se os seguintes parâmetros:

$$K_{cr} = 1,83;$$

$$P_{cr} = 0,019;$$

E por conseguinte:

$$K_p = 1,10;$$

$$T_i = 0,00967;$$

$$T_d = 0,002375.$$

## 5.4 Aplicação do controle

Exibe-se nas figuras 56 e 57 os dados obtidos após a aplicação das ações de controle, indicando que o sistema se estabiliza e que os valores de referência são atingidos. Os valores dos parâmetros métricos estão em milímetros, e a velocidade angular em rad/s.

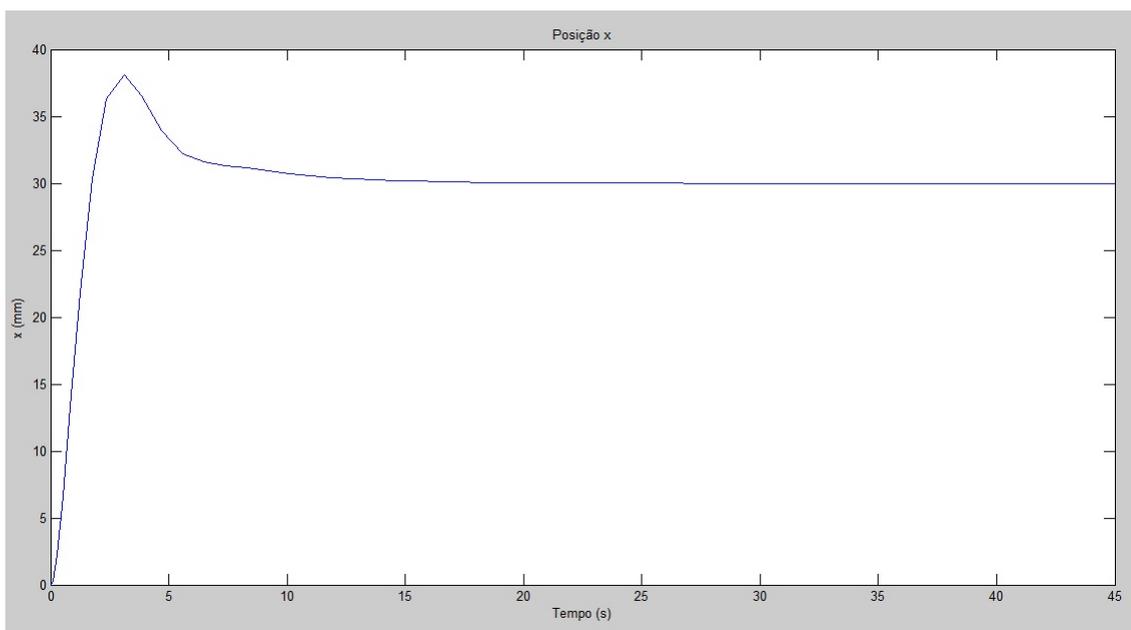
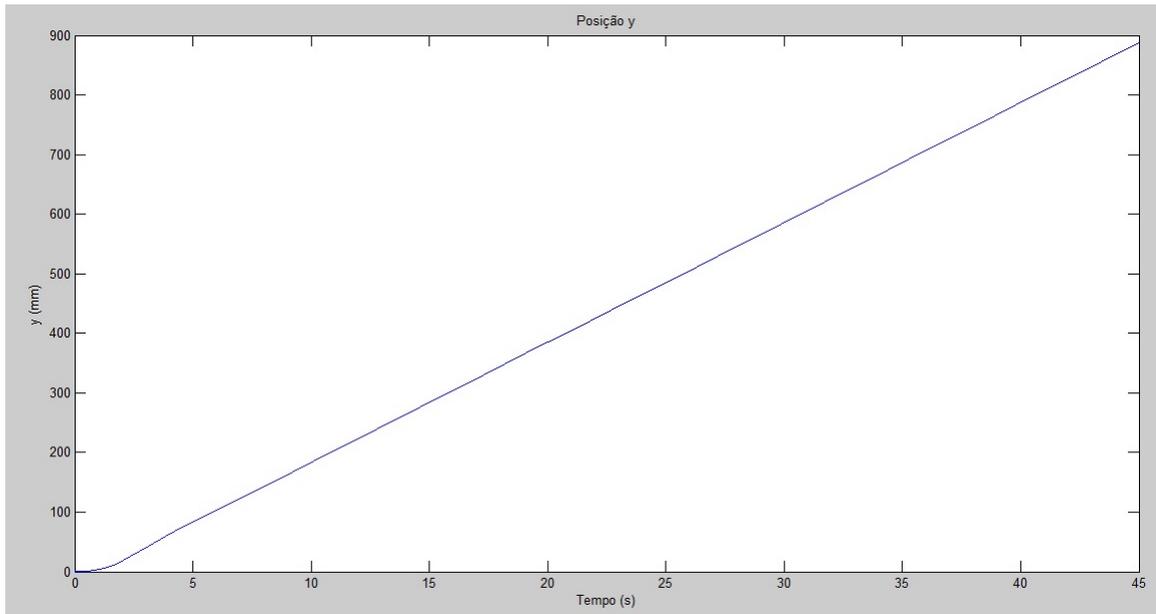


Figura 56 – Posição x *versus* tempo

Figura 57 – Posição  $y$  versus tempo

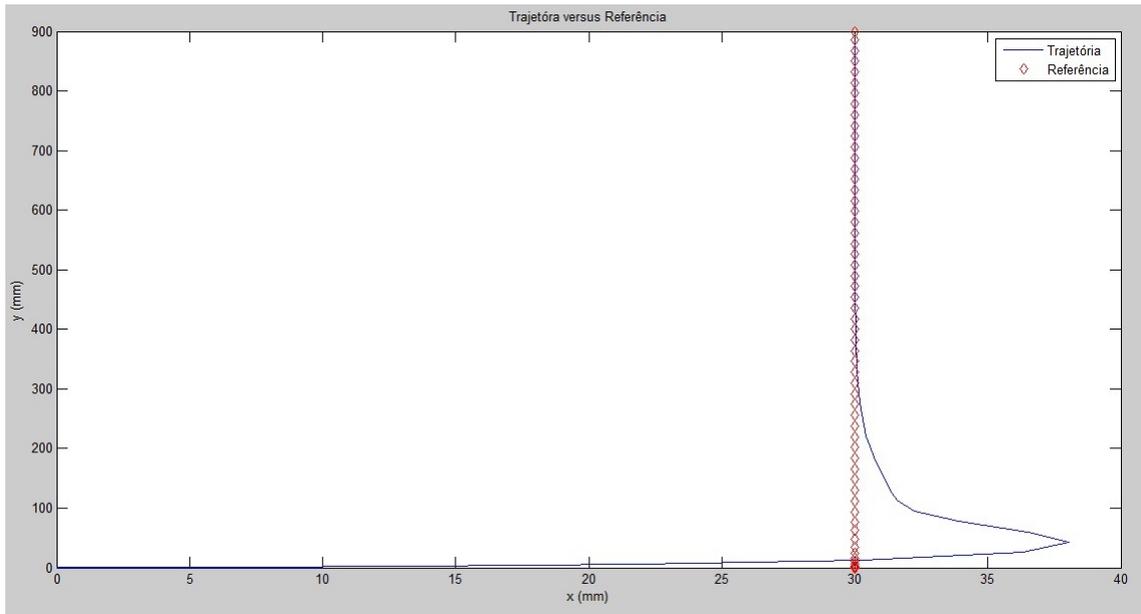
A leitura das especificações do regime transitório estão em resumo na tabela 7.

Especificação	Valor
Tempo de atraso ( $t_d$ )	1,12 s
Tempo de subida ( $t_r$ )	2,33 s
Tempo de pico ( $t_p$ )	4,52 s
Tempo de acomodação 5% ( $t_s$ )	8,44 s
Máximo sobressinal ( $M_p$ )	36,01 mm

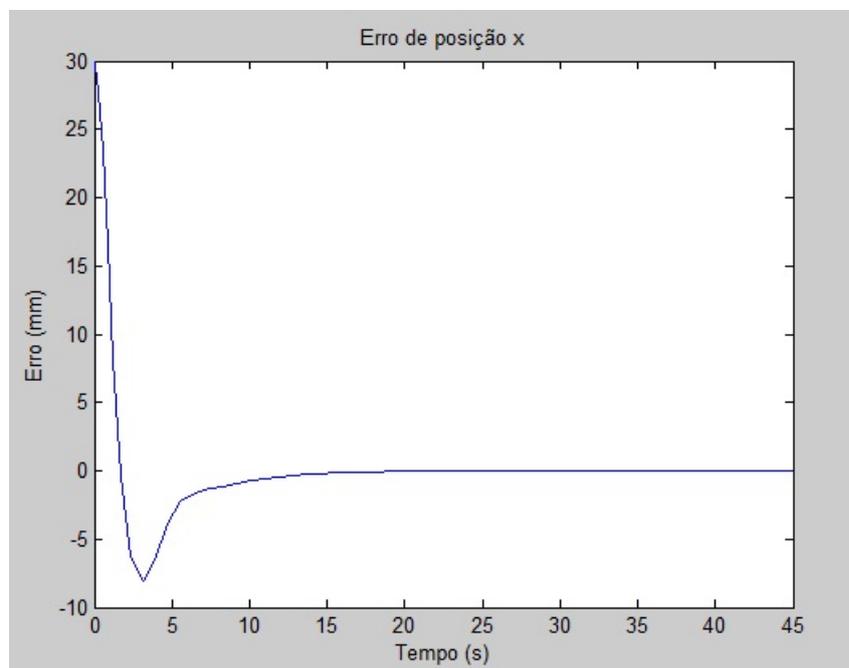
Tabela 7 – Especificações do regime transitório

Utilizando os dados da tabela 7, por meio da equação 2.6, tem-se o máximo sobressinal percentual em 20,04 %.

No teste, o movimento do robô inicia na origem e desloca-se em direção a uma trajetória retilínea em  $x = 30$  e  $y \rightarrow \infty$ . Os erros de posição são compensados pelos controladores PID que atuam diretamente nas velocidades. Na figura 58 exibe-se a trajetória do robô em direção à reta que serve de referência para o deslocamento:

Figura 58 – Trajetória *versus* referência

Os erros decaem, mostrando a tendência do sistema a se estabilizar. A posição em  $x$  é alcançada em aproximadamente 6 s. O posicionamento em  $y$  segue a tendência de alcançar a reta da trajetória de referência, conforme exibido nas figuras 59 e 60.

Figura 59 – Erros de posição  $x$ 

Visualizar a atividade do controlador no decorrer do tempo serve de base para identificar em qual momento as velocidades são acionadas, ou seja, em quais circunstâncias um movimento linear ou angular se faz necessário para a correção dos desvios de trajetória.

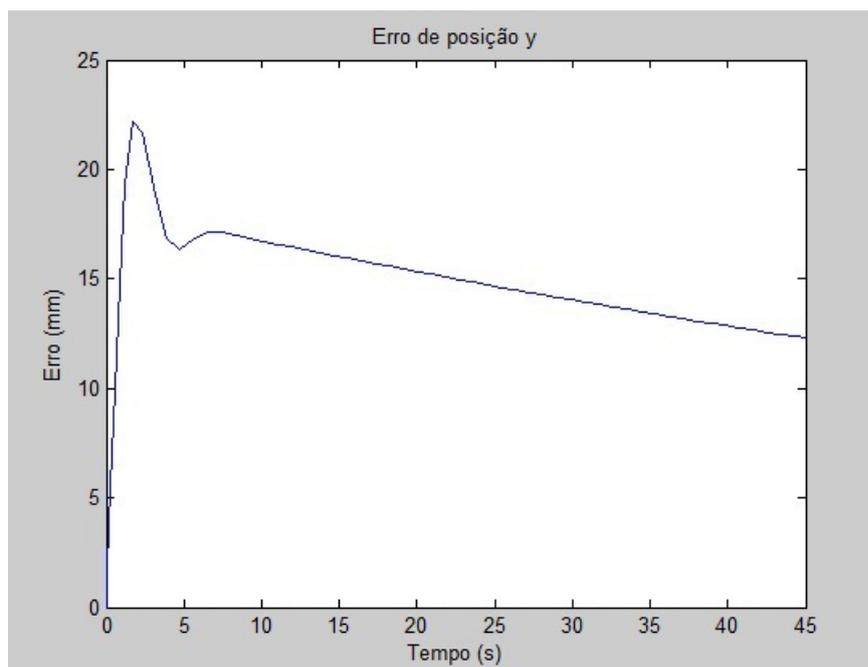
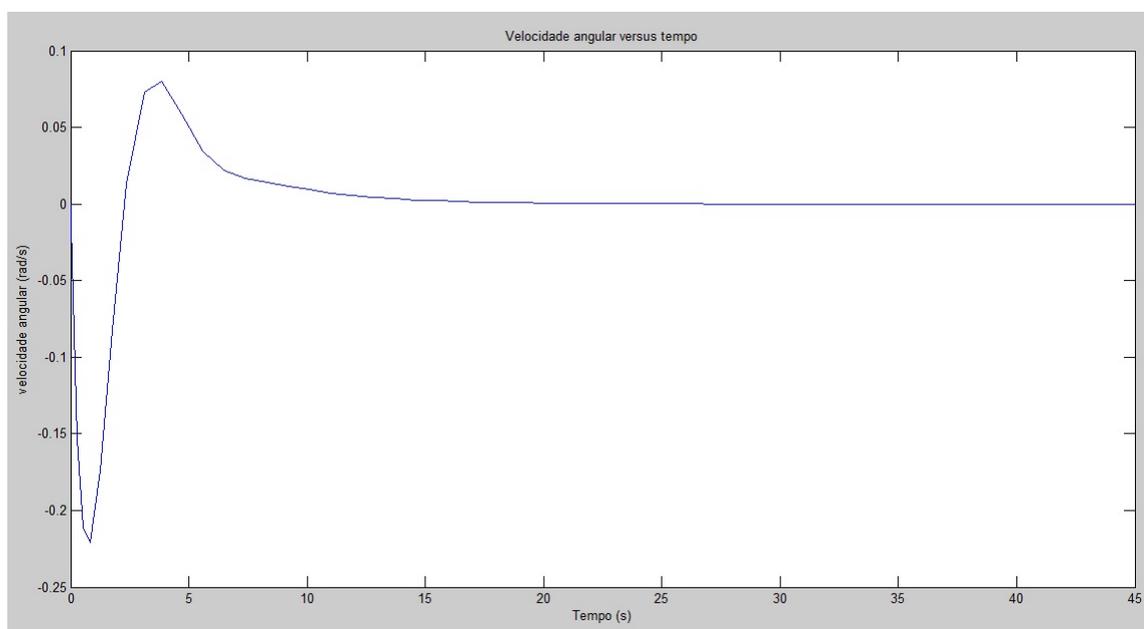
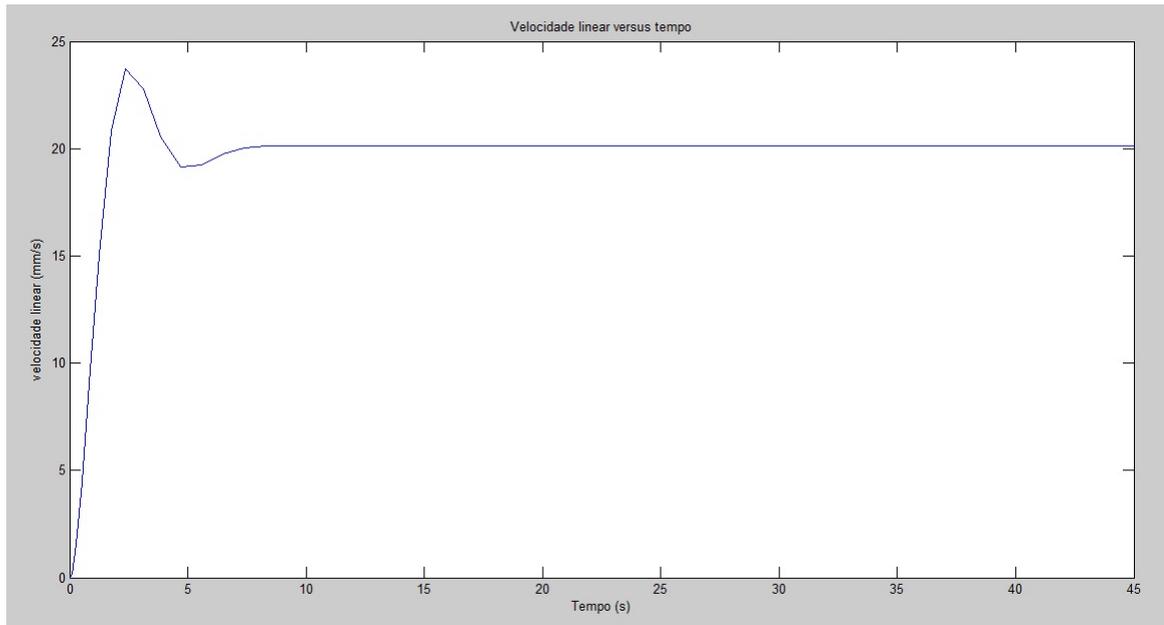


Figura 60 – Erro de posição y

Conforme exibido por meio das figuras 61 e 62, a velocidade angular cai para zero após o robô atingir a posição x desejada, e de forma análoga, a velocidade linear se mantém em 20mm/s quando o veículo atinge a reta, mantendo o deslocamento linear.

Figura 61 – Velocidade angular *versus* tempo

Figura 62 – Velocidade linear *versus* tempo

O desempenho do sistema pode ser avaliado em malha aberta, comprovando a instabilidade do mesmo sem a realimentação. O PID atua sem o sinal de erro em sua entrada, e dessa forma, o sistema não atinge a referência especificada. Exibe-se nas figura 63 e 64 um comparativo das respostas em malha aberta e com controle PID, além da referência.

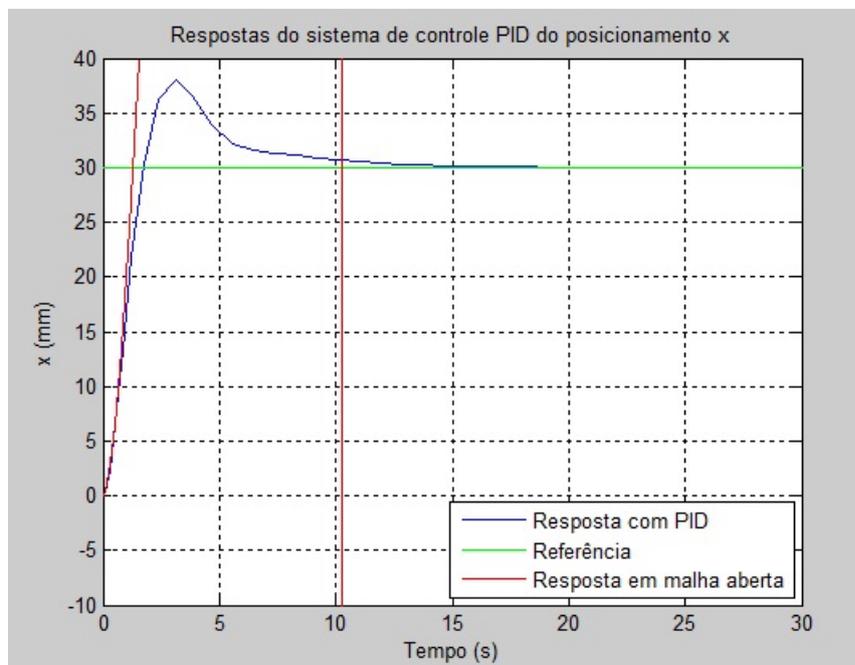


Figura 63 – Respostas do sistema de controle PID do posicionamento x

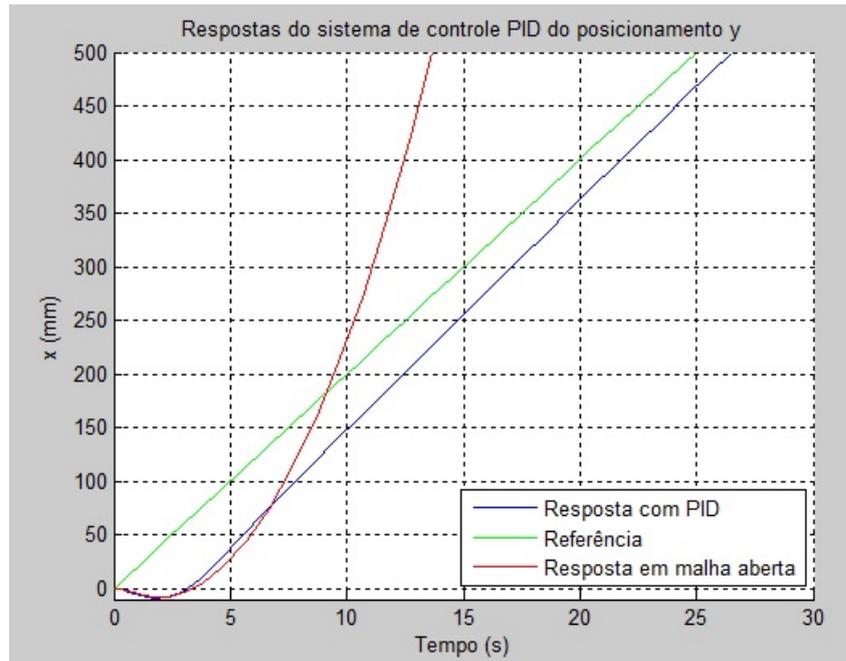


Figura 64 – Respostas do sistema de controle PID do posicionamento y

### 5.4.1 Distúrbios

A estabilidade está comprovada, mas se faz necessária a adição de distúrbios que simulem a saída súbita do robô de sua trajetória, causada por irregularidades na superfície de deslocamento, por exemplo. O pulso a seguir foi gerado para simular um toque no robô após este atingir a estabilidade do posicionamento em x. Este pulso gera um desvio de 6 mm no sentido positivo de x, ou seja, para a direita, com um tempo de duração de 1 s. O pulso é introduzido na malha de realimentação da posição x. De forma análoga, um pulso é gerado para desvios negativos no sentido de x. Os pulsos e suas defasagens são exibidos por meio da figura 65.

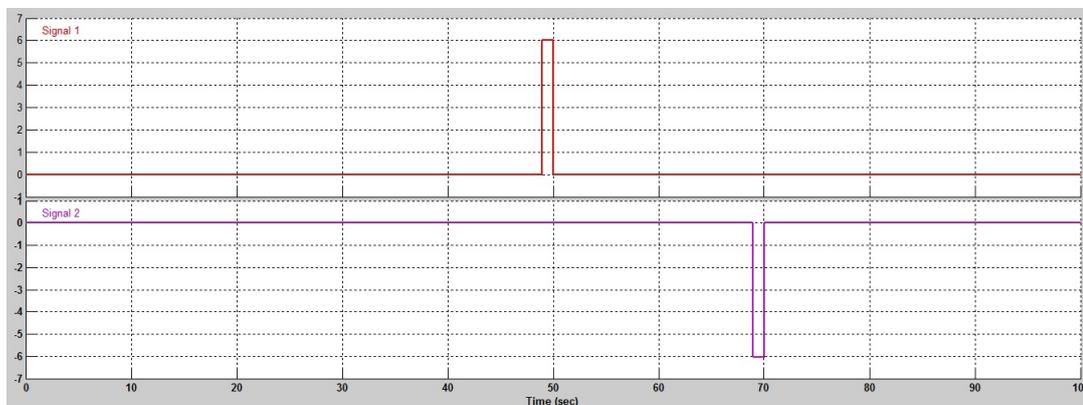


Figura 65 – Pulsos de distúrbio

A velocidade angular definida pela equação 2.4 pode ser notada quando o sinal de controle referente à esta velocidade é analisado.

Conforme exibido na figura 66, no primeiro instante, em  $t=49s$ , a velocidade imediatamente adota um valor positivo, gerando uma rotação no sentido anti-horário. Esta ação de controle fica bem clara quando a equação 2.4 é analisada, pois para que se tenham valores positivos, a velocidade angular da roda direita deve ser maior que a velocidade da roda esquerda. De forma análoga, em  $t=69s$ , a ação inversa é realizada, gerando uma rotação no sentido horário (negativa), com a velocidade da roda esquerda maior que a da roda esquerda.

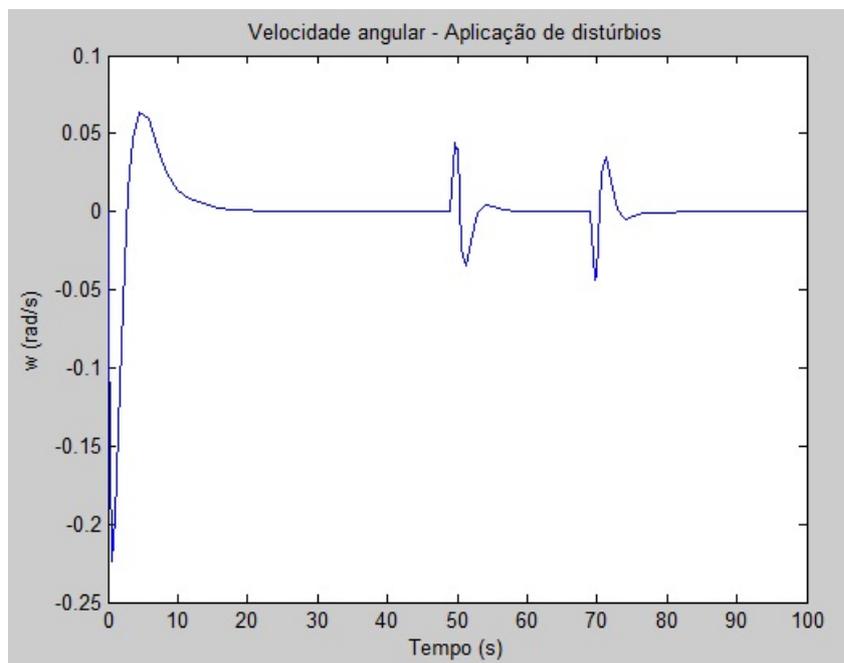


Figura 66 – Velocidade angular com aplicação de distúrbios

Como resultado, apesar de apresentar sobressinal na resposta, o sistema se estabiliza e retorna para a trajetória referenciada, evidenciado na figura 67.

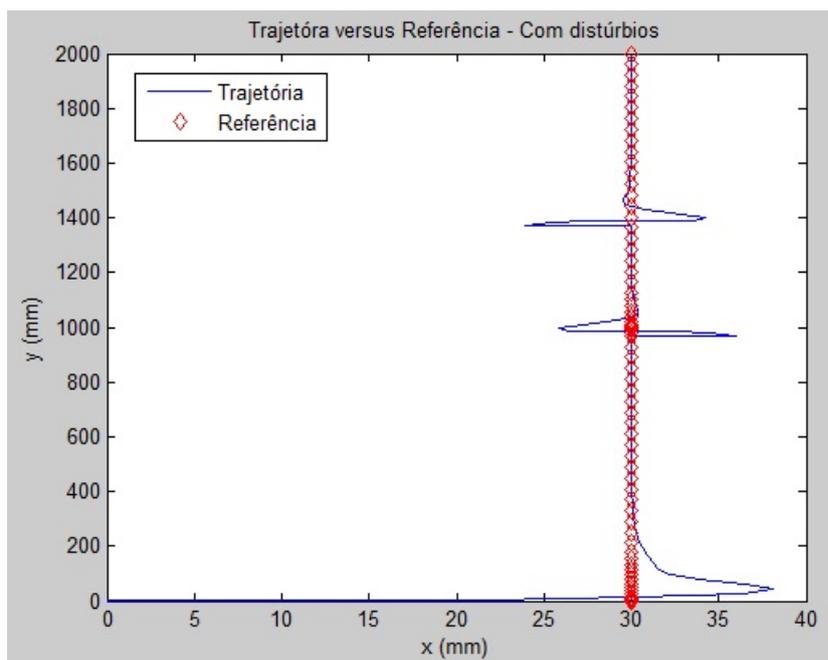


Figura 67 – Trajetória com distúrbios

## 6 Conclusões

Nos trabalhos relacionados que envolvem a modelagem e a simulação do deslocamento de robôs móveis, todo o posicionamento é definido pela velocidade das rodas. Os cálculos são feitos considerando que sensores acoplados aos motores (*encoders*) determinam a real posição da planta. A utilização de um sensor que indique essa posição diretamente proporciona uma leitura mais precisa e conseqüentemente um controle mais efetivo em aplicações reais. Como não depende de um meio fixo para se referenciar, o sensor elimina a dependência de pontos de referência posicionados no perímetro do deslocamento.

A realização do controle com base apenas no modelo cinemático se mostrou bem sucedida, mantendo os parâmetros de trajetória mesmo com a aplicação de distúrbios na planta. O sobressinal apresentado é causado, em parte, pela distância do ponto de interesse em relação ao eixo que centraliza as rodas do robô móvel, limitada pelo acoplamento do sensor utilizado. O ajuste do ganho proporcional pode minimizar esta característica, mas o tempo de resposta poderá ser afetado. Os parâmetros de desempenho devem ser definidos para cada aplicação, optando por um sistema mais rápido ou não, de acordo com a precisão requerida.

O erro de posicionamento no deslocamento linear em  $y$  é causado pela definição de uma reta cujos pontos variam com o tempo em uma taxa de 20 mm/s, forçando o modelo a desempenhar tal velocidade linearmente. Um compensador das características dinâmicas reduziria este erro, considerando que o atraso nas acelerações seria evidenciado na modelagem dinâmica do robô.

O sensoriamento proposto pode ser adaptável em qualquer veículo, como um método de posicionamento. Contudo, estudos da superfície de deslocamento devem ser feitos a fim de determinar o seu comportamento, pois tanto a cor quanto as irregularidades do piso afetam a leitura. Descobrir possíveis erros acumulados e a resolução máxima para cada superfície e introduzi-los nos cálculos de posicionamento mantém a precisão do sistema. Porém, como o posicionamento é incremental, falhas na leitura ou uma retirada do robô da superfície irá causar uma perda de referência do mesmo.

Apesar de ser um sistema não-linear, a análise individual de cada malha de controle pode ser considerada como tal, envolvendo assim os mesmos métodos para controle linear e para o estudo da estabilidade. Deste modo, é possível a implementação de sistemas de controle mais simples, podendo ser embarcados em microcontroladores sem a necessidade de um processamento computacional mais elevado, como é o caso da aplicação de controladores cinemáticos.

## 6.1 Dificuldades Encontradas

A união de tecnologias que trabalham separadamente podem causar alguns problemas, principalmente de comunicação. A comunicação do sensor óptico com o microcontrolador demandou grande esforço, pois exigiu o estudo prévio e desenvolvimento de algoritmos, não havendo protocolos que simplesmente permitissem a conexão direta.

A falta de informações detalhadas das características físicas do robô utilizado não permitiu a modelagem dinâmica do mesmo, inviabilizando o estudo da dinâmica para modelagem e aplicação de um controlador que compense os desvios causados por características intrínsecas da planta (robô-sensor).

O acoplamento do sensor ao robô gerou dificuldades pelo fato de necessitar estar bem alinhado com a superfície de deslocamento, além de garantir que as rodas estejam com atrito suficiente para que o robô se movimente.

Embora o algoritmo de controle tenha sido deduzido, não foi embarcado de forma completa por haver a necessidade de rodar dois PID's distintos de forma simultânea, um para cada malha. Foi utilizado nos testes apenas o controlador para a velocidade angular. Como cada um precisa de um tempo de amostragem definido, o cálculo desse tempo ainda está em análise, pois não é possível a compilação simultânea dos dois.

## 6.2 Trabalhos Futuros

- Definir e aplicar os métodos para ajuste do PID de acordo com parâmetros de desempenho especificados, utilizando métodos lineares como alocação de polos e lugar das raízes para auxiliar a sintonia;
- Unir as técnicas de controle multivariável em cascata e a modelagem dinâmica da planta, garantindo assim que sistemas que trabalham com transporte de cargas elevadas possam operar com eficiência e segurança;
- Utilizar sensores de posicionamento absoluto, mais robustos e abrangentes, como o GPS (*Global Positioning System*), possibilitando a maior integração com outros projetos já implantados;
- Realizar testes para outras trajetórias e comparar o desempenho com os trabalhos anteriores;
- Criar uma *interface* gráfica que mostre a trajetória do robô e permita a programação de rotas.

## Referências

- AVAGO. *ADNS 5020 optical mouse sensor Datasheet*. [S.l.], 2007. Disponível em: <www.avagotech.com>. Citado 11 vezes nas páginas 29, 40, 41, 42, 43, 44, 45, 48, 49, 50 e 51.
- BARROS, M. *Estudo da automação de células de manufatura para montagens e soldagem industrial de carrocerias automotivas*. Tese (Doutorado) — Dissertação submetida à Escola Politécnica da Universidade de São Paulo, como requisito para a obtenção do título de Mestre em Engenharia, São Paulo, 2006. Citado na página 12.
- BORENSTEIN. *Where I AM? Sensors and Methods for Autonomous Mobile Robots Positioning*. [S.l.: s.n.], 1995. Citado na página 14.
- CANO. *Técnica de Navegação de um robô Móvel baseado em um Sistema de Visão para Integrá-lo em uma Célula Flexível de Manufatura (FMC)*. Dissertação (Mestrado) — Departamento de Engenharia Mecânica, Universidade de Brasília, 2006. Citado na página 12.
- CRUZ, C. D. L. *Control de Formación de Robots Móviles*. Tese (Doutorado) — Universidad Nacional de San Juan, San Juan, Argentina, 2006. Citado 2 vezes nas páginas 18 e 19.
- CUNHA, A. F. O que são sistemas embarcados. *Saber Eletrônica-Ano*, 2007. Citado na página 12.
- DIAS, W. S. Construção de um robô veicular autoônomo. *VII Congresso Norte Nordeste de Pesquisa e Inovação - CONNEPI*, 2012. Citado na página 13.
- DORF, H. C. *Sistema de controle Moderno*. [S.l.: s.n.], 1998. Citado na página 12.
- E.SILVA, L. Utilização do estudo de robôs seguidores de linha como estratégia de avaliação na disciplina controle de processos no curso de engenharia de computação. *Anais do XXXIV COBENGE. Passo Fundo: Ed. Universidade de Passo Fundo*, 2006. Citado na página 13.
- FERREIRA, D. S. F.; CERQUEIRA, J. d. J. F.; COSTA, A. C. P. L. d. Controle nao-linear de robô móvel unicycle utilizando linearização entrada-saída por realimentação. *Anais do XIX Congresso Brasileiro de Automática, CBA*, p. 5262–5267, 2012. Citado 4 vezes nas páginas 18, 32, 33 e 34.
- FERREIRA, I. A. P. F. Sistemas de controle e supervisão de sistemas embebidos: tipo scada. 2012. Citado 2 vezes nas páginas 27 e 28.
- FILHO, T. F. Aplicação de robôs nas indústrias. 1999. Citado na página 16.
- GOES. *Avaliação das estratégias de controle digital aplicado a um veículo guiado automaticamente por referência no chão*. Dissertação (Mestrado) — Universidade do Estado do Amazonas, 2013. Citado 2 vezes nas páginas 13 e 20.

- GOMES, G. G.; SANTOS, V. L. dos; OLIVEIRA, J. A. N. de. Controle pid embarcado em plataforma com barramento avalon. *CBA 2012*, p. 755–762, 2010. Citado na página 23.
- GRANJA, D. B.; RODRÍGUEZ, D. M. *Optimización de algoritmos para seguidor de línea utilizando técnicas de control PID con el robot Pololu 3pi e incorporación de control inalámbrico por radio frecuencia*. Tese (Doutorado), 2012. Citado na página 24.
- MARTINS, F. N. *Modelagem e Compensação da Dinâmica de Robôs Móveis e sua Aplicação em Controle de Formação*. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2009. Citado 3 vezes nas páginas 18, 35 e 36.
- MATHWORKS. *Matlab, the language of Technical Computing*. 2014. Disponível em: <<http://www.mathworks.com/products/matlab/>>. Citado na página 30.
- MELO, L. F. d. *Proposta de Simulador Virtual para Sistema de Navegação de Robôs Móveis Utilizando Conceitos de Prototipagem Rápida*. Dissertação (Mestrado) — Universidade Estadual de Campinas. Faculdade de Engenharia Mecânica. Comissão de Pós Graduação em Engenharia Mecânica., 2007. Citado 2 vezes nas páginas 36 e 37.
- NASCIMENTO, T. P. do. *Controle de Trajetória de Robôs Móveis Omni-direcionais: Uma Abordagem Multivariável*. Tese (Doutorado) — Universidade Federal da Bahia, 2009. Citado 3 vezes nas páginas 16, 37 e 38.
- OGATA, K. *Engenharia de controle Moderno*. 5. ed. [S.l.: s.n.], 2010. Citado 4 vezes nas páginas 12, 20, 21 e 22.
- OLIVEIRA, J. H. d. *Desenvolvimento de um protótipo de veículo auto guiado para célula produtiva*. Dissertação (Mestrado) — Instituto de Engenharia do Paraná, 2011. Citado 2 vezes nas páginas 13 e 14.
- PEDRINI, H. *Análise de Imagens Digitais*. [S.l.: s.n.], 2008. Citado na página 29.
- POLOLU. *Pololu 3pi Robot User's Guide*. [S.l.], 2014. Disponível em: <<http://www.pololu.com/docs/0J21/all>>. Citado 2 vezes nas páginas 24 e 46.
- SANDI, L. Sistema para navegação e guiagem de robôs móveis autônomos. *SBA. Controle e Automação*, v. 9, p. 107–118, 1998. Citado na página 16.
- SANTOS, E. A. d. S. *Logística baseada em AGVs*. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, 2013. Citado 2 vezes nas páginas 16 e 17.
- SIDNEI. Projeto: Robô seguidor de linha. 2013. Citado na página 16.
- STEPHAN, R. M. *Acionamento, comando e controle de máquinas elétricas*. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 25 e 29.