

Heiji Inuzuka
Leandro da Silva Lima

DESENVOLVIMENTO DE UM ROBÔ MÓVEL OMNIDIRECIONAL: O ROBÔ ROHL

*Monografia apresentada ao Curso de
Engenharia Mecatrônica da Universidade de
Brasília como requisito parcial para a obtenção
do título de Engenheiro de Mecatrônica.*

Orientador: Prof. Geovany Araújo Borges (ENE)

Curso de Engenharia Mecatrônica
Departamento de Engenharia Elétrica
Faculdade de Tecnologia
Universidade de Brasília
Junho, 2005

Heiji:

Dedico este trabalho aos meus pais Shoji Inuzuka e Sanae Teraoka Inuzuka e minhas irmãs Sayuri Inuzuka e Yassue Inuzuka por todo o apoio nesta longa jornada.

Leandro:

Dedico este trabalho aos meus pais Miguel do Carmo Lima Neto e Nele da Silva Lima, por terem me colocado no mundo, por estarem sempre do meu lado e por terem me incentivado a estudar.

Agradecimentos

Heiji:

Gostaria de agradecer aos meus pais pelo carinho e por todo o incentivo que me foi dado para estudar; à minha irmã Sayuri por ter convivido comigo nos últimos três anos em Brasília; ao professor Geovany A. Borges pela oportunidade oferecida e pela atenção concedida neste trabalho; ao doutorando Flávio de Barros Vidal pela companhia e pelos auxílios concedidos para este trabalho; aos demais amigos do Laboratório de Controle e Visão Computacional(LCVC) pelo apoio e companhia e aos demais amigos da Engenharia Mecatrônica da Universidade de Brasília pela convivência nesses quatro anos e meio.

Leandro:

Agradeço a Deus pelas oportunidades que tive em minha vida; à minha tia Maria Auxiliadora Carmo Lima, à minha tia Maria Inês Carmo Lima, ao meu tio e grande amigo Vilmar de Jesus Luz, à minha irmã Mara Rúbia da Silva Lima, ao meu pai Miguel do Carmo Lima Neto e à minha mãe Nele da Silva Lima, por terem contribuído financeiramente com meus estudos; à minha tia Maria Eunice da Silva de Souza e ao meu tio José Fernandez de Souza, por terem me acolhido em sua casa durante estes quatro anos e meio de Universidade; à minha namorada Ana Claudia Gonçalves Gomes, por ter sido minha companheira e por ter me ajudado a suportar a distância dos meus pais; ao professor Geovany Araújo Borges pela dedicada orientação e pelos inúmeros conselhos; a toda minha família e todos meus amigos, por terem me incentivado, em todos os momentos, a estudar; aos meus colegas e professores, por terem compartilhado comigo parte de seus conhecimentos.

Resumo

RELATÓRIO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

Autores: Heiji Inuzuka e Leandro da Silva Lima.

Orientador: Geovany Araújo Borges.

Curso de Graduação em Engenharia Mecatrônica

Brasília, junho de 2005

Resumo: Neste trabalho foi desenvolvido uma plataforma móvel omnidirecional, o robô ROHL (Robô Omnidirecional Heiji e Leandro). Este robô é equipado com três pares de rodas holonômicas, três motores de corrente contínua, circuitos de acionamento e controle, duas baterias recarregáveis e microcontroladores. Os projetos mecânicos, eletrônicos e de softwares foram desenvolvidos e implementados no robô, que pode ser utilizado como plataforma de desenvolvimento para vários projetos de pesquisa em robótica móvel. Os softwares desenvolvidos permitiram a realização de vários testes, nos quais verificou-se a omnidirecionalidade do robô, bem como a detecção de algumas falhas do sistema de controle. Estas falhas levaram à substituição da plataforma de controle utilizada, um DSP, por um outro sistema de controle via PC, que será substituído por uma placa Vortex86-6047. A substituição do sistema de controle possibilitou a realização de um controle de velocidade. O controle implementado obteve excelente desempenho quando aplicado ao robô em suspensão, uma vez que acompanhou a referência com sobrepasso pequeno. Ao se aplicar um controle de velocidade para o robô no solo, os resultados foram bastante satisfatórios, apesar da existência de deslizamento das rodas.

Abstract

UNDERGRADUATE REPORT

Authors: Heiji Inuzuka and Leandro da Silva Lima

Supervisor: Geovany Araújo Borges.

Curso de Graduação em Engenharia Mecatrônica

Brasília, June, 2005

Abstract: An omnidirectional wheeled platform was developed in this project, the robot named ROHL (Robô Omnidirecional Heiji e Leandro). This robot is equipped with three pairs of holonomic wheels, three driving DC motors, driving and controlling circuits, two rechargeable batteries and microcontrollers. The mechanical, electronic and software design had been developed and implemented in the robot, that may be used as a development platform for another research projects in mobile robotics. The developed software had allowed the accomplishment of some tests, in which were verified the omnidirection of the robot, as well as the detection of some imperfections about the control system. These imperfections had led to the substitution of the control platform based on DSP by another system of control through the computer, that will be substituted by a control platform based on vortex86-6047 processor with a PC-104 bus. The substitution of the control system made possible the accomplishment of velocity control. The implemented control got excellent results when applied to the robot in suspension because the results followed the reference with small overshoot. By applying a velocity control for the robot in the ground, the results had been sufficiently satisfactory, even with the existence of landslide of the wheels.

Conteúdo

1	Introdução	1
1.1	Objetivo	1
1.2	Motivação	1
1.3	Robótica: uma breve introdução	2
1.3.1	Robótica Móvel	2
1.3.2	Robôs Omnidirecionais	5
1.3.3	Rodas Holonômicas	5
1.4	O projeto ROHL	6
1.5	Apresentação do Manuscrito	6
2	Fundamentação Teórica	7
2.1	Modelamento Cinemático de robôs móveis	7
2.2	Odometria	9
2.3	Controle de velocidade	11
2.4	Ambiente de desenvolvimento em tempo real: RTX (Real Time Extension)	12
2.5	Comunicação RS485	12

2.6	Acionamento	14
3	Desenvolvimento	16
3.1	Estrutura mecânica	16
3.1.1	Base e Corpo	16
3.1.2	Caixas de engrenagem	17
3.1.3	Rodas	18
3.2	Modelamento Cinemático	19
3.3	Alimentação	21
3.3.1	Circuito da fonte	21
3.3.2	Motores e Baterias	22
3.3.3	Carregador de baterias	22
3.4	Acionamento dos motores	23
3.4.1	Placa de potência	23
3.5	Processamento com Arquitetura DSP(Processador Digital de Sinais)	24
3.5.1	Placa do DSP	24
3.5.2	Placa de Geração de MLP (Modulação em Largura de Pulso)	25
3.6	Processamento com Arquitetura x86	27
3.6.1	Placa Vortex86-6047	27
3.6.2	Placa dos microcontroladores	28
3.6.3	Gravador de AVRs	28
3.6.4	Placa de conversão RS232/RS485	29
3.7	Desenvolvimento de <i>Software</i>	30
3.7.1	Módulo DSP	30
3.7.2	Módulo PC com AVRs	31
3.8	Protocolo de comunicação	32
3.9	Fotos do ROHL	34
4	Experimentos	35
5	Conclusões	45

CONTEÚDO	vi
Bibliografia	47
Anexos	49
A Esquemáticos	50
B Descrição do CD	55

Lista de Figuras

1.1	Roda sueca.	6
2.1	(a) Orientação do robô, (b) Parâmetros da Roda Sueca.	8
2.2	RTX e Windows NT trabalhando juntos.	13
2.3	Pinos do CI DS485N.	14
2.4	(a) Ponte H, (b) Acionamento de motores com ponte H.	15
3.1	Dimensões da base.	17
3.2	(a) Foto do ROHL com plataforma 1, (b) Foto do ROHL com plataformas 1 e 2, (c) Foto do ROHL plataformas 1, 2 e 3.	17
3.3	Foto da caixa de engrenagem em detalhe.	18
3.4	(a) Foto da roda utilizada no robô, (b) Foto do par de rodas.	18
3.5	Localização das rodas no robô.	19
3.6	Referencial do robô.	20
3.7	(a) Circuito da fonte de alimentação, (b) Foto do circuito da fonte de alimentação.	22
3.8	Circuito de carregamento das baterias.	23
3.9	(a) Circuito de um sensor óptico, (b) Circuito de acionamento de um motor.	24

3.10	Foto do circuito do DSP.	25
3.11	(a) Circuito de geração de MLP, (b) Circuito de tratamento do sinal do <i>resolver</i> . .	26
3.12	Sinais analógico e digital do <i>resolver</i>	26
3.13	Foto da placa Vortex86-6047.	27
3.14	Circuito de um AVR.	29
3.15	(a) Foto do gravador de AVRs, (b) Esquemático do gravador de AVRs.	30
3.16	Foto da placa de conversão RS232/RS485.	31
3.17	Comunicação PC com AVRs.	32
3.18	Protocolo de comunicação.	33
3.19	(a) Foto do ROHL com DSP, (b) Foto do ROHL visto por baixo, (c) Foto dos componentes da primeira plataforma, (b) Foto do ROHL com AVRs.	34
4.1	(a) Identificação do motor 1, (b) Validação do motor 1.	39
4.2	(a) Identificação do motor 2, (b) Validação do motor 2.	40
4.3	(a) Identificação do motor 3, (b) Validação do motor 3.	41
4.4	Diagrama de blocos do controle.	42
4.5	Resposta controle PI para frequência de corte 1,0 Hz do filtro de medida de velocidade. .	42
4.6	Resposta controle PI para frequência de corte 1,5 Hz do filtro de medida de velocidade. .	43
4.7	Controle dos motores com o robô em contato com o chão.	44
A.1	Esquemático da placa de potência	51
A.2	Esquemático da placa do DSP	52
A.3	Esquemático da placa de geração de MLP	53
A.4	Esquemático da placa dos microcontroladores	54

Lista de Tabelas

2.1	Tabela verdade na transmissão	14
2.2	Tabela verdade na recepção.	15
3.1	Tabela de parâmtros das rodas do ROHL	19

Introdução

1.1 Objetivo

Este trabalho tem por objetivo a construção de uma plataforma móvel omnidirecional com rodas holonômicas, bem como os circuitos de alimentação, de acionamento e de controle de velocidade. A longo prazo tem-se como objetivo construir um robô assistente para portadores de deficiências físicas.

1.2 Motivação

A operação em ambientes complexos de larga escala, que se modificam dinamicamente, compostos tanto de obstáculos estáticos como de obstáculos móveis são problemas relacionados à robótica móvel. Para que o robô possa se locomover neste tipo de ambiente, ele deve ser capaz de adquirir e utilizar conhecimento sobre o mesmo, estimar uma posição dentro dele, possuir a habilidade de reconhecer obstáculos e responder em tempo real as situações que possam ocorrer neste ambiente. Além disso, todas estas funcionalidades devem operar em conjunto.

A localização é essencial em sistemas de controle para robôs móveis autônomos, possibilitando a navegação, a exploração e adaptação do mapa do ambiente pelo robô, bem fazer com que ele siga planos de percurso. A partir das entradas sensoriais, o robô deve ser capaz de inferir sua posição

e orientação relativa a um mapa global. De acordo com o artigo [Heinen 2002], a integração de um módulo localizador, em um sistema de controle robótico, amplia sua capacidade de navegação e sua robustez.

Algumas aplicações possíveis para os robôs móveis são: transporte, vigilância, inspeção, limpeza de casas, exploração espacial e auxílio a deficientes físicos. Entretanto, os robôs móveis autônomos ainda não são muito utilizados em aplicações domésticas ou industriais, principalmente devido à falta de um sistema de controle robusto, confiável e flexível que permitiria que estes robôs operassem em ambientes dinâmicos, pouco estruturados, e habitados por seres humanos.

1.3 Robótica: uma breve introdução

Conforme visto na página da *web* ¹, conceituar a palavra robô é uma tarefa difícil. Essa palavra deriva do tcheco e significa "trabalho forçado", sendo que foi inventada pelo escritor tchecoslovaco Karel Capek em 1920. Em [Michelon 2000], o *Robot Institute of América* define um robô como um manipulador multifuncional e reprogramável, projetado para movimentar ferramentas ou dispositivos especializados através de movimentos variáveis e programados para a execução de uma infinidade de tarefas.

Segundo a página da *web* ², as crescentes demandas de uma população mundial, que hoje ultrapassa os seis bilhões, têm impulsionado um avanço exponencial, tanto científico quanto tecnológico. A revolução industrial abriu as portas à produção em larga escala de produtos e bens de consumo, fomentando um ciclo de desenvolvimento de máquinas cuja complexidade vem aumentando a cada dia. Gradualmente, essas máquinas se tornaram engenhocas eficientes com alto grau de automatismo, mas com pouca flexibilidade quanto aos tipos e mesmo a pequenas variações de um mesmo produto. A demanda por flexibilidade e maior qualidade dos itens fabricados teve como resultado a criação, nos anos 60, dos robôs manipuladores, hoje presentes nos diversos segmentos da indústria. De acordo com a mesma referência, atualmente existem cerca de um milhão de robôs manipuladores em operação no mundo, em aplicações que abrangem desde a indústria automobilística até a mais recente e rentável aplicação em biotecnologia. Como consequência, a complexidade das tarefas demandadas por esse progresso continua aguçando o anseio pela criação de máquinas com competências similares às do ser humano.

1.3.1 Robótica Móvel

Na página da *web* ² tem-se que por volta dos anos 70 teve início, em maior intensidade, as pesquisas em robôs móveis. Diferentemente dos manipuladores, cuja incumbência é de realizar continuamente seqüências de operações e com alto grau de repetições em ambientes controlados e bem

¹<http://paginas.fe.up.pt/~eol/SP-leec/TRABALHOS/ROB/histframe.html>, acesso em 01/02/2005

²<http://ftp.inf.pucpcaldas.br/CDs/SBC2003/pdf/arq0262.pdf>, acesso em 01/02/2005

estruturados, os robôs móveis tipicamente têm como missão atuar em ambientes dinâmicos e não estruturados. Colocado dessa forma, o espectro e complexidade dos problemas que os sistemas robóticos móveis precisam resolver é muito amplo. A aplicabilidade dos robôs móveis em tarefas como a exploração espacial, inspeção submarina de cabos e oleodutos, transporte de cargas de alto risco e o rastreamento de minas terrestres, tem, mais recentemente, apontado para um crescimento considerável da pesquisa nessa área. Robôs móveis são agentes autônomos cujo habitat pode ser o solo, o ar, a água ou ainda o espaço, e que são dotados de diversos sistemas cujas propriedades possibilitam a execução de tarefas sem a necessidade de intervenção externa. Uma das principais características do comportamento autônomo decorre da necessidade do sistema em se adaptar, e eventualmente se auto-organizar, para contornar situações imprevistas. Robôs deslocam-se em um ambiente através das informações obtidas de seu conjunto de sensores e que são adequadamente processadas em um ou mais processadores. Isso o habilita a se deslocar, interagir e perceber as variações sofridas pelo ambiente agindo de forma "inteligente". A percepção, neste contexto, é entendida como a correspondência entre um modelo do mundo, definido internamente no robô, e o mundo real, externo a este. Como consequência, as ações resultantes de um robô móvel decorrem de processamento intenso ("raciocínio"), que constitui a sua capacidade de inferência sobre o mundo.

Dando continuidade à referência do parágrafo anterior, não existe uma taxonomia única para robôs móveis, e as existentes são baseadas em dimensões como o meio em que atuam, a classe de tarefas para a qual foi projetado, o tipo de locomoção, a forma e a flexibilidade de sua estrutura física, sua competência de navegação e modo de guiar.

Quanto ao meio no qual irá navegar, os robôs podem ser classificados em :

- Terrestre - quando a navegação se dá no solo através de rodas, pernas (duas, quatro ou seis pernas são as configurações mais usuais);
- Aéreo - quando a navegação ocorre no ar. Esses robôs podem ser movidos por empuxo aerodinâmico (aviões, helicópteros) ou por flutuação (dirigíveis e balões);
- Aquático(barcos) ou subaquático (submarinos);
- Espaciais - navegação espacial. Robôs utilizados na exploração interplanetária ou a bordo de espaçonaves.

Em geral, quando se projeta um robô pensando na especificidade de sua aplicação ou tarefa, os robôs móveis podem ser classificados como:

- Veículo de Operação Autônoma (*Automated Guided Vehicle* - AGV);
- Veículo Aéreo não Tripulado (*Aerial Unmanned Vehicle* - AUV);
- Veículo Submarino de Operação Remota (*Remotely Operated Vehicles* - ROV);

- Robôs para pesquisa;
- Robôs educativos;
- outros tipos (companhia, trabalhos domésticos, etc.).

Por outro lado, a capacidade de tomada de decisão, também denominada de nível de autonomia do robô, favorece a classificação dos robôs móveis em:

- Tele-operado - Operador humano controla remotamente as ações do robô;
- Semi-autônomo - Parte do controle é realizada por operador humano, porém o robô possui a capacidade de ação e tomada de decisão autônoma;
- Totalmente autônomo - O robô possui capacidade de decisão, controle e ação sem intervenção humana.

Ainda com relação à página da *web* ², uma importante característica de um robô móvel é a capacidade de (re)agir ao ambiente em seu entorno a partir das informações obtidas de seu sistema sensorial. Um sensor é um dispositivo capaz de detectar e quantificar parâmetros específicos por meio de elementos que transformam uma grandeza física em outra, tipicamente um sinal elétrico (transdutores). São eles os responsáveis por fornecer informações sobre os estados do sistema. Existem dois tipos de sensoriamento que são utilizados em robôs móveis:

- proprioceptivo - medem estados internos do robô, como as baterias;
- exteroceptivo - medem sensoriamentos externos, como ultra-som para ver se há obstáculo.

Em [Campion, Bastin e D'Andréa-Novel 1996] os robôs móveis com rodas foram classificados em cinco tipos, cuja nomenclatura é da seguinte forma: "Tipo(δ_m , δ_s)". Nessa nomenclatura, (δ_m) é o grau de mobilidade do robô (δ_s) é o grau de navegabilidade. A classificação adotada por Campion é a descrita a seguir.

- Tipo (3,0) - este tipo não possui rodas convencionais fixas e nem rodas orientáveis centradas. Eles são chamados de robôs omnidirecionais, por se moverem em qualquer direção do plano sem necessidade de reorientação. O robô ROHL se enquadra nesse tipo de robô;
- Tipo (2,0) - este tipo não possui rodas convencionais de orientação centrada, mas possuem ou uma ou várias rodas convencionais fixas com eixo comum;
- Tipo (2,1) - este tipo não possui rodas convencionais fixas e possui pelo menos uma roda de orientação centrada, quando houver mais de uma, as orientações das mesmas devem ser coordenadas, de forma que δ_s seja igual a 1;

- Tipo (1,1) - este tipo possui uma ou mais rodas convencionais fixas com eixo comum e uma ou mais rodas de orientação centrada, com a condição de que o centro de pelo menos uma delas não esteja localizado sobre o eixo das rodas fixas e suas orientações devem ser coordenadas, de forma que δ_s seja igual a 1;
- Tipo (1,2) - este tipo não possui rodas convencionais fixas e pelo menos duas rodas de orientação centrada, quando houver mais de duas, as orientações das mesmas devem ser coordenadas, de forma que δ_s seja igual a 2.

Atualmente é muito comum a substituição de robôs móveis de quatro rodas por robôs com três rodas, conforme visto em [Kim, Lee e Kwon 2000]. Uma vantagem destes sistemas de três rodas é a garantia de que nenhuma das rodas ficará suspensa.

1.3.2 Robôs Omnidirecionais

Um robô omnidirecional é um robô capaz de movimentar-se em qualquer direção no plano sem necessidade de rotação do próprio corpo. Nos robôs mais comuns, com duas rodas, para obter movimento em uma determinada direção é necessário que ele esteja apontando naquela direção.

Muitas configurações são utilizadas para obter robôs omnidirecionais, entretanto, são mais utilizados robôs com rodas holonômicas ou rodas orientáveis com eixo deslocado.

1.3.3 Rodas Holonômicas

De acordo com Campion, Bastin e D'Andrea-Novel, existem duas classes básicas de rodas: rodas convencionais e rodas holonômicas (denominada por ele de rodas suecas). Por sua vez, as rodas convencionais são subdivididas em três subclasses: rodas fixas, rodas de orientação centrada e rodas de orientação descentrada.

As rodas holonômicas são rodas que combinam movimento na direção paralela ao plano da roda com o deslizamento na direção ortogonal a este plano. O resultado dessa combinação de movimentos é a adição de mais um grau de liberdade à roda, ou seja, esses tipos de rodas possuem dois graus de liberdade.

Colocando-se alguns roletes, dispersos ao longo da superfície de contato da roda e que fazem o contato direto com o solo (Figura 1.1), diminui-se o atrito na direção de deslizamento. São encontradas diversas nomenclaturas para este tipo de roda como, por exemplo, rodas suecas no artigo [Campion, Bastin e D'Andréa-Novel 1996], rodas universais em [Wada e Mori 1996] e rodas *Mecanum* em [Nagatani, Sofue e Tanaka 2000].



Figura 1.1: Roda sueca.

1.4 O projeto ROHL

Neste projeto foi construído um robô móvel omnidirecional, o robô ROHL. Ele foi implementado com o intuito de ser utilizado como plataforma para implementação de controle de velocidade e odometria. Este robô é equipado com três rodas holonômicas. Estas rodas foram colocadas nos vértices de um triângulo equilátero.

O trabalho foi realizado em duas etapas, projeto de graduação um e projeto de graduação dois. Na primeira etapa do projeto, foi realizado o projeto mecânico e eletrônico do ROHL, bem como sua construção. Foram, também, desenvolvidos alguns softwares que permitiram a realização de alguns testes com o robô, com estes testes, foram detectados alguns problemas. Na segunda etapa do projeto foram encontradas e aplicadas soluções para os problemas detectados na primeira etapa. Com isso, foi possível obter o modelo que representa cada um das rodas, como também, a obtenção de um controle para a velocidade das rodas.

1.5 Apresentação do Manuscrito

Este trabalho está organizado em cinco capítulos e dois apêndices. No Capítulo 2 são apresentados alguns conceitos teóricos empregados no desenvolvimento do robô. As descrições de projeto e implementação da estrutura mecânica e dos circuitos eletrônicos estão no Capítulo 3. O Capítulo 4 apresenta alguns experimentos realizados com o robô e os respectivos resultados obtidos. No Capítulo 5 são apresentadas as conclusões deste trabalho e as propostas de trabalhos futuros. Os esquemáticos dos circuitos implementados estão no Apêndice A. A descrição dos arquivos do CD em anexo é apresentada no Apêndice B.

Fundamentação Teórica

Foram selecionados alguns artigos relacionados a odometria, modelamento cinemático, métodos de estimativa de posição de robôs móveis e métodos de controle de velocidade. Estes artigos foram utilizados como ponto de partida para construção do robô ROHL. Com eles foi possível rever alguns aspectos teóricos e práticos que pudessem ser aplicados em nosso trabalho. Alguns manuais de equipamentos, bem como *datasheets* de componentes eletrônicos, também foram estudados de forma a possibilitar a utilização dos mesmos no ROHL. Serão relacionados a seguir, os conceitos que foram extraídos destes materiais.

2.1 Modelamento Cinemático de robôs móveis

A posição das rodas holonômicas é descrita em [Campion, Bastin e D’Andréa-Novel 1996] com quatro parâmetros: α , β , l e γ (Figura 2.1). A expressão de restrição de movimento, apresentada no mesmo artigo, para este tipo de roda foi representada por:

$$[-\sin(\alpha + \beta + \gamma) \cos(\alpha + \beta + \gamma) l \cos(\beta + \gamma)] R(\theta) \dot{\xi} + r \cos \gamma \dot{\varphi} = 0, \quad (2.1)$$

com $R(\theta) = \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\xi = \begin{pmatrix} x \\ y \\ \theta \end{pmatrix}$, r sendo raio da roda, θ sendo a orientação do robô com relação à uma base inercial e $\dot{\varphi}$ a velocidade angular da roda

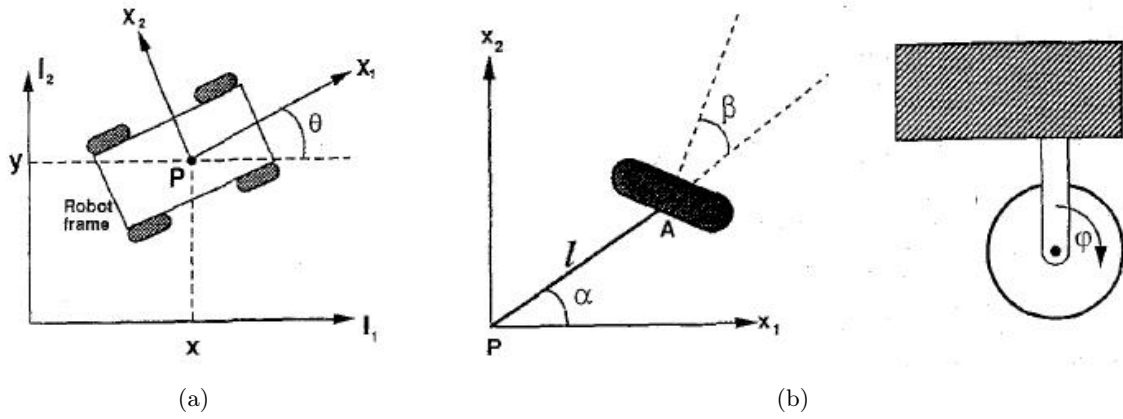


Figura 2.1: (a) Orientação do robô, (b) Parâmetros da Roda Sueca.

Em [Campion, Bastin e D'Andréa-Novet 1996], também foi apresentado um modelo cinemático geral para robôs móveis com rodas. Este modelo é dado por:

$$\frac{d\xi}{dt} = R^T(\theta)S(\beta_c)\eta \quad (2.2)$$

e

$$\frac{d\beta_c}{dt} = \zeta, \quad (2.3)$$

em que η representa um vetor variável no tempo e $S(\beta_c)$ uma matriz com os ângulos de orientação das rodas.

Para robôs sem roda de orientação centrada, a matriz S é constante e o modelo anterior se reduz à equação:

$$\frac{d\xi}{dt} = R^T(\theta)S\eta. \quad (2.4)$$

Outro modelo cinemático foi apresentado em [Kim, Lee e Kwon 2000], baseado em aproximação geométrica, que é empregado em robôs móveis com três rodas que têm movimento de tração e movimento de orientação.

Um modelo cinemático para veículos omnidirecionais de rodas com movimento de tração e de orientação foi apresentado em [Wada e Mori 1996], mas que apresentam um deslocamento ao longo

da sua direção de tração, entre o eixo da roda e o eixo de orientação, eles apresentaram também a cinemática inversa obtida a partir da pseudo-inversão matricial.

Pin e Killough, em [Pin e Killough 1994], apresentaram um modelo cinemático derivado a partir de uma nova configuração de rodas apresentada por eles, chamadas de rodas ortogonais. Essa configuração de rodas foram projetadas para serem aplicadas na construção de robôs móveis omnidirecionais.

A partir de todos estes modelos, e observando o desenvolvimento matemático apresentado em cada um, foi possível obter o modelo cinemático do robô desenvolvido neste trabalho. Este modelo cinemático será apresentado mais adiante, na seção 3.2.

2.2 Odometria

Conforme o artigo [Borenstein e Feng 1996], odometria é um método que leva em consideração medidas de deslocamento das rodas e é extremamente utilizado para determinar o posicionamento instantâneo de um robô móvel, é simples de ser implementada, de baixo custo e realizável em tempo real. A desvantagem da odometria é o acúmulo ilimitado de erros. Dois métodos básicos de estimativa de posição são empregados em robótica móvel: posicionamento absoluto e relativo. A odometria se baseia no método relativo.

Um modelo odométrico para um robô móvel omnidirecional, equipado com quatro rodas holonômicas de roletes inclinados de um ângulo de 45° em relação ao plano da roda, foi apresentado em [Nagatani, Sofue e Tanaka 2000]. Todavia, devido a um escorregamento das rodas na direção lateral, o movimento produzido por elas não se dava exatamente na direção de 45° . Portanto, foi assumido um vetor virtual de movimento com direção, que foi determinado experimentalmente. O modelo odométrico obtido para este robô é representado por:

$$P[t + \tau] = P[t] + \tau \begin{bmatrix} v_f[t] \cos \theta[t] - v_s[t] \sin \theta[t] \\ v_f[t] \sin \theta[t] + v_s[t] \cos \theta[t] \\ \dot{\theta}[t] \end{bmatrix}, \quad (2.5)$$

em que:

- $\dot{\theta}[t]$: velocidade angular do robô;
- $\tau[t]$: período de amostragem;
- $P[t]$: vetor de posição do robô;
- $\theta[t]$: ângulo de orientação das coordenadas do robô em relação ao sistema de referencia global;
- $v_f[t]$: velocidade longitudinal;

- $v_s[t]$: velocidade lateral.

De acordo com o artigo [Nagatani, Sofue e Tanaka 2000], se um robô se locomove em uma superfície regular, e sem escorregamento das rodas, o método de odometria é suficiente para estimar a posição do robô. Caso contrário, o valor de v_s irá mudar durante o percurso, corrompendo os dados obtidos com odometria.

Borenstein e Feng apresentaram, em [Borenstein e Feng 1996], métodos práticos para mensurar e reduzir erros de odometria em robôs móveis com transmissão diferencial. Os métodos apresentados são aplicáveis a dois tipos de erros sistemáticos, considerados dominantes, por eles. Estes erros são erros associados à incerteza sobre o ponto de contato efetivo das rodas e ao diâmetro diferente de cada roda. Os erros de odometria que Borenstein e Feng procuraram corrigir também são preocupantes para a odometria do robô ROHL.

Para medir os erros de odometria, Borenstein e Feng programaram o robô para percorrer uma trajetória quadrática, tanto no sentido horário quanto no sentido anti-horário. Para o caso do erro associado à incerteza sobre o ponto de contato efetivo das rodas, foi verificado um ângulo nos vértices do quadrado, ao invés de 90° . Esse ângulo pode ser calculado pela posição do robô no final do percurso e pela medida do lado do quadrado.

Para o caso do erro associado ao diâmetro diferente de cada roda, um erro de orientação incremental é gerado em cada lado. Este erro de orientação é devido a uma curvatura no movimento. O valor deste erro também pôde ser calculado pelos mesmos parâmetros que permitiram calcular o valor do erro associado à incerteza sobre o ponto de contato efetivo das rodas.

Em [Wang 1998] foram analisados alguns erros associados à odometria de um robô com duas rodas de tração, sendo que estes erros podem ser ocasionados por diversas fontes. A partir desta análise, foi obtida a estimativa de localização e as matrizes de covariâncias de incertezas. As imprecisões nas medidas odométricas associadas a cada roda, foram assumidas como tendo distribuição normal, idêntica e independentemente distribuída (iid). Desta forma, a estimativa de posição do robô foi representada pela equação:

$$U_n = U_{n-1} + \Delta U_n. \quad (2.6)$$

Com essa representação, a matriz de covariância associada a essa estimativa foi dada pela equação:

$$\text{cov}[U_n] = \text{cov}[U_{n-1}] + \text{cov}[\Delta U_n] + \text{cov}[U_{n-1}, \Delta U_n] + \text{cov}[\Delta U_n, U_{n-1}]. \quad (2.7)$$

em que $\text{cov}[*]$ significa auto-covariância e $\text{cov}[*,*]$ significa covariância cruzada.

Este método pode ser empregado na obtenção da matriz de covariância de incerteza do robô ROHL, desde que seja utilizada uma estimativa de posição com as devidas alterações, que o torne compatível com nosso modelo.

2.3 Controle de velocidade

Em [Thuilot, D'Andréa-Novel e Micaelli 1996] são desenvolvidas duas técnicas de controle para robôs equipados com pelo menos duas rodas direcionais e suas respectivas leis de controle. Entretanto, vale ressaltar que robôs com este tipo de rodas não são o foco deste trabalho, no entanto, os métodos apresentados são válidos como ponto de partida para as estratégias de controle a serem aplicadas.

A primeira técnica de controle foi denominada realimentação dinâmica e sua lei de controle de lei de caminho da trajetória. A segunda técnica foi denominada realimentação tempo-variada e sua lei de controle de lei de estabilização. Essas técnicas podem ser aplicadas somente em robôs com rodas direcionais montadas de forma que não deslizem. Para estes robôs, o modelo em espaço de estados proposto foi o da equação:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \sin \alpha \\ -\cos \alpha \\ \sigma \end{pmatrix} R \Psi_1, \quad (2.8)$$

em que:

- R é raio da roda;
- Ψ_1 é velocidade angular da roda 1;
- x e y representam a posição do centro da roda 1 em relação ao sistema de referência principal;
- θ é a orientação do sistema de referência que passa pelo centro da roda 1;
- α é o ângulo entre a coordenada x do robô e o plano perpendicular ao da roda 1;
- σ é a curvatura instantânea vista do centro da roda 1.

Como apresentado no artigo [Thuilot, D'Andréa-Novel e Micaelli 1996], este modelo é completamente controlável, não completamente linearizável pelas leis de realimentação de espaço estático e não estabilizável pelas leis de tempo-invariante. Mas, ao se aplicar a primeira técnica de controle, o sistema pode ser linearizado. Para que essa técnica se torne possível de ser aplicada, o sistema não pode apresentar singularidades intrínsecas e nem configuração de repouso conhecida. O problema de estabilização pode ser solucionado com a aplicação da segunda técnica de controle, mas para isso, o sistema deve ter configuração de repouso conhecida. Com a aplicação da segunda técnica, o modelo em espaço de estados foi rescrito e dele foi extraído um vetor que representasse uma esfera unitária. Cada não-singularidade foi associada com pontos diametricamente opostos neste vetor.

Uma lei de controle global foi obtida com a fusão das duas técnicas de controle. Essa lei é composta de três etapas: na primeira, as variáveis são inicializadas por um sistema de controle de malha aberta; na segunda, a lei de realimentação dinâmica é aplicada; e na última, a lei de realimentação dinâmica é substituída pela lei de tempo-variante. Os testes realizados com a aplicação dessa lei, em trajetórias sem obstáculos e em trajetórias com obstáculos, mostraram sua eficácia.

2.4 Ambiente de desenvolvimento em tempo real: RTX (Real Time Extension)

Como descrito em [Verturcom Inc 1998, 1999], RTX é uma extensão para o Windows NT que possui capacidade de realizar atividades em tempo real. O RTX possibilita a execução de componentes ou módulos que requerem um tempo de resposta rápido e preciso, podendo trabalhar juntamente com aplicações que não utilizam tempo real.

O RTX adiciona um subsistema em tempo real, denominado RTSS, ao Windows NT (ver Figura 2.2). Este subsistema possui ambiente de execução e API próprios. O RTSS possui escalonador de atividades próprio, sendo que as atividades do RTSS têm preferência com relação às atividades do Windows NT. Para que o RTSS não domine totalmente o sistema, há um período máximo que este pode atuar continuamente no sistema. O RTSS também possibilita o tratamento de interrupções em tempo real. A camada de abstração de hardware (HAL) mantém isolamento entre as interrupções do RTSS e do Windows NT. Quando as interrupções do RTSS estão sendo tratadas, as interrupções do Windows NT são mascaradas, sendo que as interrupções do Windows NT não podem mascarar as do RTSS.

O RTSS também disponibiliza objetos de comunicação entre processos para possibilitar a comunicação e sincronização entre programas em tempo real e não tempo real. O objeto de memória compartilhada é uma região de memória física que pode ser mapeada num espaço de endereço virtual de um processo. O objeto semáforo é um objeto de sincronização que mantém uma contagem entre zero e certo valor máximo especificado. O objeto *Mutex* é um objeto de sincronização que é assinalado quando não pertence a nenhuma atividade.

2.5 Comunicação RS485

RS485 é um padrão de comunicação serial que se baseia na transmissão de um sinal através da diferença entre duas tensões. Essas tensões são transmitidas num par de fios trançados, tendo assim robustez contra ruídos. Isto pode ser explicado pelo fato de que um possível ruído afetaria ambos os fios, mantendo desta forma a diferença entre essas duas tensões.

O fato de possuir robustez contra ruídos permite uma transmissão a longas distâncias, mas

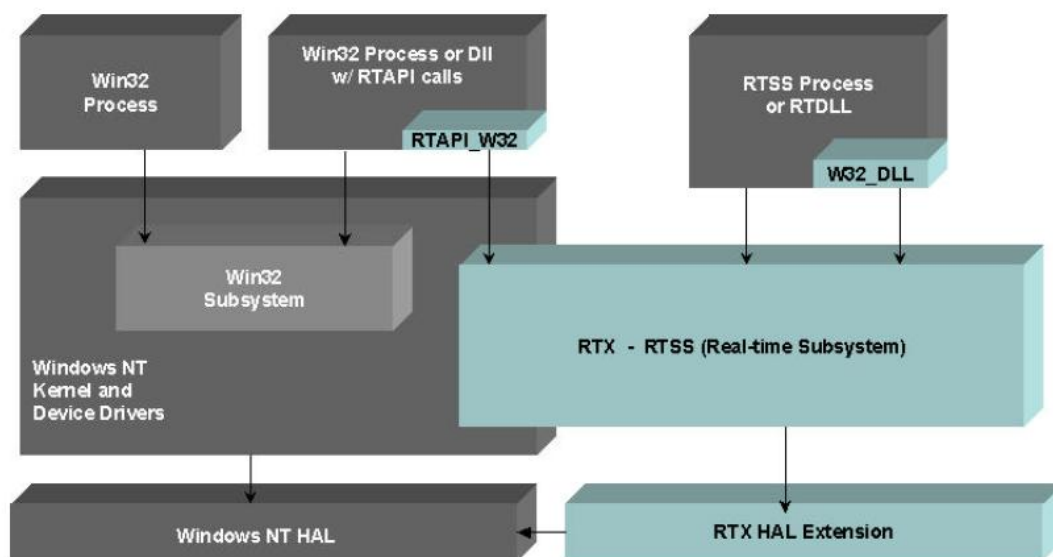


Figura 2.2: RTX e Windows NT trabalhando juntos.

quanto maior essa distância, menor deve ser a taxa de transferência para não ocorrer erros nesta transmissão.

Neste padrão de comunicação pode haver vários dispositivos conectados ao mesmo barramento, sendo que para mais de trinta e dois dispositivos, deve-se utilizar transmissores ou repetidores para a melhoria dos sinais.

Como pode ser visto na página da *web* ³, normalmente coloca-se um resistor de 120 Ω em uma das pontas de uma linha de transmissão entre os sinais A e B. Isto para que a diferença entre os dois sinais seja perto de zero quando não houver nenhuma transmissão.

O CI utilizado neste projeto para fazer a conversão entre RS485 e RS232 é o DS485N, podendo operar em uma transmissão de até 2.5 Mbps. Como pode ser visto no *datasheet* do DS485, este CI possui oito pinos: RO, RE, DE, DI, GND, A, B e VCC; que podem ser vistos na Figura 2.3.

RO é o pino de saída para recepção, tendo como sinal alto quando o sinal A é maior do que o sinal B em 200 mV e sendo baixo caso contrário. RE é o pino que habilita o pino de saída para recepção quando em nível lógico baixo. DE é o pino que habilita o pino de entrada da transmissão quando em nível lógico alto. DI é o pino de entrada da transmissão. Quando DI está em nível lógico baixo, A fica em nível lógico baixo e B em alto; e estando em nível lógico alto, A fica em nível lógico alto e B em baixo. A e B são os sinais de comunicação pela diferença entre os dois. As tabelas verdades de transmissão e recepção podem ser vistas, respectivamente, nas Tabelas 2.1 e 2.2

³<http://www.hw.cz/english/docs/rs485/rs485.html>, acesso em 13/06/2005

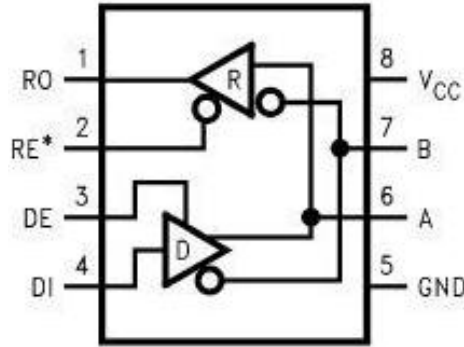


Figura 2.3: Pinos do CI DS485N.

RE*	DE	DI	A	B
X	H	H	H	L
X	H	L	L	H
X	L	X	Z	Z

Tabela 2.1: Tabela verdade na transmissão

2.6 Acionamento

Para o acionamento dos motores de corrente contínua podem ser utilizados sinais modulados em largura de pulso (MLP), mais conhecidos como PWM (*Pulse Wide Modulation*). Uma vez que o modelo do motor representa um filtro passa-baixas, ele responde basicamente ao valor médio do sinal MLP.

Como o DSP não disponibiliza canais de MLP, utilizados para acionamento dos motores, torna-se necessário gerar estes sinais externamente. Uma opção para gerar sinal MLP é o CI 8254, da *Intel*, que é um temporizador com intervalo de tempo programável. Ele possui um barramento de dados interno de 8 bits. Este barramento pode ser acessado pelo microcontrolador, de forma a programar o modo de operação e fazer a leitura dos contadores internos. O 8254 possui três contadores independentes, que quando programados de forma correta, podem fornecer sinais de MLP.

O nível de tensão do sinal MLP disponível por microcontroladores ou por CIs que geram tais sinais, pode não corresponder ao nível de tensão necessário para alimentar os motores. A corrente exigida pelos motores, de modo geral, não é compatível com a corrente máxima que estes componentes suportam. Dessa forma torna-se necessário a utilização de um circuito que converta o nível de tensão do sinal MLP e que permita os motores operarem com a corrente necessária. O circuito como o da Figura 2.4 (a), conhecido como ponte H, pode ser utilizado para estes fins. A Figura 2.4 (b) mostra como o acionamento de um motor pode ser feito com este circuito.

RE*	DE	A-B	R0
L	L	$\geq +0.2V$	H
L	L	$\leq -0.2V$	L
H	X	X	Z
L	L	OPEN*	H

Tabela 2.2: Tabela verdade na recepção.

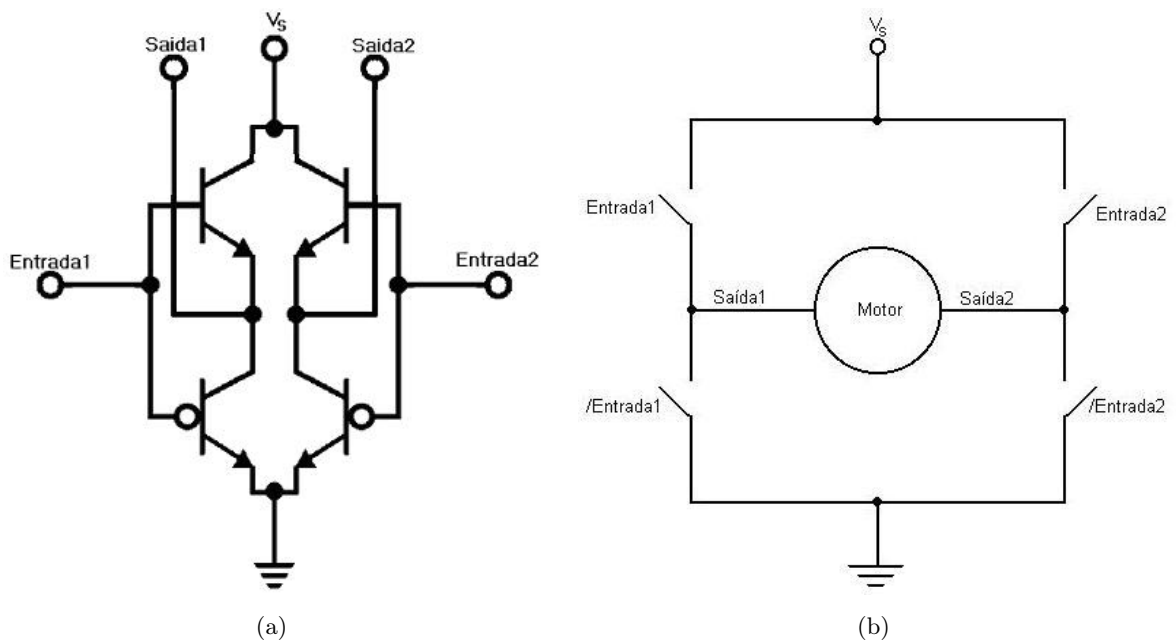


Figura 2.4: (a) Ponte H, (b) Acionamento de motores com ponte H.

Desenvolvimento

3.1 Estrutura mecânica

3.1.1 Base e Corpo

A base foi projetada de forma que as rodas se localizassem nas pontas de um triângulo equilátero de 25 centímetros de lado. Retirando três cantos desse triângulo, para inserir as rodas neste espaço, o formato da base (Figura 3.1) tornou-se um hexágono irregular com três lados de 25 centímetros e os outros três com 10 centímetros.

A base do robô foi construída com barras de alumínio em formas de 'L', que foram dobradas para adquirir a forma do hexágono desejado. Para dobrar essa barra foi necessário fazer um corte em forma de 'V', em um dos lados do 'L' da barra, em cada ponto de dobra.

Cada plataforma foi feita de chapas de alumínio com dimensões iguais às da base. As paredes de sustentação foram feitas com barras de alumínio em formas de 'L', assim como foi feita a base, mas com formato retangular. Para a sustentação da segunda plataforma há três paredes de dimensões 25x12 centímetros, enquanto que a sustentação da terceira plataforma se dá por três paredes de dimensões 10x12 centímetros.

Fotos do corpo do robô podem ser observadas na Figura 3.2.

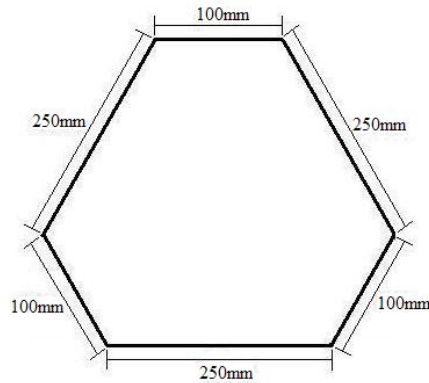


Figura 3.1: Dimensões da base.

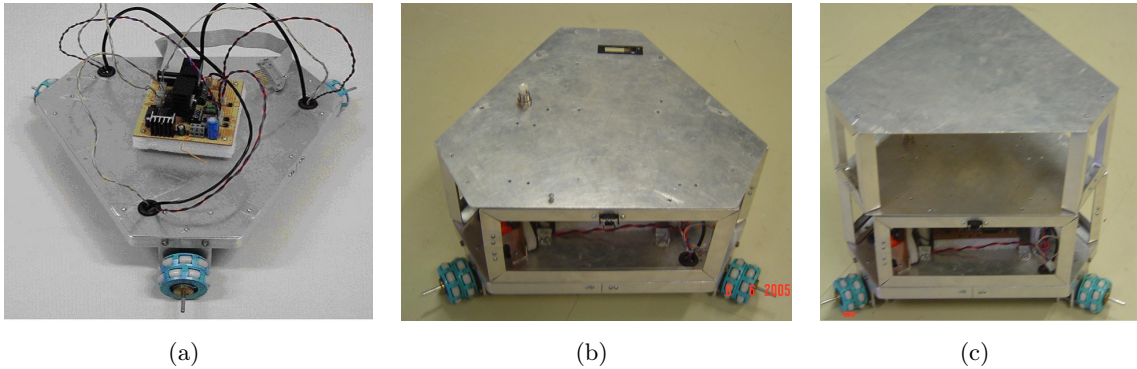


Figura 3.2: (a) Foto do ROHL com plataforma 1, (b) Foto do ROHL com plataformas 1 e 2, (c) Foto do ROHL plataformas 1, 2 e 3.

3.1.2 Caixas de engrenagem

O robô possui três caixas de engrenagem, uma para cada motor, que foram fixadas nos menores lados da base do motor e na plataforma do primeiro andar através de parafusos de 1/8". Cada caixa de engrenagem é composta por duas chapas de alumínio de 3,35 centímetros de espessura e dimensões 60,00 x 50,80 centímetros, que são unidas através de 4 parafusos de 1/8". O motor, que possui 12 dentes, é fixado na caixa de engrenagem em uma das chapas. A engrenagem para redução, que possui 60 dentes, foi fixada na caixa de engrenagem através de um eixo atravessando as duas chapas. Para evitar atrito durante a rotação desse eixo, foram colocados dois rolamentos, um em cada chapa. No meio desse eixo da engrenagem foram colocados um par de rodas. Foi deixado um pedaço de eixo a mais para o caso de, futuramente, colocar um possível sensor para medir a velocidade da roda. Foi colocado, em cada caixa de engrenagem, um sensor de passagem óptico que detecta a passagem de um pino que foi preso à engrenagem do motor. Foto da caixa de engrenagem pode ser vista na Figura 3.3.

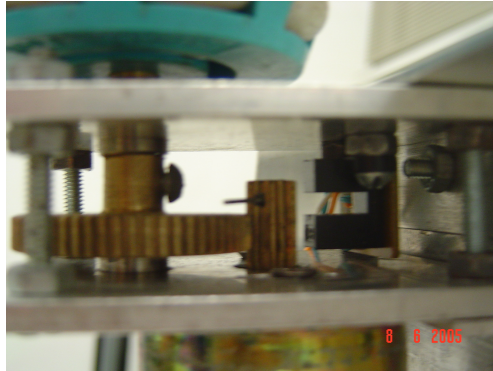


Figura 3.3: Foto da caixa de engrenagem em detalhe.

3.1.3 Rodas

O Robô possui três pares de rodas holonômicas. Cada roda (Figura 3.4 (a)) possui oito roletes que giram livremente no sentido perpendicular ao sentido de giro da roda, o que é necessário para a locomoção omnidirecional, uma vez que as rodas são fixas e não mudam suas direções. Em cada par (Figura 3.4 (b)), as rodas são defasadas de 22.5° , para garantir que sempre haverá contato entre os rolamentos livres e o chão.



(a)



(b)

Figura 3.4: (a) Foto da roda utilizada no robô, (b) Foto do par de rodas.

Cada par de rodas foi fixado em cada lado do hexágono que possui 10 centímetros, estando a um ângulo de 120° umas das outras. Isto foi feito para que as rodas ficassem fixas nas pontas de um triângulo equilátero virtual, o qual pode ser visto na Figura 3.5.

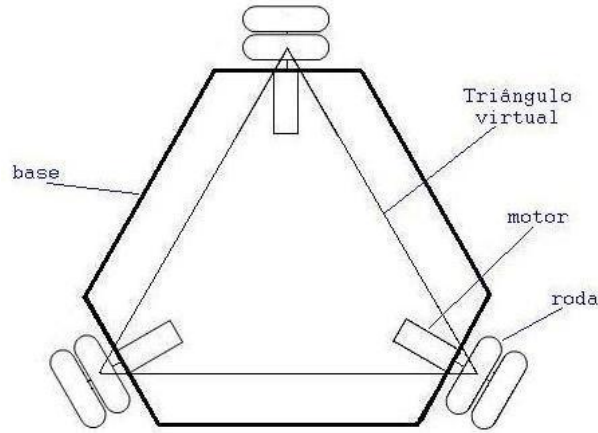


Figura 3.5: Localização das rodas no robô.

3.2 Modelamento Cinemático

Partindo-se da equação 2.1, apresentada em [Campion, Bastin e D'Andréa-Novel 1996], e substituindo os valores da Tabela 3.1 (ver Figura 2.1 (b)) para cada uma das rodas, foi possível obter o modelo cinemático do robô ROHL, de acordo com o referencial da Figura 3.6.

Rodas	α	β	γ	l
1	$\frac{\pi}{2}$	0	0	L
2	$\frac{7\pi}{6}$	0	0	L
3	$\frac{11\pi}{6}$	0	0	L

Tabela 3.1: Tabela de parâmetros das rodas do ROHL

O modelo cinemático obtido é dado por:

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix} = -r \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} -\frac{2}{3} & \frac{1}{3} & \frac{1}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{1}{3L} & \frac{1}{3L} & \frac{1}{3L} \end{pmatrix} \begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix}, \quad (3.1)$$

com

- r : raio das rodas;
- L : distância da roda ao centro do robô (considerado a mesma para todas);
- $x(t)$, $y(t)$: representa a posição, no instante t , do sistema de referência do robô em relação ao sistema de referência externo;

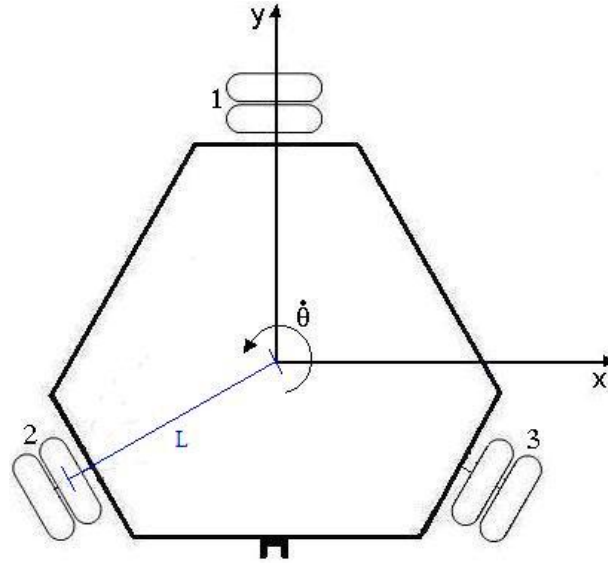


Figura 3.6: Referencial do robô.

- $\theta(t)$: orientação, no instante t , do sistema de referência do robô em relação ao sistema principal;
- $\phi_1(t)$, $\phi_2(t)$ e $\phi_3(t)$: são as velocidades angular das rodas, no instante t .

É importante ressaltar que o modelo cinemático obtido é um modelo simplificado, uma vez que ele foi obtido considerando-se que o ponto de contato das rodas do robô se encontram exatamente nos vértices de um triângulo equilátero. Sendo assim, foram considerados equidistantes em relação ao centro do robô. Outra simplificação do modelo é o fato de que todas as rodas possuem o mesmo diâmetro e que o mesmo é constante para cada uma. Na prática, o diâmetro das rodas não é constante e apresentam variações de uma roda para outra.

O sentido positivo das velocidades das rodas foi considerado como sendo o sentido que faz com que o robô gire no sentido horário, quando todas estiverem girando com mesma velocidade e no mesmo sentido.

A partir da equação 3.1, a cinemática inversa do ROHL pode ser representada por:

$$\begin{pmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \\ \dot{\phi}_3 \end{pmatrix} = -\frac{1}{r} \begin{pmatrix} -1 & 0 & L \\ \frac{1}{2} & -\frac{\sqrt{3}}{2} & L \\ \frac{1}{2} & \frac{\sqrt{3}}{2} & L \end{pmatrix} \begin{pmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{pmatrix}. \quad (3.2)$$

Reescrevendo $\dot{x}$ e $\dot{y}$ em função de v_p e α , em que v_p é o módulo da velocidade de navegação, dado por:

$$v_p = \sqrt{\dot{x}^2 + \dot{y}^2}, \quad (3.3)$$

e α é o ângulo da velocidade de navegação, representado por:

$$\alpha = \arctan\left(\frac{\dot{y}}{\dot{x}}\right). \quad (3.4)$$

Assim têm-se as equações:

$$\dot{x} = v_p \cos \alpha, \quad (3.5)$$

$$\dot{y} = v_p \sin \alpha, \quad (3.6)$$

$$\dot{\theta} = \omega. \quad (3.7)$$

A cinemática inversa pode ser representada por:

$$\dot{\phi}_1 = \frac{1}{r}(v_p \cos \alpha \cos \theta + v_p \sin \alpha \sin \theta - L\omega), \quad (3.8)$$

$$\dot{\phi}_2 = -\frac{1}{2r}(v_p \sqrt{3} \cos \alpha \sin \theta + v_p \cos \alpha \cos \theta - v_p \sqrt{3} \sin \alpha \cos \theta + v_p \sin \alpha \sin \theta + 2L\omega), \quad (3.9)$$

$$\dot{\phi}_3 = -\frac{1}{2r}(-v_p \sqrt{3} \cos \alpha \sin \theta + v_p \cos \alpha \cos \theta + v_p \sqrt{3} \sin \alpha \cos \theta + v_p \sin \alpha \sin \theta + 2L\omega). \quad (3.10)$$

3.3 Alimentação

3.3.1 Circuito da fonte

O circuito da fonte de alimentação (Figura 3.7 (a)) é constituído de um transformador de tensão de 220V/32V, uma ponte de diodos, um resistor de 560Ω/5W, dois capacitores de 4800 μF com tensão 50 V e um fusível de 1 A. O resistor possui a função de descarregar os capacitores quando a fonte for desligada, o fusível possui a função de proteger a fonte de um eventual curto circuito e os capacitores possuem função de diminuir a oscilação do sinal de saída. A ponte de diodos faz a retificação de 32 V de tensão alternada para 43 V de tensão contínua, o qual alimentará o robô através de um conector específico. Este circuito foi montado em uma tábua de madeira e sua saída é conectada ao robô através de um conector específico encontrado na parte de trás do robô. Foto do circuito da fonte pode ser vista na Figura 3.7 (b).

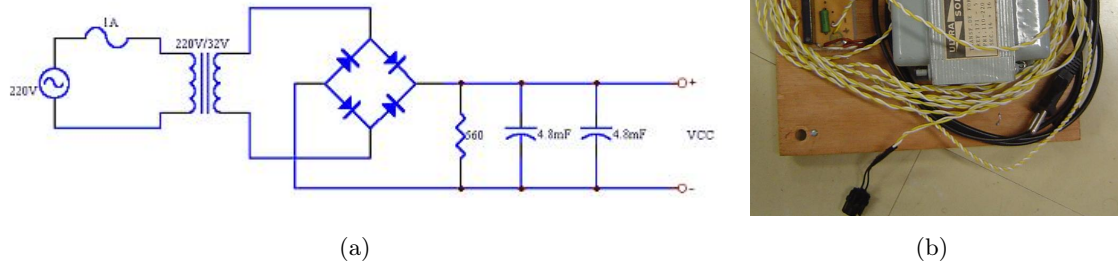


Figura 3.7: (a) Circuito da fonte de alimentação, (b) Foto do circuito da fonte de alimentação.

3.3.2 Motores e Baterias

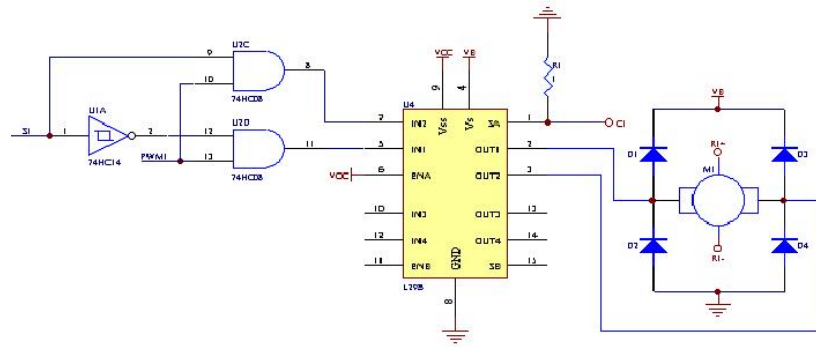
São utilizados motores DC que operam até 24 V com aproximadamente 2500 rpm. Estes motores possuem *resolvers* internos, que são utilizados para a medição de velocidade das rodas. Não se têm muitas especificações destes motores porque são motores retirados de antigas impressoras da Epson (antiga Shinsho Seiki).

São utilizadas duas baterias recarregáveis seladas da DIAMEC em série. Cada bateria possui 12 V, e podem fornecer 7,2 A/h durante 20 horas. As baterias se encontram na primeira plataforma do robô.

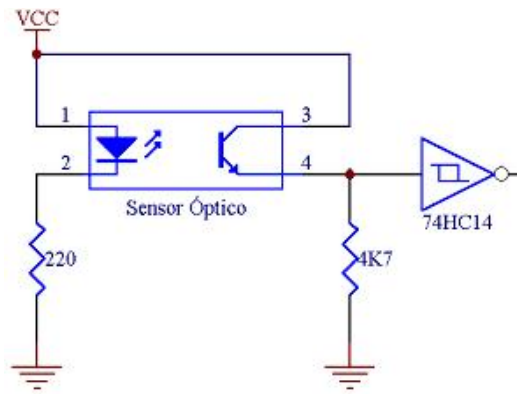
3.3.3 Carregador de baterias

O circuito de carregamento das baterias carrega as baterias quando a alimentação da fonte é conectada ao robô, e este não esteja ligado. Caso o robô esteja ligado, a energia da fonte é utilizada para alimentar o robô. Se a fonte não está conectada ao robô, as baterias se encarregam de alimentar o robô. Este circuito foi cedido pelo professor Geovany A. Borges, do Departamento de Engenharia Elétrica da Universidade de Brasília. O circuito se encontrava em um de seus robôs, chamado "Asterisco". Este circuito se localiza na primeira plataforma do ROHL. O esquemático do circuito de carregamento das baterias pode ser visto na Figura 3.8.

conforme pode ser visto na Figura 3.19, no final deste capítulo. O diagrama completo da placa pode ser visto no Apêndice A.



(a)



(b)

Figura 3.9: (a) Circuito de um sensor óptico, (b) Circuito de acionamento de um motor.

3.5 Processamento com Arquitetura DSP(Processador Digital de Sinais)

3.5.1 Placa do DSP

No início do trabalho, foi utilizado o DSP TMS320C31 da *Texas Instruments* como unidade central de processamento. Como pode ser visto em [Texas Instruments 1997], a unidade de processamento do DSP possui barramento de deslocamento de 32 bits, unidade lógica aritmética, capacidade de fazer contas em ponto flutuante, arquivo de registros. O DSP C31 dispõe de ferramentas de desenvolvimento específicos para a programação em linguagem C, tais como compilador, montador e um programa para enviar o código hexadecimal do computador para o DSP. Além da placa do

DSP, há uma placa filha, que foi construída pelo professor Geovany A. Borges. Essa placa contém alguns dispositivos, tais como porta paralela, porta serial, memória *Flash*, dispositivo sonoro, *leds*, entre outros. A memória *Flash* possibilita o armazenamento de códigos de programa, que podem ser acessadas através de uma chave com dois estados, que são de liberar ou não o acesso à memória *Flash*.

Devido a problemas encontrados, os quais serão explicados mais à frente no documento, o DSP deixou de ser utilizado no robô. Foto da placa do DSP pode ser vista na Figura 3.10. Apesar de não ser utilizada atualmente, esta placa sofreu alterações em seus circuitos. Para fins de documentação, o diagrama do circuito do DSP se encontra no Apêndice A.

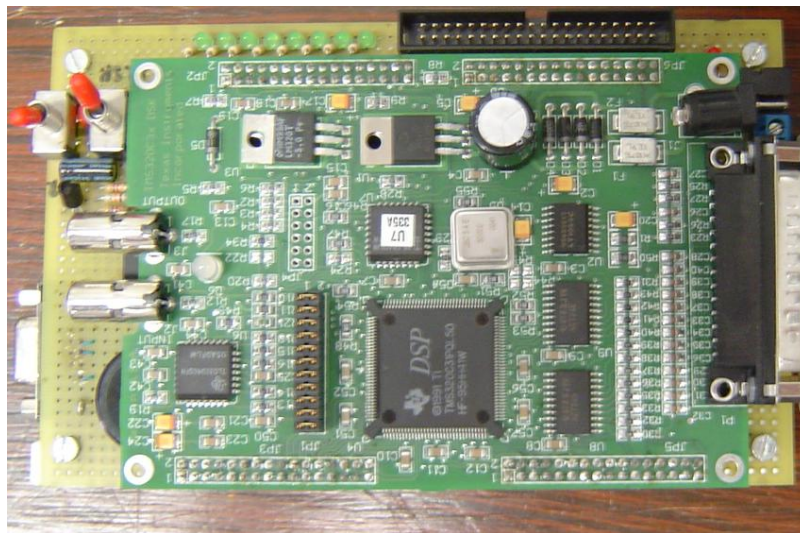


Figura 3.10: Foto do circuito do DSP.

3.5.2 Placa de Geração de MLP (Modulação em Largura de Pulso)

Esta placa possuía a principal função de geração de MLP e envio destes sinais para a placa de potência, mas também tinha a função de fazer o tratamento dos sinais dos *resolvers* dos motores. Esta placa se comunica com a placa de potência através de um conector de vinte pinos e com o DSP através de um conector de quarenta pinos.

O circuito de geração de sinal MLP projetado (Figura 3.11 (a)) é composto por um CI 82C54, um CI 74HC08, um CI 74HC14, um resistor de $620\ \Omega$, um resistor de $220\ K\Omega$ e dois capacitores de $1\ pF$ e um oscilador de $10\ MHz$. Duas portas inversoras foram usadas juntamente com os resistores, os capacitores e o oscilador para gerar um sinal de onda quadrada, o qual será utilizado pelo 82C54 como relógio. Uma outra porta inversora foi utilizada em conjunto com uma porta 'E', de forma que a inversora está sendo utilizada como *buffer* e a porta 'E' é utilizada para que o sinal de MLP gerado pelo 82C54 acione os motores somente quando o sinal *RESET* da placa do DSP

estiver desativado.

O circuito de tratamento de sinal do *resolver* (Figura 3.11 (b)) foi projetado utilizando um CI LM339, um resistor de 10 K Ω , dois de 4,7 K Ω um potenciômetro de 1 K Ω um capacitor de 1 μ F, um de 0,1 μ F e dois diodos. O circuito digitaliza o sinal obtido do *resolver* para poder ser utilizado como uma interrupção no microcontrolador. O circuito consiste basicamente em comparar o sinal do resolver com um sinal de 5 mV, regulado pelo potenciômetro, através do comparador LM339. Este valor foi escolhido para que possíveis oscilações perto da referência não pudessem descaracterizar o sinal. Na comparação, tem-se que o valor de saída é nível lógico 1 quando o sinal do *resolver* é maior do que 5 mV e nível lógico 0 caso contrário. O sinal do resolver e o sinal tratado podem ser visualizados na Figura 3.12;

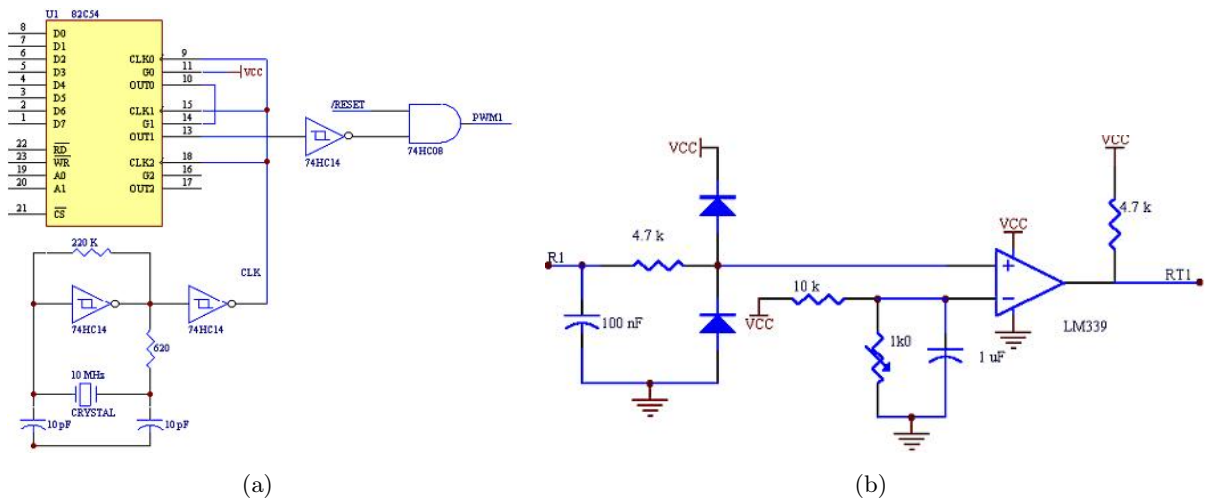


Figura 3.11: (a) Circuito de geração de MLP, (b) Circuito de tratamento do sinal do *resolver*.

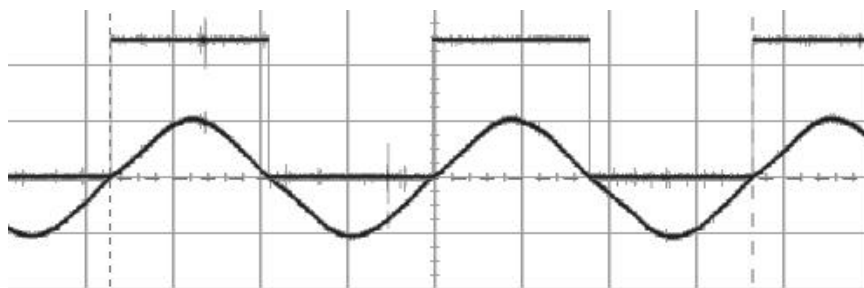


Figura 3.12: Sinais analógico e digital do *resolver*.

Os capacitores são utilizados para minimizar a influência de ruídos. O resistor colocado na saída do comparador é utilizado para limitar a corrente.

Com a desistência da utilização da placa DSP e a decisão de utilizar os microcontroladores AVR, grande parte desta placa deixou de ter função para o nosso robô. Desta forma ela também

não só foi deixada de ser utilizada, como foi destruída para aproveitamento de alguns componentes para a construção da nova placa dos microcontroladores. Esta placa se encontrava na segunda plataforma do robô. O diagrama completo da placa pode ser visto no Apêndice A.

3.6 Processamento com Arquitetura x86

Com a desistência de se utilizar o DSP, foi decidido utilizar a placa Vortex86-6047 para controlar o robô e três microcontroladores AVR para serem servo-controladores dos motores. Devido a falta de tempo para compreensão do funcionamento da placa, esta foi temporariamente substituída por um computador.

3.6.1 Placa Vortex86-6047

Esta placa contém processador 486-DX500⁺, relógio em tempo real, AMI Bios, 128 MB SDRAM, três portas RS232, uma porta RS232/485, uma porta paralela, duas portas USB, placa de vídeo, placa de rede, etc. Nesta placa será instalado o Windows NT com o RTX para fazer o controle do robô em tempo real.

Esta placa se conectará com a placa dos AVRs através do conector serial RS485. Foto da placa Vortex86-6047 pode ser vista na Figura 3.13.

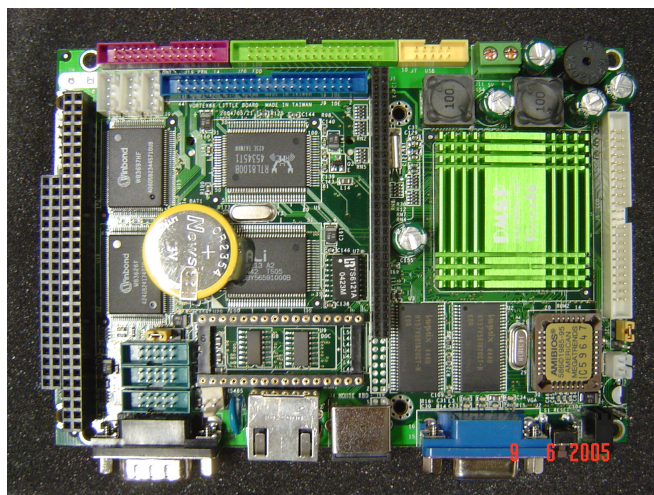


Figura 3.13: Foto da placa Vortex86-6047.

3.6.2 Placa dos microcontroladores

Esta placa tem as mesmas funções da placa de geração de MLP, com a diferença de que agora estão sendo utilizados microcontroladores ao invés de CI's para gerar estes sinais. Esta placa se comunica com a placa de potência através de um conector de 20 pinos e com o computador através da porta serial.

O circuito dos AVR's possui três microcontroladores AVR ATmega8, juntamente com o circuito de gravação desses AVR's e circuito de comunicação através da RS485. Os AVR's são controlados pelo computador ou pela placa Vortex86-6047 pelo método mestre-escravo, onde cada AVR tem a função de gerar PWM e coletar dados de velocidade de cada motor. A comunicação é feita a uma taxa de 115200 bps e são utilizados cristais de 16 MHz como *clock* externo nos AVR's. Foi colocada uma chave para modificar o estado do *RESET* nos AVR's, afim de poder ligá-los ou desligá-los. Esta chave pode ser vista na Figura 3.14.

O circuito de comunicação consiste basicamente na utilização de três CIs DS485, um para cada AVR. Este CI transforma os sinais A e B que vêm do computador para um sinal único que entra no AVR ou o inverso. Nesta placa, no sinal A foi colocado um resistor de 470 Ω para 5 V e no sinal B o mesmo resistor para 0 V. Isto para garantir que não fiquem em alta impedância quando nada estiver sendo enviado. Também foi colocado um resistor de 120 Ω entre A e B para que a diferença na tensão entre eles seja bastante pequena quando não houver nada sendo transmitido. No sinal de recepção de dados no AVR foi colocado um resistor de 4,7 K Ω para 5 V e no sinal de habilitação de transmissão foi colocado um resistor de 10 K Ω para 0 V, também para garantir que não fiquem em alta impedância quando não houver nada sendo transmitido.

Nos sinais de PWM e de sentido, quem podem ser vistos na Figura 3.14, foram colocados diodos de proteção para garantir que o sinal não será menor do 0 V e nem maior do que 5 V. Nos sinais de PWM também foram colocados resistores de 4,7 K Ω para 0 V, a fim de garantir que os motores ficarão parados quando nenhum sinal de PWM tiver sido gerado.

O circuito de tratamento de sinal do *resolver* foi retirado da placa de geração de MLP e seu funcionamento foi explicado na seção 3.5.2. O circuito de tratamento dos *resolvers* foi calibrado para que o sinal da saída fique em nível lógico 1 se o sinal do *resolver* for maior do que 30 mV, tensão observada através de experimentos em que a roda está girando continuamente sem falhas a uma baixa velocidade. Esta placa se encontra na segunda plataforma do robô. Foto desta placa pode ser vista na Figura 3.14 e seu diagrama completo se encontra no Apêndice A.

3.6.3 Gravador de AVR's

O circuito de gravação dos AVR's consiste basicamente em receber os dados (RESET, MISO, MOSI, SCK, VCC, GND) enviados do computador através da porta paralela e repassar para os pinos adequados do AVR. Para isto foi montada uma gravadora que consiste basicamente em um

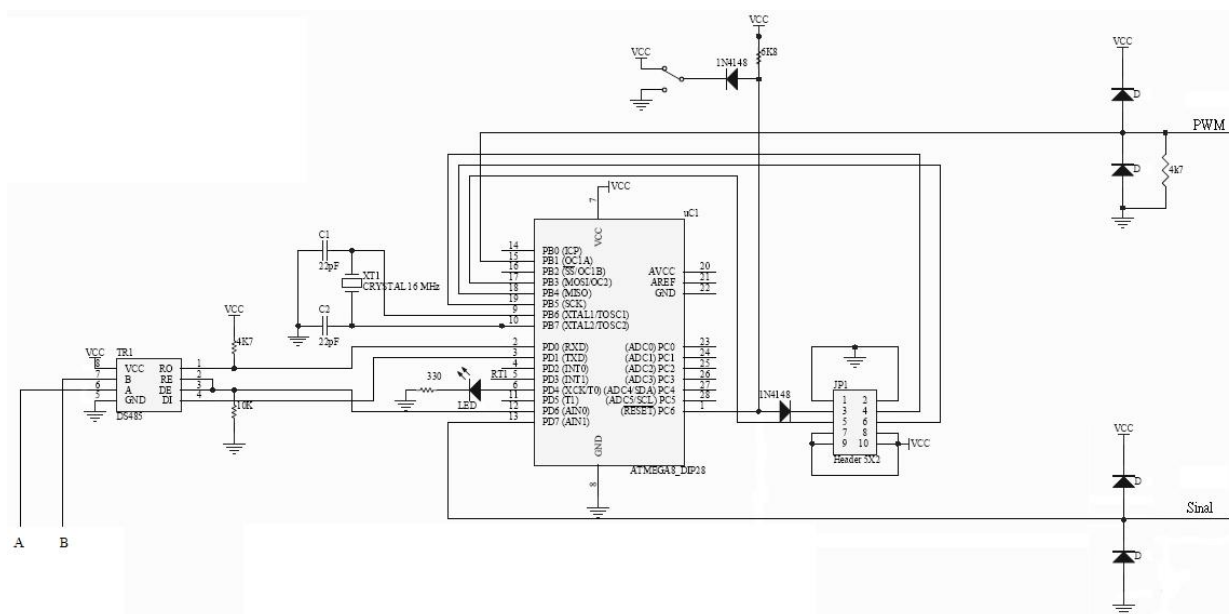


Figura 3.14: Circuito de um AVR.

buffer para os dados, mas colocando resistores nestes sinais indo para VCC, garantindo que estes sinais não ficarão em alta impedância quando não houver nenhum dado sendo passado. Há também LEDs para indicar a situação da gravação, que pode ser de ERRO, VERIFICANDO, GRAVANDO e OK.

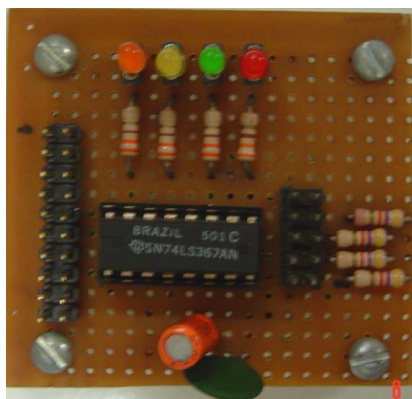
Esta placa não se encontra anexada ao robô. Foto e esquemático deste circuito podem ser vistos na Figura 3.15.

3.6.4 Placa de conversão RS232/RS485

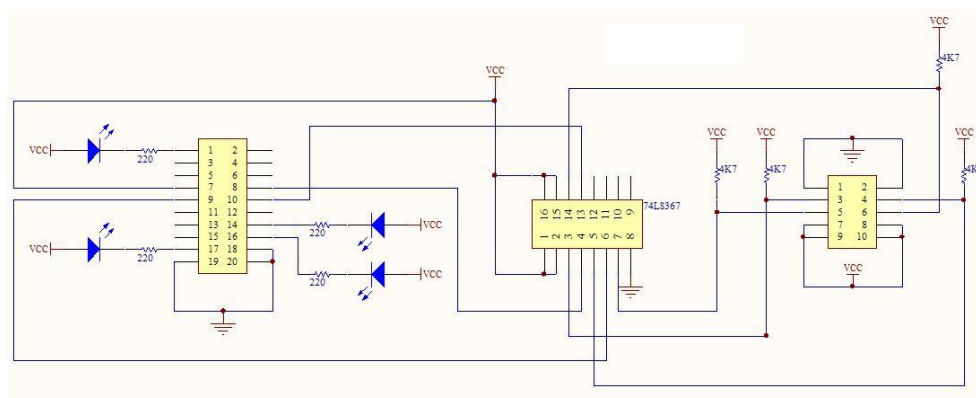
Foi construída uma placa de comunicação para converter dados do padrão RS232 para RS485 ou vice-versa, já que o computador só possui comunicação RS232. Esta placa só será utilizada até que a placa Vortex86-6047 seja inserida no robô, uma vez que esta placa já possui comunicação RS485. O projeto foi feito pelo mestrando Alexandre S. Martins, do Departamento de Engenharia Elétrica da Universidade de Brasília.

Esta placa pode ser alimentada tanto por 12 V, quanto por 5 V, sendo que a escolha é feita apenas com a mudança de um *jumper*. A placa se comunica com o computador através de um conector serial fêmea de nove pinos e se comunica com a placa dos microcontroladores através de conectores de dois pinos, que serão interligados por um cabo blindado.

O principal componente possuído na placa é o DS485, cujo funcionamento já foram explicados nas Seções 2.5 e 3.6.2. Foto do circuito pode ser visualizada na Figura 3.16.



(a)



(b)

Figura 3.15: (a) Foto do gravador de AVR, (b) Esquemático do gravador de AVR.

3.7 Desenvolvimento de *Software*

3.7.1 Módulo DSP

Utilizando-se o microcontrolador DSP, os códigos das funções de atuação no robô foram divididos em duas partes: o C31sbc e o C31rtk.

O arquivo C31sbc.c31 possui algumas funções necessárias para o controle do robô. O professor Geovany A. Borges já havia feito algumas funções, tais como de espera, da ativação sonora e tratamento de interrupções. O que se acrescentou a este arquivo foi a comunicação através da porta serial, a geração do sinal MLP, o tratamento de interrupções através do CI 82C59.

O arquivo C31rtk.c31 possui as funções para o controle do robô em tempo real. Este arquivo, também cedido pelo professor Geovany A. Borges, possui funções de criação de tarefas e funções para adquirir o tempo atual. No momento, o tratamento de interrupções não é pré-emptivo.

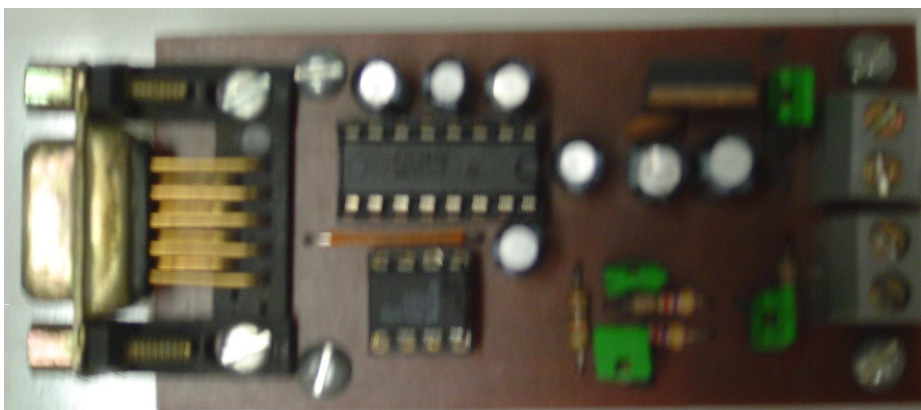


Figura 3.16: Foto da placa de conversão RS232/RS485.

Foram desenvolvidos também vários arquivos de testes para controle do robô, tais como de rotação, de gerar a trajetória de um hexágono, obedecer a comandos da porta serial e aquisição dos parâmetros do sistema.

Durante a aquisição de dados do sistema, foram encontrados problemas na interrupção, o que atrapalhava o funcionamento do DSP. Chegou-se a pensar que o problema estava sendo causado por interferências eletro-magnéticas. Então foram feitas trocas dos fios por fios blindados, mas nada adiantou. Observou-se posteriormente que o problema estava sendo causado pela programação em tempo real. Para solucionar este problema, tentou-se adaptar uma versão do Ucos, executivo de tempo real desenvolvido para outro tipo de DSP, entretanto essa solução tornou-se inviável devido ao tempo necessário para implementá-la. Tentou-se então transformar o sistema de tempo real em um sistema pré-emptivo, mas devido a complexidade encontrada e a falta de tempo, foi decidido que este DSP não seria mais utilizado.

Os programas desenvolvidos para o módulo DSP podem ser visualizados no CD em anexo e a descrição dos arquivos deste CD é apresentada no Apêndice B.

3.7.2 Módulo PC com AVR

Neste módulo foi realizada uma comunicação do tipo servidor-escravo, em que o computador é o servidor e os AVR são os escravos, ou seja, os AVR só comunicam se o computador lhes deu uma ordem para comunicar. Figura demonstrativa dessa comunicação pode ser vista na Figura 3.17.

Na programação de cada AVR há funções para ficar esperando comandos para cada AVR e assim poderem realizar funções como gerar sinal MLP, calcular velocidade do motor e enviar esta velocidade para o computador.

Na programação do computador, os códigos das funções foram divididos em duas partes: Rohl_Lib e Rohl_RTX. O arquivo Rohl_Lib.c contém as funções de enviar comandos para al-

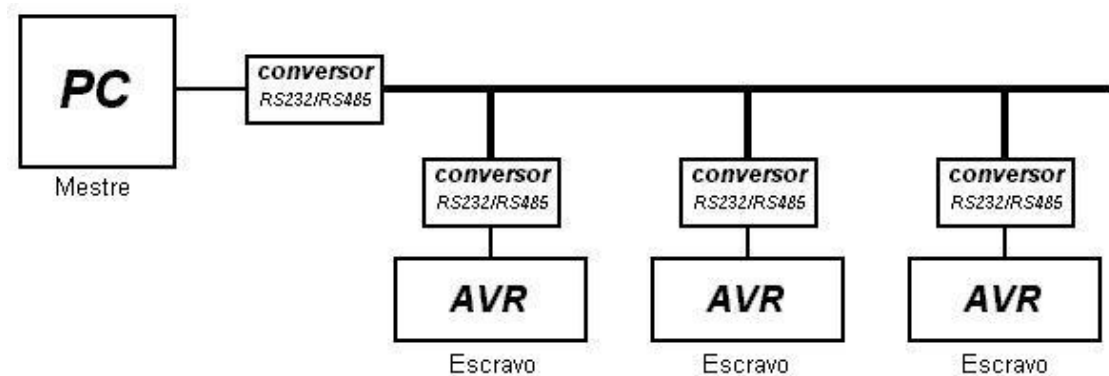


Figura 3.17: Comunicação PC com AVR.

um AVR para geração de sinal MLP, cálculo de velocidade e envio dessa velocidade. O arquivo Rohl_RTX.c é responsável por gerar uma tarefa que faça o controle do robô a cada intervalo de tempo.

Para aquisição de dados para fazer a caracterização dos motores, a tarefa do programa Rohl_RTX.c gerava um sinal MLP randômico a cada 2 segundos e coletava dados de velocidade a cada 10 ms, sendo que foram adquiridos 4000 dados de velocidades de cada roda. Há também dois arquivos que foram cedidos pelo professor Geovany A. Borges: SerialLibRTX.c e GMatrix.h. Estes arquivos, respectivamente possuem funções para comunicação serial e para manipulação de matrizes para aquisição de dados. Com os dados de modulações em largura de pulso, velocidades e tempos, utilizando a função `arx` do Matlab[®] foi realizada a identificação do sistema.

Com a identificação dos motores realizada, foi feito um controle proporcional integrativo (PI) na realização da tarefa no arquivo Rohl_RTX para fazer o controle das velocidades dos motores. Os parâmetros deste controlador foram ajustados heurísticamente, através da avaliação de resultados encontrados na execução deste controlador. Foram feitos programas no Matlab[®] para visualização gráfica dos dados coletados do robô.

Os arquivos relacionados a este módulo podem ser visualizados no CD em anexo e a descrição dos arquivos deste CD é apresentada no Apêndice B.

3.8 Protocolo de comunicação

Foi feito um protocolo de comunicação entre o computador e os microcontroladores do tipo mestre-escravo, sendo que os microcontroladores são os escravos, ou seja, só respondem ao que lhes forem solicitados pelo computador.

Há dois tipos de bytes a serem transmitidos: os bytes de controle e os bytes de dados. A

diferença entre os bytes de controle e de dados está no bit 7, que será nível lógico 1 para byte de controle e nível lógico 0 para byte de dado. Dentre os bytes de controle, há diferença entre os bytes de controle que o computador envia e os que o AVR envia.

O byte de controle que o PC envia inclui o sentido de rotação do motor, a operação e o endereço do AVR. O sentido de rotação só é válido para a operação Set Velocidade(Estipular um PWM para o motor desejado), sendo que o próximo byte a ser enviado será um caractere correspondente ao valor do sinal MLP, que é um valor entre 0 e 100. No caso das operações Calcular Velocidade(Comando para AVR calcular velocidade) e Get Velocidade(Comando para AVR enviar a velocidade do motor) é enviado apenas o byte de controle correspondente.

O byte de controle enviado do AVR para o CPU contém apenas a quantidade de dados e o endereço do PC, sendo que em seguida são enviados os bytes correspondentes ao dado. O único caso em que isto ocorre é na transmissão da velocidade do motor para o computador, quando lhe é pedido. Nesta transmissão da velocidade, cada algarismo é enviado em forma de um caractere em separado.

Os tipos de bytes de transmissão, juntamente com a explicação de cada bit desses bytes, podem ser observados na Figura 3.18.

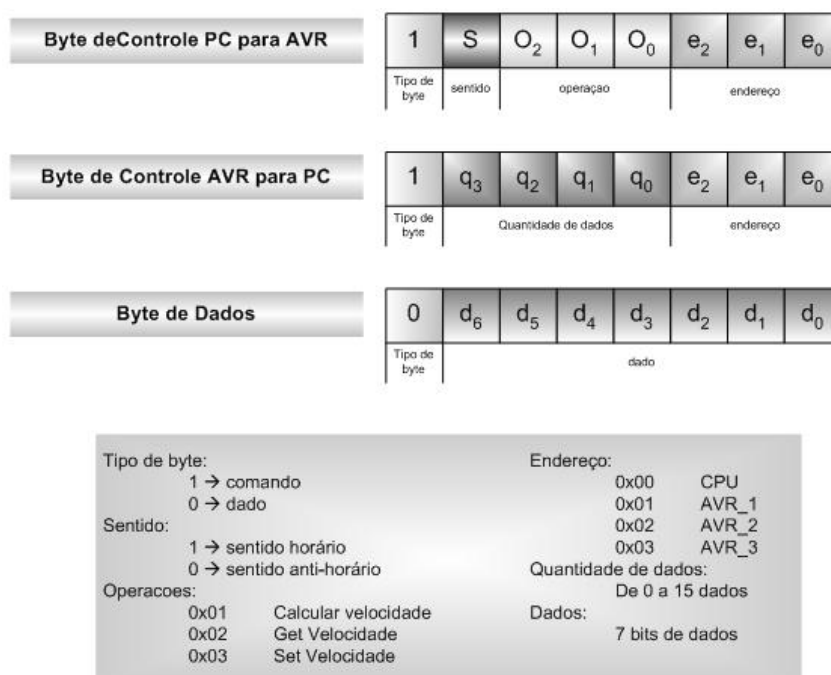
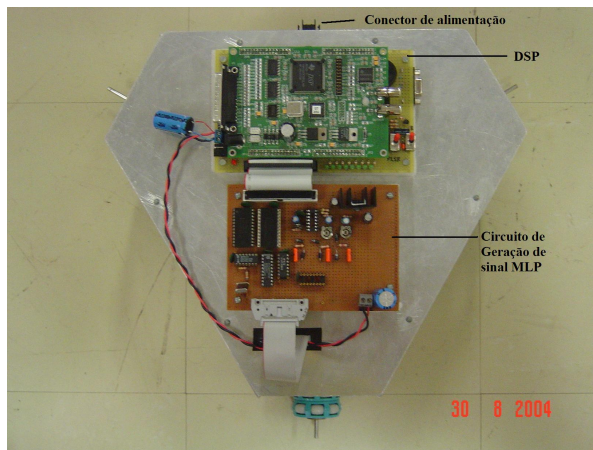


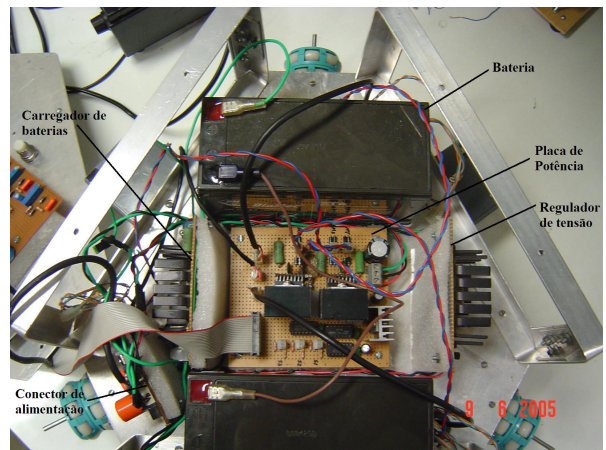
Figura 3.18: Protocolo de comunicação.

3.9 Fotos do ROHL

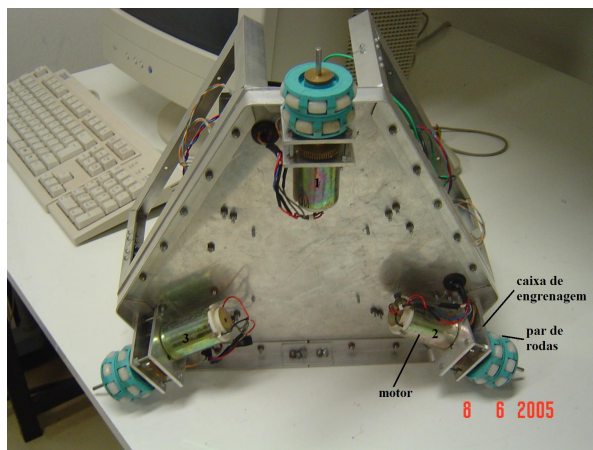
Nesta seção foram colocadas algumas fotos do ROHL para mostrar o resultado final. Na Figura 3.19 (a) tem-se a vista do ROHL por cima quando ainda eram utilizados o DSP e a placa de geração de sinal MLP. Os componentes encontrados na primeira plataforma do robô podem ser observados na Figura 3.19 (b). Na Figura 3.19 (c) tem-se a vista inferior do ROHL, em que podem ser vistos o acoplamento das caixas de engrenagem, rodas e motores na base do robô. Nesta foto, pode-se ver a numeração dos motores conforme foi determinamos. Foto atual do robô, com a placa de microcontroladores, pode ser vista na Figura 3.19 (d). A placa vortex86-6047 será colocada no espaço encontrado na segunda plataforma.



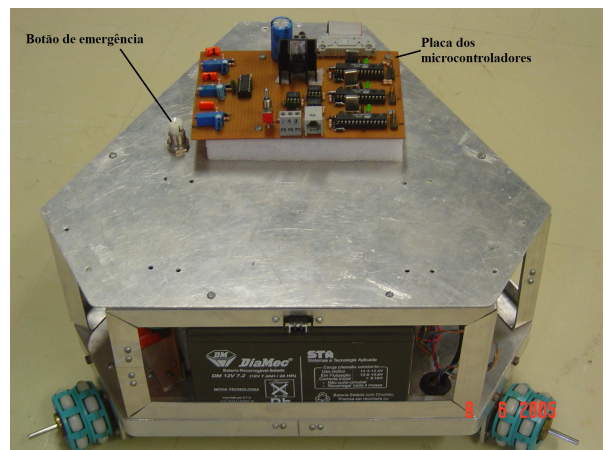
(a)



(b)



(c)



(d)

Figura 3.19: (a) Foto do ROHL com DSP, (b) Foto do ROHL visto por baixo, (c) Foto dos componentes da primeira plataforma, (b) Foto do ROHL com AVR.

Experimentos

Os circuitos eletrônicos construídos para o robô foram previamente testados, para verificar se estavam funcionando de forma correta. Somente após a verificação de que os mesmos funcionavam de forma satisfatória e atendiam os requisitos para os quais foram projetados é que eles foram interligados, possibilitando a realização dos testes com o robô.

Quando o microcontrolador DSP foi utilizado, verificou-se que o mesmo gerou de forma satisfatória os sinais MLP nos CI's 8254. Com isso, foi possível obter o ciclo de trabalho desejado para acionar os motores. O ciclo de trabalho desejado pode ser comprovado com o auxílio do osciloscópio.

A capacidade de gerar sinal MLP, com ciclo de trabalho confiável, permitiu ao DSP executar um programa que fazia com que o robô percorresse uma trajetória hexagonal, sem mudar sua orientação. Este programa foi realizado de forma a acionar um par de rodas do robô, de forma intercalada e com ciclos de trabalho constantes, fazendo com que o robô realizasse uma trajetória próxima à desejada. Para este caso, a trajetória foi obtida com aplicação de técnicas de controle em malha aberta.

Com a execução deste programa, foi possível comprovar a omnidirecionalidade do robô, apesar de sua orientação não ter se mantido constante durante todo o percurso. Este fato se deve basicamente ao escorregamento das rodas, à simplificação do modelo cinemático e à diferença entre as características de cada motor.

As interrupções do CI 8259 e o sinal proveniente dos *resolvers* foram testados com a conexão destes sinais, obtidos após o circuito de tratamento dos mesmos, às entradas de interrupção do CI 8259 e com a conexão do pino de solicitação de interrupção do mesmo a uma entrada de interrupção externa do DSP. Com isso, foi executado, no DSP, um programa que interpretava a solicitação de cada interrupção e de acordo com a interrupção solicitada, enviava um caractere pela serial, indicando qual interrupção estava sendo tratada. Com este teste foi possível comprovar o funcionamento das solicitações e do tratamento das interrupções provenientes do CI 8259, através dos sinais dos *resolvers*.

Um programa foi desenvolvido para que o DSP recebesse, de um computador portátil instalado sobre o ROHL, caracteres pela porta serial. De acordo com o caractere recebido, o ROHL executaria um movimento para frente, para trás, para direita ou para esquerda. A movimentação do robô foi feita com o mesmo acionamento, dos motores, utilizado no teste para desenvolver trajetória hexagonal. O computador portátil foi programado, por outros alunos, para receber dados de um celular via *bluetooth*. Os dados foram corretamente interpretados pelo DSP, o que possibilitou ao ROHL executar, de forma satisfatória, as trajetórias solicitadas.

Um programa com duas tarefas em tempo real foi desenvolvido para ser executado no DSP. Uma delas seria responsável por calcular e armazenar a velocidade das rodas do ROHL, obtidas por interrupções solicitadas a partir dos sinais dos *resolvers*. A outra deveria gerar ciclos de trabalhos aleatórios para os motores. Após um determinado tempo de execução o programa deveria enviar alguns dados pela porta serial. O objetivo deste programa seria fazer a identificação do modelo de cada um dos motores com auxílio da função "arx" contida na *toolbox* do MatLab[®]. Entretanto, os resultados desejados não foram obtidos devido a problemas com o gerenciamento, por parte do DSP, das interrupções externas, proveniente dos sinais dos sensores, e das interrupções internas, provenientes das tarefas de tempo real.

A alternativa encontrada e que gerou resultados satisfatórios, foi a substituição do DSP por um sistema de tempo real para o PC, o RTX. Com essa substituição, o sistema de acionamento e de leitura de velocidade das rodas passou a ser feito por microcontroladores AVR ATmega8.

Cada roda possui um AVR responsável por seu acionamento e medida de velocidade. Os testes realizados com um programa para comunicação serial do PC, o programa Terminal, mostrou a eficiência dos AVRs e do protocolo de comunicação implementado, uma vez que os AVRs geraram os sinais MLP e mediram a velocidade de forma satisfatória. Neste teste, os AVRs foram programados para receber, via comunicação serial, um ciclo de trabalho para o sinal MLP e para enviar o valor da velocidade medida. Cada tarefa a ser executada deveria ser solicitada pelo computador.

O ciclo de trabalho gerado pelos AVRs pode ser comprovado com o auxílio do osciloscópio. A medição de velocidade também foi verificada com auxílio do osciloscópio, convertendo-se a medida da frequência dos sinais gerados pelos *resolvers* em velocidade e comparado o resultado obtido com o valor medido pelos AVRs.

Com a validação do funcionamento da geração de sinais MLP e de medição de velocidade

pelo AVR, como também do sistema de comunicação, foi possível desenvolver um programa na plataforma de tempo real, RTX, para gerar sinais de MLP aleatórios para cada uma das rodas e obter as velocidades das mesmas, com o objetivo de coletar dados para identificação do modelo dos motores utilizados no ROHL. É importante ressaltar que os dados foram obtidos com o robô suspenso, ou seja, as rodas não estavam em contato com o chão.

A identificação do modelo foi feita através dos motores, entretanto, as velocidades foram calculadas com relação às rodas. Com as velocidades e os sinais MLP das mesmas, utilizando a função "arx" da *toolbox* do Matlab[®], foi possível obter as funções de transferência para as rodas, representadas pelas equações 4.1, 4.2 e 4.3, em que v_i é a velocidade na i -ésima roda e u_i é a modulação em largura de pulso na i -ésima roda. As Figuras 4.1 (a), 4.2 (a) e 4.3 (a) representam as curvas obtidas com a identificação. Nestas figuras podemos observar que as curvas obtidas com as funções de transferência que representa o modelo de cada uma das rodas se aproxima bastante da curva com as velocidade medidas para cada uma das rodas, comprovando a linearidade do sistema. Para validação destes modelos, foi feito um arquivo ".m" no Matlab[®]. Este arquivo substitui os valores de MLP nas funções de transferências encontradas (4.1, 4.2 e 4.3) para encontrar os valores de velocidade dos motores. Essas velocidades encontradas podem ser vistas nas Figuras 4.1 (b), 4.2 (b) e 4.3 (b).

$$\frac{v_1(t)}{u_1(t)} = \frac{0,9006}{1 - 0,2861 z^{-1}}, \quad (4.1)$$

$$\frac{v_2(t)}{u_2(t)} = \frac{1,034}{1 - 0,178 z^{-1}}, \quad (4.2)$$

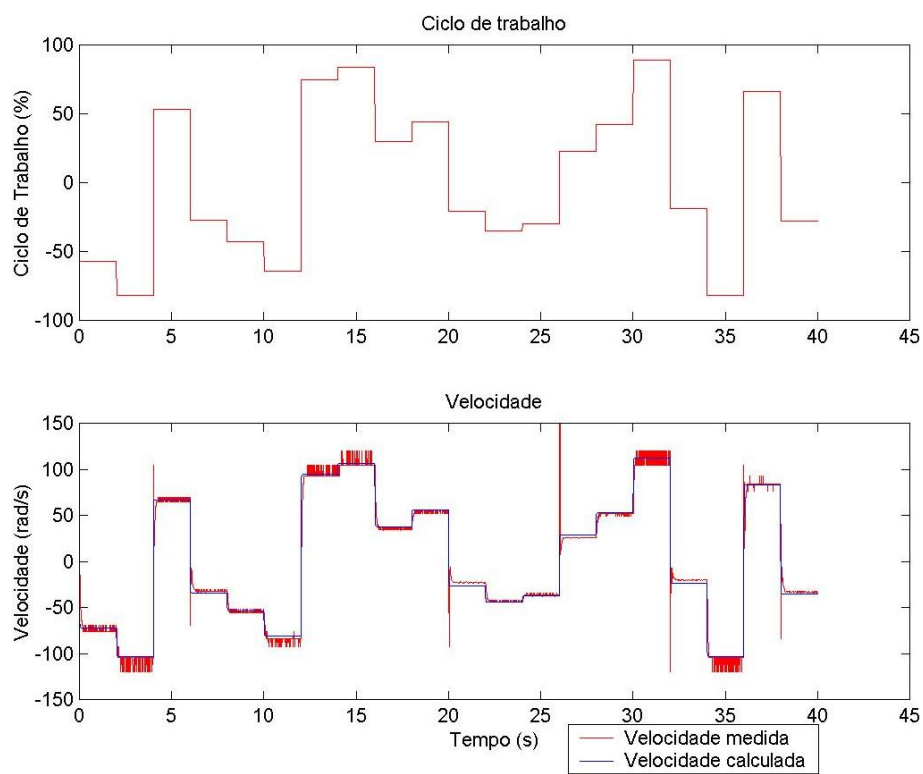
$$\frac{v_3(t)}{u_3(t)} = \frac{1,109}{1 - 0,09289 z^{-1}}. \quad (4.3)$$

Com a comprovação da linearidade do sistema e com a obtenção das funções de transferência das rodas, foi possível projetar um controlador PI para controlar a velocidade das rodas. O diagrama de blocos desse controle pode ser visualizado na Figura 4.4.

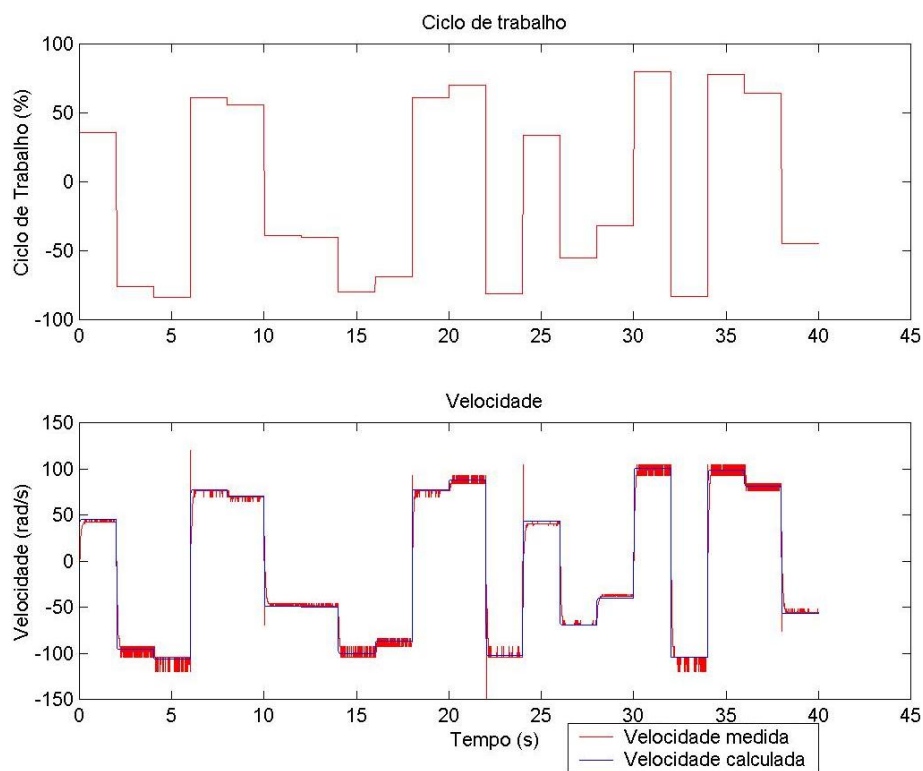
No arquivo Rohl_RTX.c foi desenvolvida uma tarefa para implementar esse controle PI, com taxa de amostragem de 10 ms. Para verificar o funcionamento do controlador, foram geradas velocidades de referências randômicas a cada 2 segundos. Através da execução deste programa, heurísticamente chegou-se aos parâmetros, que minimizaram o tempo de estabilização e o sobrepasso, para o controlador PI de cada motor. Para o motor 1, o melhor K_i foi de 12,1, enquanto que para o motor 2 foi de 12,0 e para o motor 3 foi de 10,5. Os valores de K_p para todos os motores foi de 2,01. Como as leituras de velocidade oscilavam muito, foi implementado um filtro passa-baixas para eliminar os componentes de alta frequência. Nas Figuras 4.5 e 4.6 têm-se os resultados para uma frequência de corte de 1 Hz e de 1,5 Hz respectivamente. Pode ser observado que os resultados se encontram melhores para uma frequência de corte de 1,5 Hz.

Esse controle foi implementado com o robô em suspensão. Quando o mesmo foi colocado no solo, o resultado obtido não foi o mesmo, isso se deve ao fato de que o controle implementado não considera o acoplamento entre as rodas e também ao deslizamento de cada uma das rodas.

Na tentativa de minimizar essas imperfeições, foi projetado uma trajetória hexagonal para o robô. O intuito era de fazer com que sempre estivessem dois motores movimentando o robô em linha reta, evitando deslizamento. Por causa do posicionamento das rodas do robô, a melhor trajetória encontrada foi a de um hexágono. O resultado obtido é mostrado nas Figura 4.7. Com essa figura, pode-se observar que o robô conseguiu fazer o controle de velocidade de forma satisfatória, entretanto, alguns pontos discrepantes que podem ser observados na figura são devido ao deslizamento das rodas. O deslizamento fez com que a trajetória realizada pelo robô não fosse a trajetória desejada, ou seja, um hexágono.

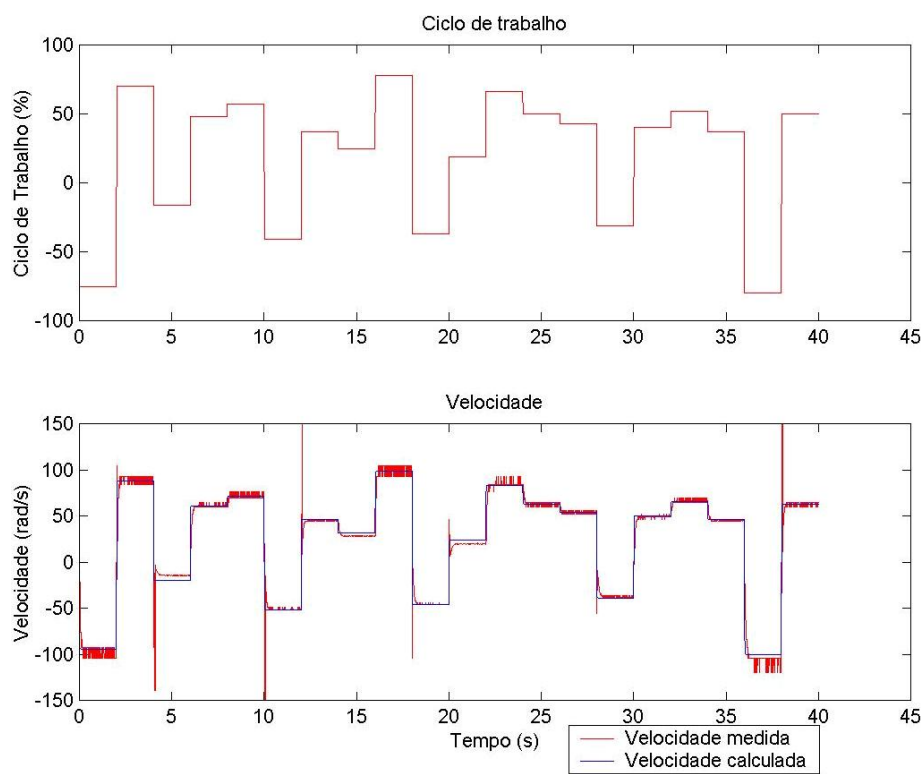


(a)

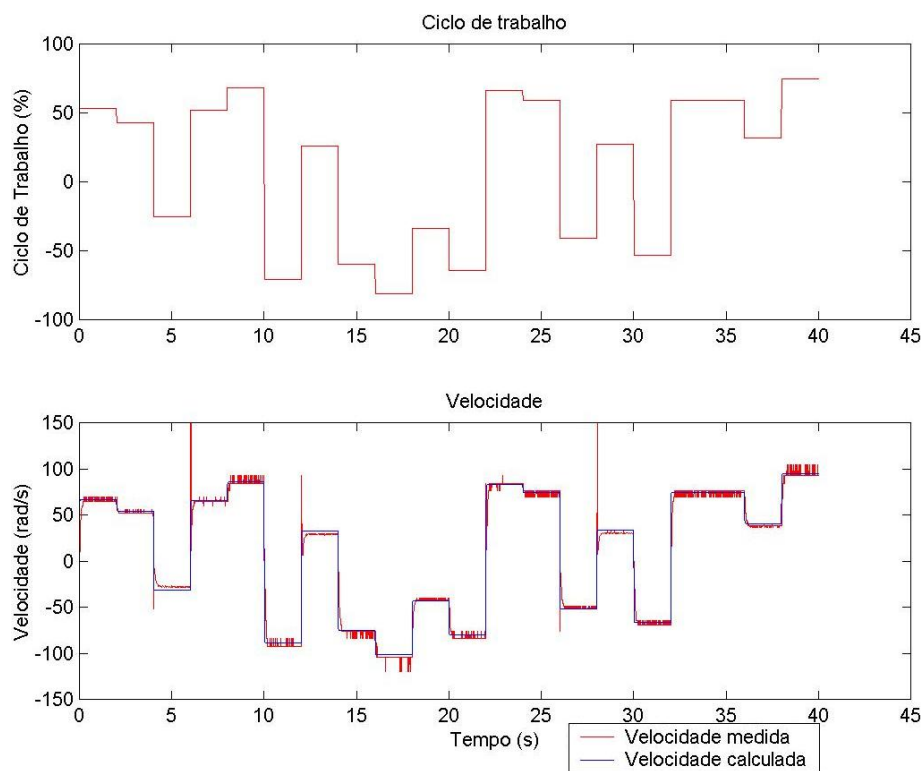


(b)

Figura 4.1: (a) Identificação do motor 1, (b) Validação do motor 1.

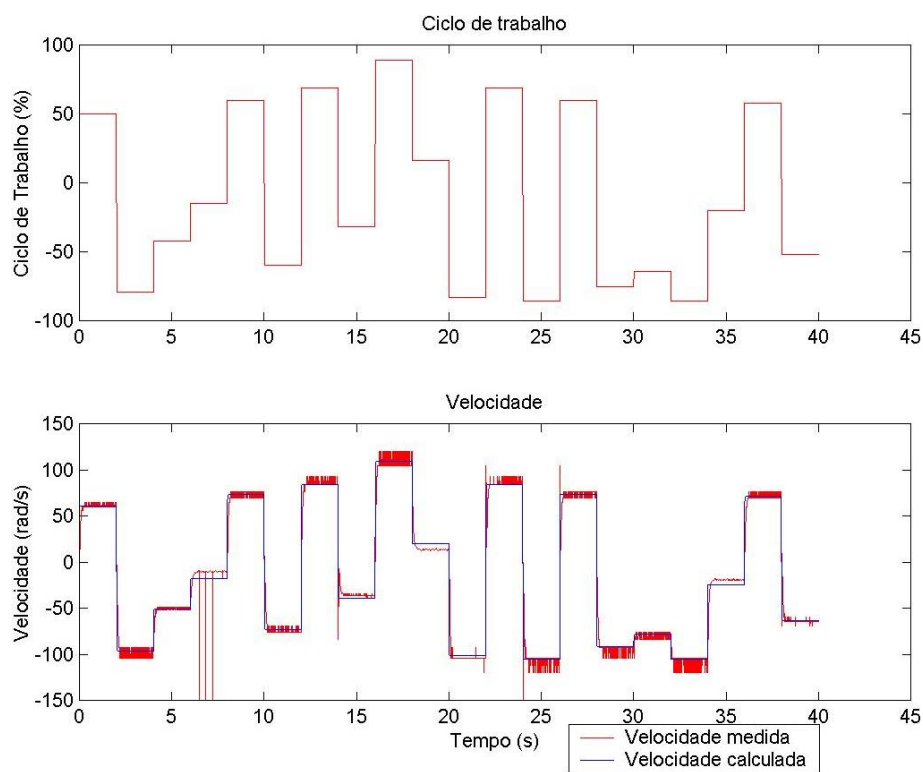


(a)

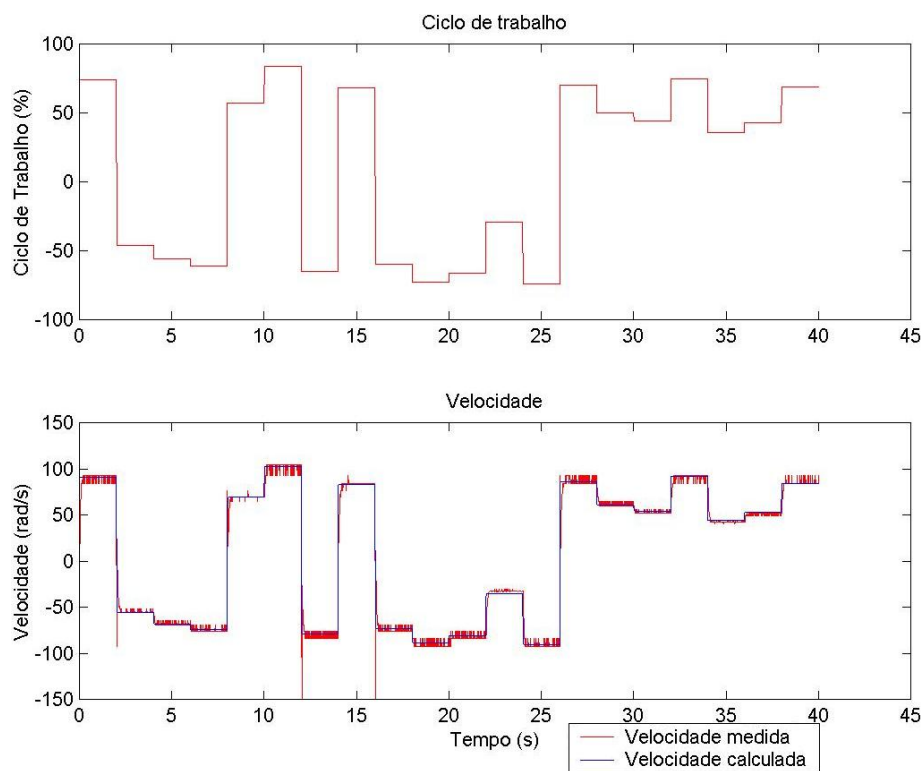


(b)

Figura 4.2: (a) Identificação do motor 2, (b) Validação do motor 2.



(a)



(b)

Figura 4.3: (a) Identificação do motor 3, (b) Validação do motor 3.

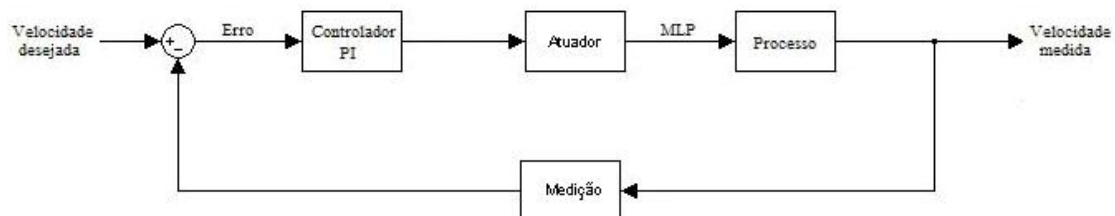


Figura 4.4: Diagrama de blocos do controle.

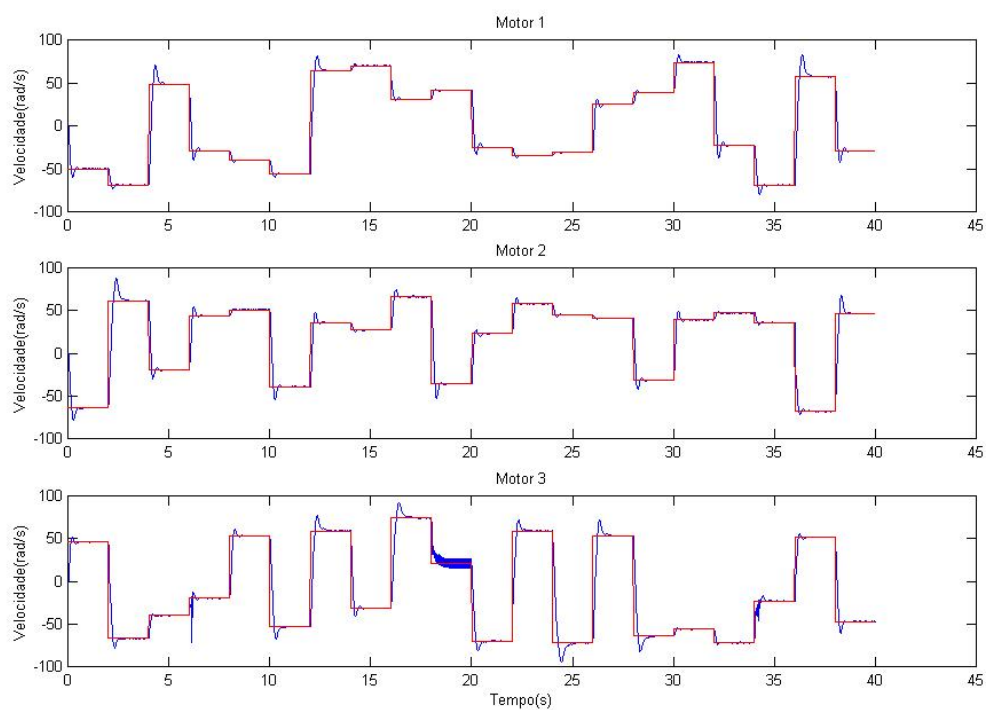


Figura 4.5: Resposta controle PI para frequência de corte 1,0 Hz do filtro de medida de velocidade.

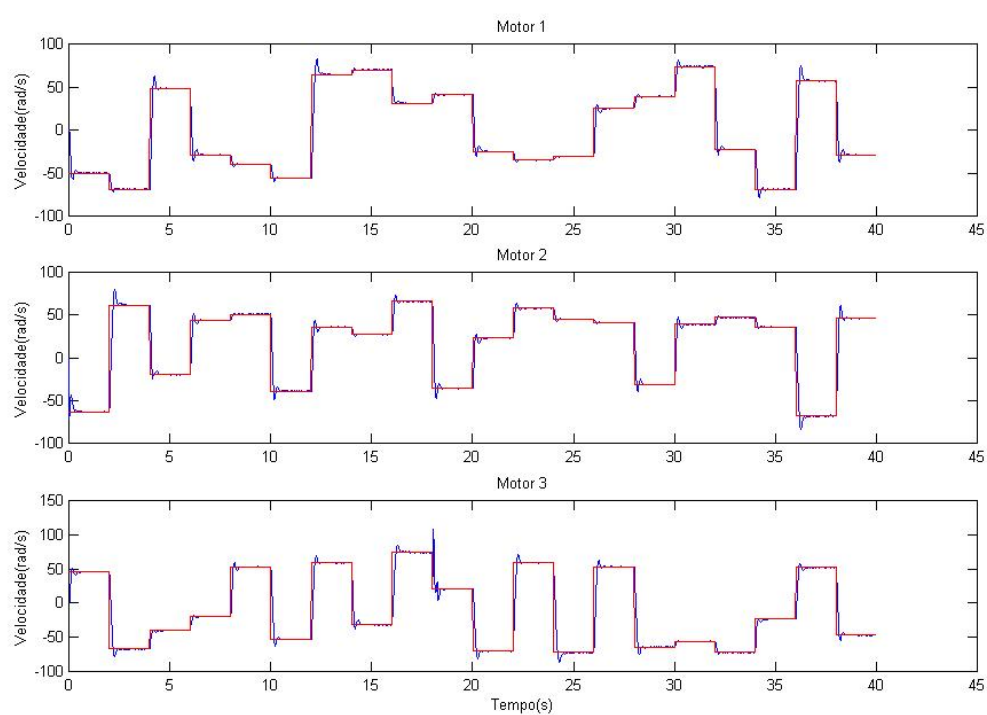


Figura 4.6: Resposta controle PI para frequência de corte 1,5 Hz do filtro de medida de velocidade.

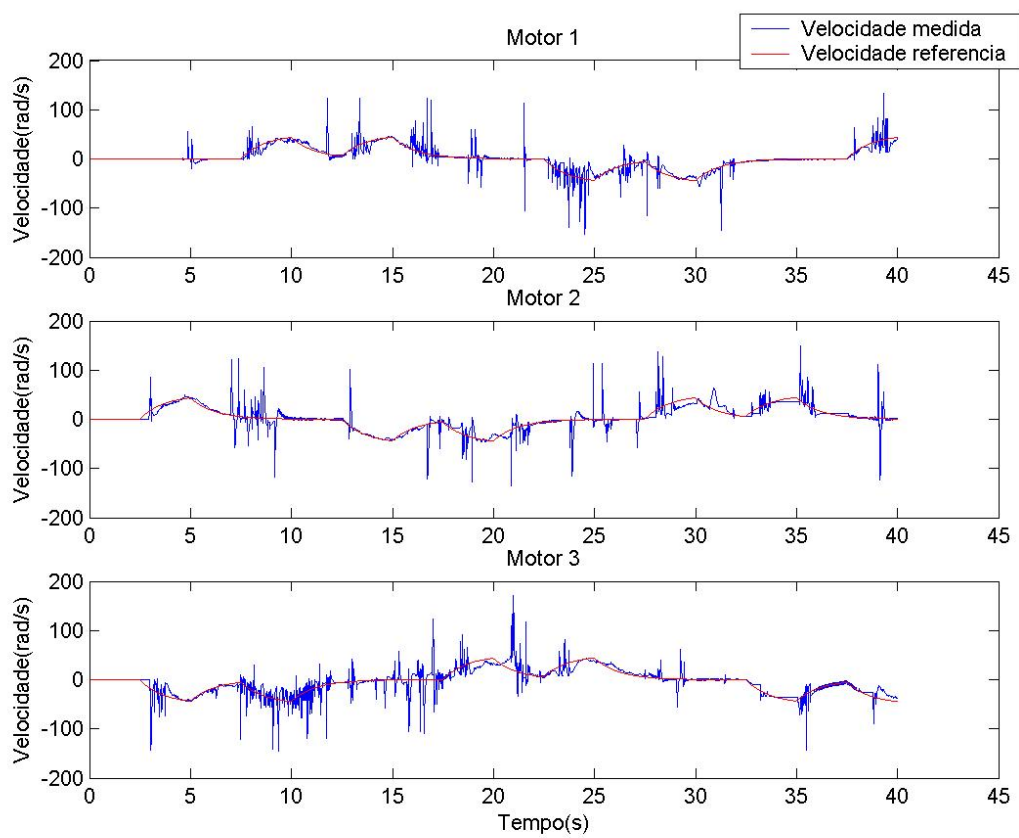


Figura 4.7: Controle dos motores com o robô em contato com o chão.

Conclusões

Neste trabalho foram revisados alguns conceitos relacionados à odometria, modelamento cinemático, mecânica, eletrônica, programação, métodos de estimativa de posição de robôs móveis e métodos de controle de velocidade. Alguns destes conceitos foram utilizados para o desenvolvimento do robô ROHL.

Parte do material utilizado na estrutura mecânica e alguns componentes eletrônicos utilizados no ROHL foram manufaturados a partir de material disponível nos prédios SG11 e SG9. O restante do material que não estavam disponíveis nesses prédios, foi adquirido pelos alunos envolvidos no projeto ou pelo professor Geovany A. Borges, orientador do projeto. Desta forma, o processo de construção do robô pôde ser realizado.

Algumas etapas da implementação do projeto mecânico foram realizadas com a ajuda dos técnicos do SG09, as demais etapas, bem como a implementação dos circuitos eletrônicos projetados, foram implementadas exclusivamente pelos próprios alunos envolvidos no projeto, com a supervisão do professor orientador.

O sistema RTX foi uma solução para o problema encontrado com a implementação de tarefas em tempo real no DSP, uma vez que a tentativa de tornar o sistema implementado no DSP em tempo real preemptivo falhou. Com os resultados obtidos com este sistema, conseguiu-se comprovar a omnidirecionalidade do ROHL, a linearidade dos motores e a eficiência do sistema RTX em realizar as tarefas em tempo real. A eficiência do protocolo de comunicação também pode ser comprovada,

mesmo com a falta de rotinas de detecção e correção de erro. Estes resultados possibilitaram a implementação do sistema de controle com o sistema RTX.

Os resultados obtidos com o robô em suspensão foram excelentes. Já com o robô no solo, os resultados foram bastante satisfatórios, mesmo com o deslizamento das rodas.

O objetivo deste trabalho foi atingido, uma vez que a plataforma móvel omnidirecional foi construída, juntamente com seus circuitos de alimentação, acionamento e controle de velocidade.

Com o robô em seu estado atual, podem-se realizar algumas modificações e melhorias em seu desenvolvimento. Algumas sugestões para trabalhos futuros são a instalação da placa Vortex86-6047 com um sistema de tempo real; substituir as rodas por rodas com maior aderência; substituir os motores por motores mais robustos; instalar novos sensores como acelerômetros ou giroscópio para melhorar o sistema de localização; instalar sensores ultra-som para detecção de obstáculos; implementar um sistema de localização no robô; implementar novas estratégias de controle mais robustas; implementar controle de trajetória para o robô.

Bibliografia

- [Borenstein e Feng 1996]BORENSTEIN, J.; FENG, L. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Transactions on Robotics and Automation*, v. 12, n. 6, p. 869–880, December 1996.
- [Campion, Bastin e D’Andréa-Novel 1996]CAMPION, G.; BASTIN, G.; D’ANDRÉA-NOVEL, B. Structural properties and classification of kinematic and dynamic models of wheeled mobile robots. *IEEE Transactions on Robotics and Automation*, v. 12, n. 1, p. 47–62, February 1996.
- [Heinen 2002]HEINEN, F. J. Sistema de controle híbrido para robôs móveis autônomos. *Universidade do Vale do Rio dos Sinos*, 2002.
- [Kim, Lee e Kwon 2000]KIM, D. S.; LEE, H. C.; KWON, W. H. Geometric kinematics modeling of omni-directional autonomous mobile robot and its applications. *IEEE International Conference on Robotics and Automation*, v. 12, n. 1, p. 2033–2038, February 2000.
- [Michelon 2000]MICHELON, R. *Desvio de Obstáculos utilizando um ambiente simulado para o robô móvel KHEPERA*. [S.l.]: Universidade integrada do Alto Uruguai e das Missões, 2000.
- [Nagatani, Sofue e Tanaka 2000]NAGATANI, K.; SOFUE, S. T. and M.; TANAKA, Y. Improvement of odometry for omnidirectional vehicle using optical flow information. *Systems Engineering Dept. Okayama University*, 2000.
- [Pin e Killough 1994]PIN, F. G.; KILLOUGH, S. M. A new family of omnidirectional and holonomic wheeled platforms for mobile robots. *Transactions on Robotics and Automation*, v. 10, n. 4, p. 480–489, August 1994.

- [Texas Instruments 1997]TEXAS INSTRUMENTS. *TMS320C3x User's Guide*. [S.l.], July 1997.
- [Thuilot, D'Andréa-Novel e Micaelli 1996]THUILOT, B.; D'ANDRÉA-NOVEL, B.; MICAELLI, A. Modeling and feedback control of mobile robots equipped with several steering wheels. *IEEE Transactions on Robotics and Automation*, v. 12, n. 3, p. 375–390, June 1996.
- [Verturcom Inc 1998, 1999]VERTURCOM INC. *RTX 4.3 Users Guide*. [S.l.], 1998, 1999.
- [Wada e Mori 1996]WADA, M.; MORI, S. Holonomic and omnidirectional vehicle with conventional tires. *IEEE International Conference on Robotics and Automation*, p. 3671–3676, April 1996.
- [Wang 1998]WANG, C. M. Estimation and uncertainty analysis for mobile robots. *General Motors Research Laboratories*, p. 1230–1235, April 1998.

Anexos

Esquemáticos

A placa de potência é responsável pelo acionamento dos motores através da ponte H e envio de tensões referentes aos resolveres, sensores de passagem e correntes nos motores para a placa dos microcontroladores. Esta placa se localiza na primeira plataforma do robô.

O DSP foi o microprocessador utilizado no início do projeto para fazer o controle do robô. Esta placa se localizava na segunda plataforma do robô.

A placa de geração de MLP possuía a principal função de geração de MLP através do CI 8254 e envio destes sinais para a placa de potência, mas também tinha a função de fazer o tratamento dos sinais dos resolveres dos motores. Esta placa foi utilizada juntamente com o DSP e se localizava na segunda plataforma do robô.

A placa dos microcontroladores tem as mesmas funções da placa de geração de MLP, com a diferença de que agora estão sendo utilizados microcontraladores ao invés de CI's para a geração destes sinais. Os microcontroladores são escravos do PC, se comunicando com este através da RS 485. Esta placa se encontra na segunda plataforma do robô.

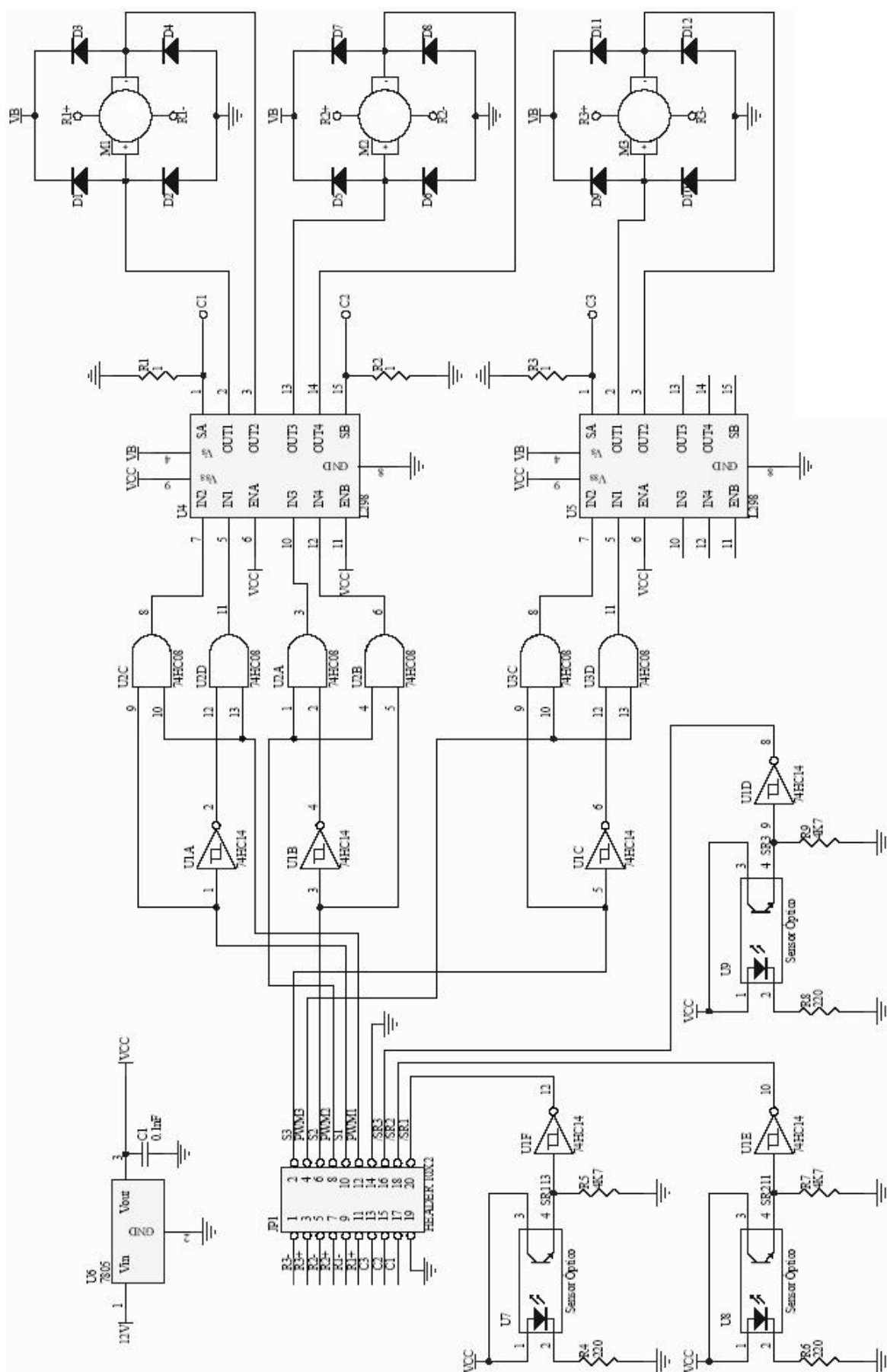


Figura A.1: Esquemático da placa de potência

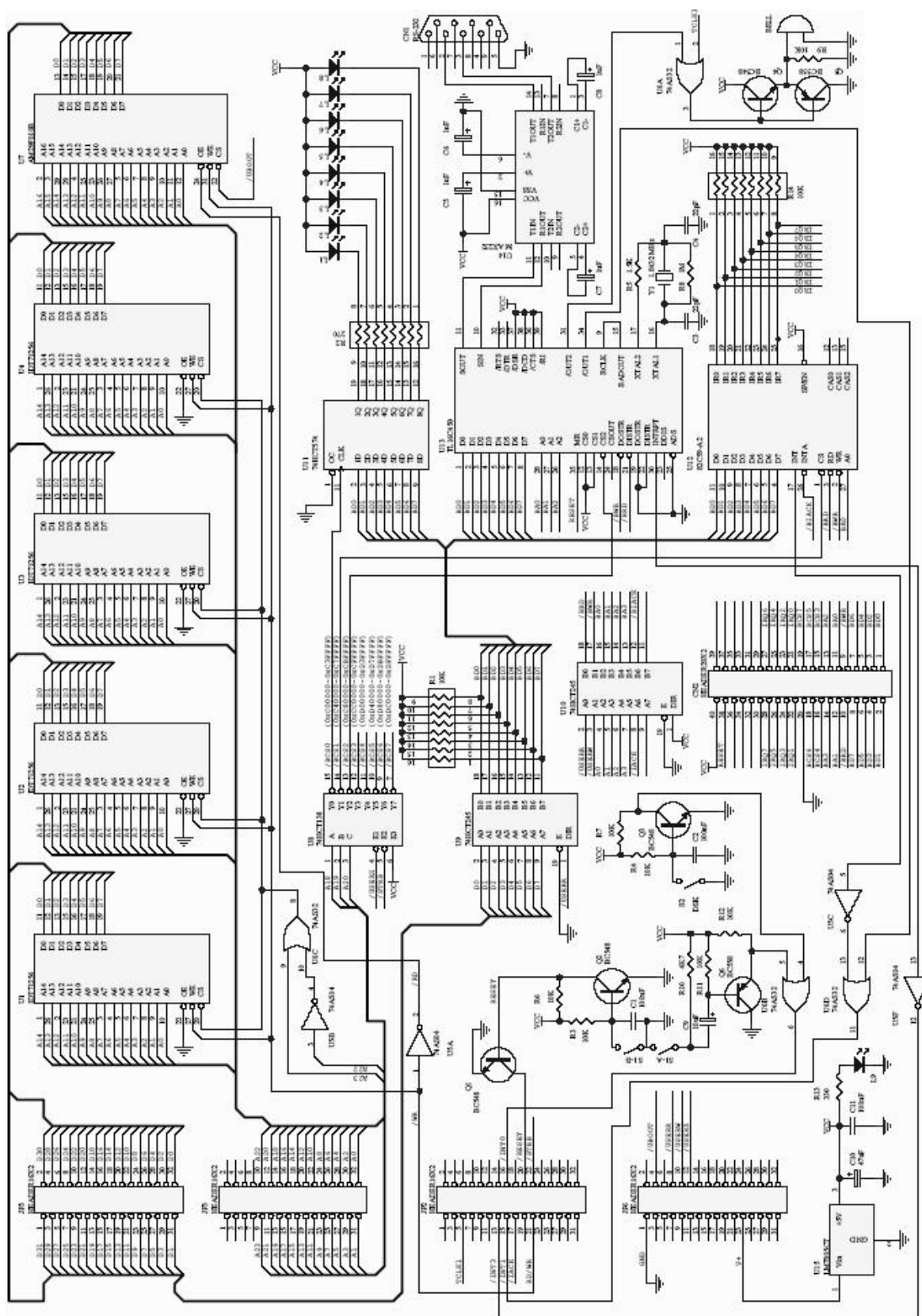
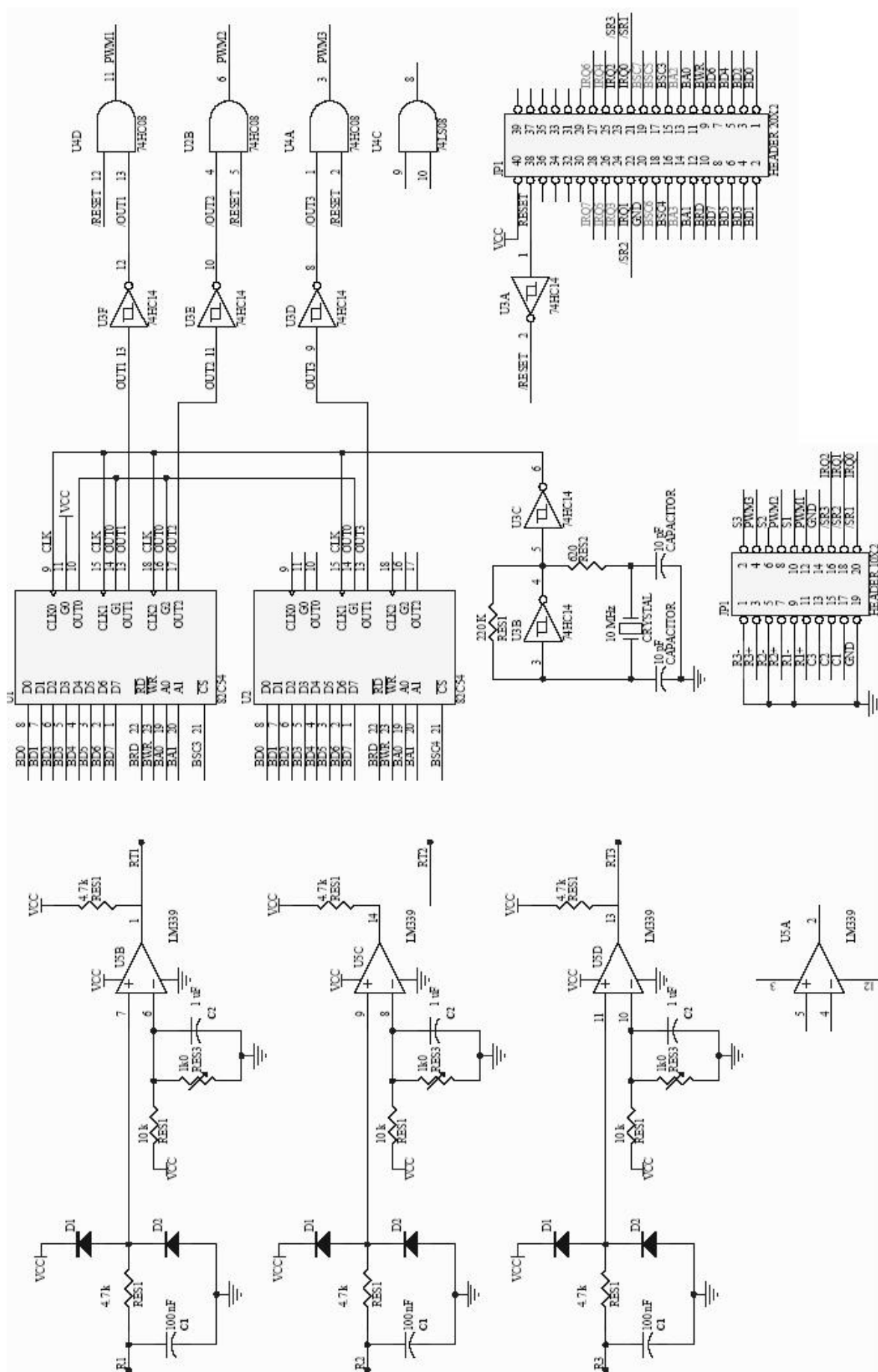


Figura A.2: Esquemático da placa do DSP



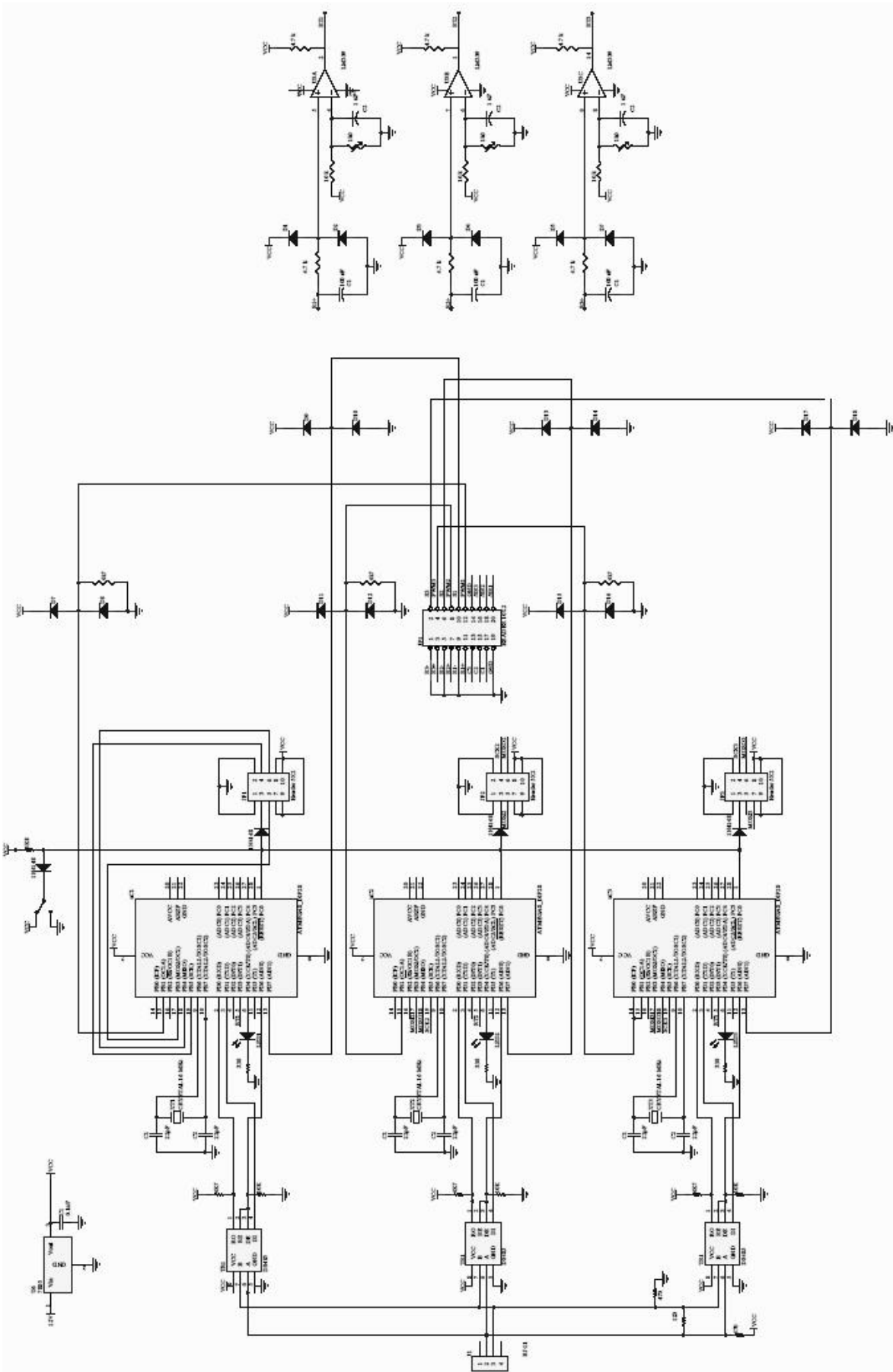


Figura A.4: Esquemático da placa dos microcontroladores

Descrição do CD

O CD em anexo contém os códigos de programas utilizados neste projeto. Ele está dividido em duas partes: códigos do módulo DSP e códigos do módulo PC com AVR.

- Módulo DSP
 - Códigos das funções base de atuação
 - * C31SBC.c - possui funções base para controle do robô, tais como de espera, de ativação sonora e tratamento de interrupções
 - * C31SBC.h - cabeçalho para o arquivo C31SBC.c
 - * C31RTK.c - possui funções para controle do robô em tempo real
 - * C31RTK.h - cabeçalho para o arquivo C31RTK.c
 - Códigos de atuação no robô
 - * hexagono.c31 - programa que gera uma trajetória hexagonal, girando duas rodas por vez durante 1 segundo
 - * wireless.c31 - programa que recebe sinais da porta serial para locomever o robô para quatro lados. Foi utilizado para controle via celular
 - * Rohl_v07 - programa para gerar sinal MLP randômico e coletar velocidades dos motores

- Módulo PC com AVR

- Códigos referentes ao PC

- * Rohl_Lib.c - funções de envio de comandos para os AVR
 - * Rohl_Lib.h - cabeçalho para o arquivo Rohl_Lib.c
 - * ROHL_RTXIdent.c - geração de tarefas para identificação do modelo do robô
 - * ROHL_RTXIdent.h - cabeçalho para o arquivo Rohl_RTXIdent.c
 - * ROHL_RTXValidacao.c - geração de tarefas para validação do modelo do robô
 - * ROHL_RTXValidacao.h - cabeçalho para o arquivo Rohl_RTXValidacao.c
 - * ROHL_RTXHexagono.c - geração de tarefas para controle no solo com trajetória hexagonal
 - * ROHL_RTXHexagono.h - cabeçalho para o arquivo Rohl_RTXHexagono.c
 - * ROHL_RTXContrPI.c - geração de tarefas para controle PI no ar
 - * ROHL_RTXContrPI.h - cabeçalho para o arquivo Rohl_RTXContrPI.c
 - * ROHL_RTXContrPIFiltroRef.c - geração de tarefas para controle PI no ar com filtro no sinal de referência
 - * ROHL_RTXContrPIFiltroRef.h - cabeçalho para o arquivo Rohl_RTXContrPIFiltroRef.c
 - * SerialLibRTX.c - funções para comunicação serial
 - * SerialLibRTX.h - cabeçalho para o arquivo SerialLibRTX.c
 - * GMatrix.h - funções de manipulação de matrizes
 - * GMatrixTypes.h - definição do tipo da matriz

- Códigos referentes aos AVR

- * ROHL_AVR.c - contém as funções de comando comuns a todos AVR
 - * ROHL_AVR.h - cabeçalho para o arquivo ROHL_AVR.c
 - * ROHL_AVR1.c - contém as funções de comando para o AVR 1
 - * ROHL_AVR1.h - cabeçalho para o arquivo ROHL_AVR1.c
 - * ROHL_AVR2.c - contém as funções de comando para o AVR 2
 - * ROHL_AVR2.h - cabeçalho para o arquivo ROHL_AVR2.c
 - * ROHL_AVR3.c - contém as funções de comando para o AVR 3
 - * ROHL_AVR3.h - cabeçalho para o arquivo ROHL_AVR3.c

- Códigos do Matlab®

- * ident.m - fazer identificação do modelo
 - * validacao.m - fazer validação do modelo
 - * plotar.m - gerar gráficos de MLP e velocidade