

Frederico Saraiva Vória

**Sistema de Estacionamento Automático
para um Veículo Autônomo**

Belo Horizonte - MG
Novembro de 2010

Universidade Federal de Minas Gerais
Curso de Graduação em Engenharia de Controle e Automação
Projeto de Fim de Curso

**Sistema de Estacionamento Automático
para um Veículo Autônomo**

Frederico Saraiva Vória

Orientador:
Guilherme A. S. Pereira

Supervisor:
Leonardo A. B. Tôrres

Belo Horizonte - MG
Novembro de 2010

Make everything as simple as possible, but not simpler.

Albert Einstein

Resumo

Na Universidade Federal de Minas Gerais, o Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos desenvolve a automatização de um veículo de passeio. O objetivo do projeto é que o veículo adquira a capacidade de executar determinadas tarefas autonomamente, ou seja, sem a presença de um condutor. O presente projeto de fim de curso aborda o desenvolvimento de um sistema capaz de perceber o espaço correspondente a uma vaga, calcular a trajetória necessária e acionar os atuadores para estacionar o automóvel paralelamente à calçada. A solução gerada foi pensada para poder ser adaptada a veículos com condução humana, facilitando assim a manobra de estacionamento. Foram utilizados sensores de ultra-som para identificar a vaga e observar o ambiente enquanto o carro se movimenta, evitando assim a colisão com obstáculos. Um programa executado no computador de bordo, a partir da estimativa do tamanho da vaga, calcula a trajetória necessária para estacionar o carro. Esse caminho é, então, transformado em valores de comando para os atuadores do ângulo da direção, da aceleração e do freio do veículo. Em alguns testes, os comandos foram seguidos corretamente e foi possível estacionar o carro dentro da vaga sem haver colisão com nenhum obstáculo.

Palavras Chave: Sistema de Estacionamento Automático; Estacionamento de Veículo Autônomo; Medição de Distância por Ultra-som; Geração de Trajetória para Estacionamento;

Abstract

At the Federal University of Minas Gerais, the Autonomous Vehicle Research and Development Group works on the automation of a passenger vehicle. The purpose of this project is to make the vehicle being capable of executing certain tasks autonomously, in other words, without the presence of a driver. This work presents the development of a system capable of perceiving the space available, calculate the trajectory and actuate to park the vehicle parallel to the sidewalk. The solution was thought to be adapted to a human driven vehicle, making parking easier. Ultrasonic sensors were used to identify the space available and observe the environment while the car is moving, avoiding obstacle collision. Departing from the estimated available space, a program running in the car embedded computer calculates the parking trajectory. At that time this path is transformed into command values to the break, acceleration and steering angle of the vehicle. In some tests the commands were followed correctly, and the car was parked in the space between other two cars without any obstacle collision.

Keywords: Automatic Parking System; Autonomous Vehicle Parking; Ultrasonic Distance Measurement; Parking Path Planning;

Agradecimentos

Agradeço a Deus, pelo zelo silencioso e sempre constante. E também por ter me dado força de vontade e motivação durante todo este tempo.

O desenvolvimento deste sistema só foi possível com a ajuda de muitos. Agradeço aos colegas do Laboratório de Sistemas de Computação e Robótica - CORO - por toda a ajuda durante os diversos experimentos realizados ao longo deste trabalho. Em especial aos colegas Elias José, Danilo Lima, Guilherme Castro, Guilherme Becker e Pedro Cotta, que sempre estiveram dispostos a me auxiliar.

Agradeço aos professores Guilherme Pereira e Leonardo Tôrres pela oportunidade de trabalhar em um projeto deste tipo, meu sonho de infância.

Dedico esta realização à minha família, pelo apoio e por acreditarem que tudo isso seria possível.

Sumário

Resumo	i
Abstract	ii
Agradecimentos	iii
Lista de Figuras	vii
Lista de Tabelas	viii
Lista de Abreviaturas	ix
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos	3
1.3 Local de desenvolvimento	4
1.4 Escopo do Trabalho	5
1.5 Revisão Bibliográfica	5
1.5.1 Medição Através de Ultra-som	5
1.5.2 Geração de Trajetória para Robôs	7
1.6 Estrutura do Texto	7
2 Visão geral do veículo autônomo	9
3 Desenvolvimento do Sistema de Localização de Vaga	13
3.1 Subsistema de medição de distância aos obstáculos	13
3.1.1 Microcontrolador	14
3.1.2 Ultra-som	15
3.1.3 Funcionamento do sensor Polaroid	17
3.1.4 Funcionamento do sensor HG-M40DAII	17
3.1.5 Circuito de alimentação dos sensores	18
3.1.6 Circuito de Acionamento e leitura dos Sensores	19
3.1.7 Funcionamento do programa executado no PIC	22
3.1.8 Informações Gerais sobre o Subsistema de Medição de Distância aos Ob- táculos	28
3.2 Subsistema de medição de velocidade	29

3.2.1	Funcionamento do sensor de velocidade	30
3.2.2	Circuito condicionador do Sinal	31
3.2.3	Programa do microcontrolador	32
3.3	Montagem do mapa	34
4	Algoritmos de Geração de Trajetórias e de Controle	36
4.1	Geração da Trajetória	36
4.1.1	Modelo de Ackermann	36
4.1.2	Cálculo da Trajetória	37
4.2	Transformando a trajetória em sinais de comando	40
4.2.1	Algoritmo de Controle	41
4.3	Programa de Operação do Veículo	42
4.3.1	Desenvolvimento do Programa	44
5	Resultados Experimentais	48
5.1	Testes do Sensor Polaroid	48
5.2	Testes do Sensor HG-M40DAII	49
5.3	Testes do Sensor de Velocidade	51
5.4	Estimação das medidas da vaga	51
5.5	Algoritmo para Geração de Trajetórias	53
5.6	Sistema de Estacionamento Automático	55
6	Conclusão	59
6.1	Conclusão	59
6.2	Trabalhos Futuros	60
	Referências Bibliográficas	62

Lista de Figuras

1.1	Acidente ao estacionar	2
1.2	Prejuízo ao motorista	2
1.3	Trânsito movimentado	2
1.4	Não há vagas	2
1.5	Esbarrão na calçada	3
1.6	Padrão da região no espaço típico para medições com feixe de ondas de 50 Khz[Polaroid 1994]	6
2.1	Astra autônomo	9
2.2	Interior do veículo autônomo	10
2.3	Diagrama dos sistema de automação implantados (adaptado de [Santos et al. 2008])	11
3.1	Subsistema de medição de distância aos obstáculos	14
3.2	Placa de Circuito Impresso desenvolvida por estudantes do CORO	15
3.3	Diagrama da medição por tempo de vôo	16
3.4	Gráfico das Amplitudes dos Ecos no tempo	16
3.5	Sensor Polaroid Série 6500	17
3.6	Diagrama de sinais do acionamento do sensor Polaroid 6500	18
3.7	Sensor HG-M40DAII	18
3.8	Diagrama de sinais do acionamento dos sensores HG-M40DAII	19
3.9	Diagrama de sinais gerados pelo PIC para o chaveamento dos sensores	20
3.10	Circuito de alimentação e acionamento dos sensores	21
3.11	Diagrama de sinais que mostra a operação de OU LÓGICO feita com os sinais dos sensores HG-M40DAII	22
3.12	Diagrama de sinais que mostra a operação de E LÓGICO feita com os sinais dos sensores HG-M40DAII e o sinal CONTROL	23
3.13	Diagrama que mostra o circuito eletrônico que realiza as operações lógicas com os sinais	23
3.14	Máquina de Estados do funcionamento do programa do microcontrolador	25
3.15	Local de instalação do sensor Polaroid 6500	28
3.16	Diagrama do local de instalação dos sensores	29
3.17	Locais de instalação dos sensores no veículo	29
3.18	Subsistema de medição de velocidade	30
3.19	Sinais coletados a partir do sensor de velocidade do sistema de freios ABS	31

3.20	Diagrama do circuito de condicionamento do sinal do sensor de velocidade . . .	32
3.21	Sinais obtidos em diferentes pontos da Figura 3.20	32
3.22	Medição das distâncias para geração do mapa	35
4.1	Modelo de Ackermann para um carro (adaptado de [Choset et al. 2005])	37
4.2	Início do cálculo da Trajetória - Definindo posição final do veículo	37
4.3	Cálculo da Trajetória - Definindo os pontos P1 e P2	38
4.4	Trajetória para o Estacionamento	38
4.5	Cálculo da Trajetória - Ponto de Tangência	39
4.6	Cálculo da Trajetória - Triângulo Retângulo extraído da Figura 4.5	39
4.7	Cálculo da Trajetória - Círculos usados para gerar a trajetória	40
4.8	Calculando o comando para o sistema de controle da direção	41
4.9	Interface gráfica do programa desenvolvido	43
4.10	Desenho do carro completando a manobra do estacionamento	44
4.11	Diagrama UML simplificado da Classe Carro	45
4.12	Diagrama UML simplificado das classes desenvolvidas neste trabalho	46
5.1	Testes em laboratório usando o sensor Polaroid 6500	49
5.2	Testes em laboratório usando um sensor HG-M40DAII	50
5.3	Dados dos sensores de velocidade e ultra-som em testes de campo	52
5.4	Dados dos sensores de velocidade e ultra-som em testes de campo	52
5.5	Imagens retiradas de um vídeo que mostra o veículo estacionando	55
5.6	Resultado da validação do algoritmo de geração de trajetórias via MATLAB . .	55

Lista de Tabelas

- 5.1 Testes Sensor Polaroid 57
- 5.2 Testes Sensor HG-M40DAII 58

Lista de Abreviaturas

PDVA	Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos
UFMG	Universidade Federal de Minas Gerais
CORO	Laboratório de Sistemas de Computação e Robótica
PWM	Do inglês, Pulse Width Modulation
GPS	Do inglês, Global Positioning System
USB	Do inglês, Universal Serial Bus
ABS	Do inglês, Anti-lock Brake System
LED	Do inglês, Light Emitting Diodes

Capítulo 1

Introdução

Aplicações envolvendo sistemas automatizados se tornam cada vez mais comuns ao passo que aumenta a diversidade de tarefas que estes podem realizar. Seu emprego em indústrias permitiu atingir níveis de produção inalcançáveis pelos operários. Além de realizar trabalhos repetitivos com maior rapidez, estes sistemas executam movimentos com maior perfeição que humanos. Outras aplicações mostram que sistemas robóticos são capazes de explorar um ambiente desconhecido ou perigoso. Através de sensores, são obtidas informações do local, que podem ser usadas para muitos fins.

Robôs auto controlados são caracterizados pela integração de vários sensores e pelo seu comportamento inteligente. Um tema muito pesquisado no que tange a robótica autônoma é a navegação. Ela pode ser entendida como os procedimentos necessários para que um sistema robótico mova-se de uma posição inicial para uma final percorrendo um caminho por ele planejado.

Essa tarefa pode ser dividida em cinco partes: obtenção das principais características do ambiente; determinação da posição do robô em relação a potenciais obstáculos; cálculo de um caminho passível de ser executado; transformação do caminho em comandos para os atuadores, objetivando que o percurso calculado anteriormente seja descrito no movimento; efetivação desses comandos pelo controlador, de forma que a trajetória seja percorrida.

A realização de uma manobra de estacionamento por um veículo não tripulado caracteriza-se como um trabalho de robótica autônoma em que é necessária a execução dessas cinco tarefas. No presente trabalho são apresentadas e discutidas as formas encontradas para efetuá-las.

1.1 Motivação

Estacionar um automóvel é uma manobra que exige do motorista uma boa orientação espacial. A dificuldade aumenta caso não se tenha muita informação sobre a localização do carro em relação à vaga onde se deseja parar. Os retrovisores nem sempre cobrem os locais em que

se necessita enxergar porque apresentam os chamados pontos-cegos, que são regiões ao redor do veículo em que não se pode ver. Devido a isso, cenas como as mostradas nas Figuras 1.1 e 1.2 tornam-se comuns nas vias de grandes cidades.



Figura 1.1: Acidente ao estacionar



Figura 1.2: Prejuízo ao motorista

Dirigir em locais que apresentam trânsito intenso é muito estressante, sobretudo em horários de pico. Condutores com limitações motoras, pouca confiança ao dirigir ou pouca experiência ao volante encontram mais dificuldade ao realizar esta tarefa. Parar em locais movimentados traz perigo aos ocupantes do veículo, ainda mais quando não se consegue estacionar com perfeição na primeira tentativa. O risco aumenta quando o condutor não mantém a calma para refazer a manobra. Quando os outros motoristas buzina ou de alguma maneira reprovam a forma como se tentou estacionar, a situação de estresse agrava-se ainda mais. Acidentes podem ser causados quando é necessária a execução de várias manobras para que o carro seja colocado dentro da vaga.

O aumento do número de automóveis em circulação tem como consequência uma maior concorrência pelos locais públicos destinados a estacionamento. Dessa forma, tem-se uma redução no número de vagas, que também têm seu espaço cada vez mais reduzido. As Figuras 1.3 e 1.4 mostram uma situação corriqueira do trânsito nas cidades brasileiras.



Figura 1.3: Trânsito movimentado



Figura 1.4: Não há vagas

Dessa forma, exige-se do condutor uma maior habilidade para parar sem causar esbarrões nos outros carros ou na calçada, motivo de prejuízo para os proprietários de veículos, como ilustrado pela Figura 1.5.



Figura 1.5: Esbarrão na calçada

Alguns sistemas de auxílio ao estacionamento utilizam sensores instalados no para-choque traseiro do veículo. Estes têm como objetivo medir a distância a eventuais objetos que estejam posicionados próximo a eles. Ao estacionar, é medida a distância em relação ao para-choque dianteiro do veículo parado atrás do espaço correspondente à vaga. Mas a medição só é feita quando algum objeto entra na região do espaço que pode ser percebida pelo sensor. Esses sistemas têm capacidade para acusar distâncias entre 40 e 150 centímetros. Quando o obstáculo está mais próximo que 40 cm do automóvel, o sistema para de medir. Muito se faz propaganda sobre esse tipo de sistema, o que tem como objetivo induzir o consumidor a acreditar que esta é uma tecnologia que será de grande serventia para o seu dia-a-dia. Mas, paga-se muito pela instalação de sensores que mostram apenas a distância medida em relação a um objeto. Tal fato, algumas vezes, até impede que o motorista realize uma manobra com maior acerto ou faz com que ele desista de parar naquele local.

Montadoras como Ford, Mercedes e BMW já apresentaram protótipos de sistemas capazes de estacionar o automóvel em que o condutor necessita parar o carro próximo à vaga e apertar um botão. No Brasil, recentemente, a Volkswagen lançou no mercado um modelo equipado com uma câmera de vídeo e atuador na direção, capaz de simplificar a manobra para o motorista. Neste, o condutor aciona acelerador e freio, seguindo orientações do sistema. Mas, em todos esses casos, a tecnologia foi importada de outro país.

Ainda é incipiente o estudo e desenvolvimento na área de automóveis autônomos dentro das universidades brasileiras. Nota-se que existe uma distância entre as montadoras e os pesquisadores, o que se traduz em pouco apoio e, conseqüentemente, pouca transferência de tecnologia. Aumentar o desenvolvimento nesta área dentro da Escola de Engenharia foi um fator que motivou em muito a realização deste trabalho. A partir dele, espera-se aumentar o apoio a novos projetos.

1.2 Objetivos

Foram objetivos deste trabalho:

1. Implementar um sistema formado por uma rede de sensores divididos em dois subsistemas. Um deles utiliza sensores ultra-sônicos, que medem as distâncias entre os obstáculos e o veículo, enquanto este se movimenta. O outro subsistema usa os dados de um dos sensores de velocidade do sistema de freios ABS, responsável por informar a velocidade do carro. Os dados coletados seriam enviados para um microcomputador para que fosse feito um mapa do local onde o automóvel se encontra. Assim, automaticamente, o espaço referente a uma possível vaga seria detectado.
2. Desenvolver um algoritmo que, utilizando os dados obtidos sobre a vaga, gerasse uma trajetória para se estacionar. O caminho gerado seria livre de colisão com possíveis obstáculos (calçada, outros veículos, postes e etc) e facilmente traduzível em comandos para os atuadores do automóvel utilizado.
3. Desenvolver um sistema de controle automático para o veículo, para que fosse possível ao mesmo seguir a trajetória de estacionamento calculada, mediante a efetivação de comandos de troca de marchas, aceleração, frenagem e direção. Este faria com que o carro seguisse a trajetória calculada e evitasse choque com os obstáculos.

Estes objetivos tiveram como referência o desenvolvimento de um sistema que pudesse ser aplicado ao CADU - Carro Autônomo Desenvolvido na UFMG. Este veículo vem sendo desenvolvido pelo grupo de Pesquisa e de Desenvolvimento de Veículos Autônomos - PDVA. Como objetivo secundário, este sistema deveria ser facilmente adaptado aos veículos tripulados, isto é, aqueles em que a atuação é feita pelo condutor. Assim, seria possível auxiliar o motorista a estacionar o veículo com mais segurança em vias movimentadas, em vagas menores, em locais ou sob condições de pouca visibilidade e evitar o choque com obstáculos ou outros veículos.

1.3 Local de desenvolvimento

O projeto foi desenvolvido na própria universidade, no Laboratório de Sistemas de Computação e Robótica, do departamento de Engenharia Elétrica. O CORO, como é chamado, é um laboratório de pesquisa e desenvolvimento que atua nas áreas de robótica, visão computacional, processamento de imagens digitais, sistemas integrados de hardware e software, sistemas a eventos discretos, instrumentação e controle por computador. É um dos laboratórios associados ao Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos (PDVA), que atua no desenvolvimento de sistemas de instrumentação, navegação e controle embarcados para veículos terrestres e aéreos, autônomos ou semi-autônomos [Laboratório CORO, 2009].

Os programas para os microcontroladores e para o microcomputador, os circuitos dos sistemas de medição de velocidade e da rede de sensores de ultra-som foram desenvolvidos dentro

do próprio laboratório, bem como testes de bancada que puderam ser realizados.

Os subsistemas desenvolvidos foram embarcados em um veículo autônomo e alguns testes em estacionamentos dentro da própria universidade foram realizados. Trabalhos anteriores automatizaram os acionamentos do câmbio, do freio, da aceleração e do ângulo da direção deste automóvel, pertencente ao grupo PDVA.

1.4 Escopo do Trabalho

Foi utilizado o conhecimento adquirido nas disciplinas de Instrumentação Industrial, Sistemas Distribuídos para Automação, Sistemas Digitais, Sistemas Processadores e Periféricos e Eletrônica Analógica e Digital para se implementar uma rede de sensores capaz de fazer a leitura do espaço referente à vaga em que se deseja estacionar o veículo.

Além de encontrar a vaga, foi desenvolvido um algoritmo que para geração de uma trajetória segura para que o veículo a seguisse e estacionasse automaticamente. Disciplinas que abordam este tipo de assunto seriam as ligadas a robótica, como Introdução a Robótica.

Foi desenvolvido um programa que, através de uma interface computador/acionamentos, envia os valores de comando para os atuadores da direção, aceleração, freio e câmbio, de acordo com a trajetória gerada, para guiar o veículo. Para isto, disciplinas como Controle em Tempo Real utilizando-se DSPs¹ e outras com foco em programação de computadores (Algoritmos e Estruturas de Dados 1 e 2, Automação em Tempo Real, Análise, Projeto e Programação Orientada a Objetos e Projeto Assistido por Computador) serviram de base para a parte de implementação computacional.

Para a realização de testes em laboratório, foram usadas as lições aprendidas nas disciplinas práticas de laboratório de Sistemas Digitais, de Eletrônica Analógica e Digital, de Circuitos Elétricos, de Eletrônica de Potência e Laboratório de Controle e Automação I.

1.5 Revisão Bibliográfica

1.5.1 Medição Através de Ultra-som

Sistemas de sensoriamento através de ultra-som são usados para fazer o mapeamento do ambiente em torno de um robô, para auxiliar os sistemas de navegação e para posicionar o robô em relação aos obstáculos. A determinação da distância se baseia na medição do tempo decorrido entre a emissão e a recepção de uma onda, após sua reflexão em um obstáculo (*time-of-flight*). Para se medir a distância, o sensor emite um feixe de ondas de frequência próxima

¹do inglês *Digital Signal Processor*

a 50 KHz. Em seguida, mede-se o tempo que leva para a onda refletida retornar ao transmissor.

Em [MAJCHRZAK, MICHALSKI e WICZYNSKI 2009] é apresentada uma forma de calibrar um medidor ultra-sônico com base na realização de testes em laboratório. O sensor foi posicionado em frente a uma parede lisa a diferentes distâncias e foram realizadas 100 medidas para cada distância. Condições diferentes das encontradas no momento da calibração do sistema são causa de erros na medição. Incluem-se as condições ambientais em que a propagação de ondas ultra-sônicas se realiza (temperatura e humidade, por exemplo) e a presença de outros emissores de ondas acústicas de frequências semelhantes. Outro fator é o objeto refletor não estar posicionado em frente ao sensor. Característica intrínseca destes medidores é a falta de informação da localização exata do objeto ao qual está sendo feita a medição da distância. Isto ocorre devido ao sensor apresentar uma região de cobertura que varia de alguns graus em torno do seu eixo principal, como mostrado na Figura refFigPatt. O objeto (ou parte dele) que estiver a uma menor distância do sensor é detectado.

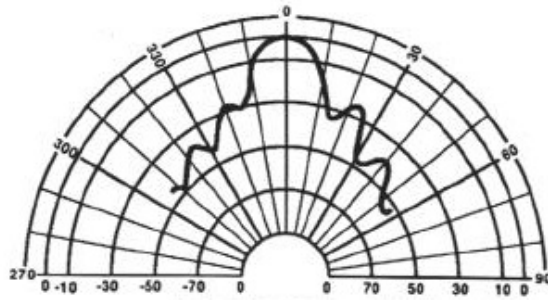


Figura 1.6: Padrão da região no espaço típico para medições com feixe de ondas de 50 KHz[Polaroid 1994]

Sensores de ultra-som são usados na indústria para a realização de diferentes tarefas. Alguns são capazes de medir a distância entre objetos usando-se diferentes frequências. Com o uso de ultra-som é possível se fazer medições de distâncias curtas, longas ou de posição angular [MARIOLI, SARDINI e A 1988]. Encontram-se aplicações para este tipo de medidor em diversas indústrias [HAUPTMANN, HOPPE e PUTTMER 2002] e em sistemas robóticos [DIRK 2002][NAKAHIRA et al. 2001]. Seu uso na indústria automobilística vem sendo cada vez maior. Em [CARULLO e PARVIS 2001] este tipo de medidor é usado em um sistema de suspensão ativa.

Em [RASSHOFER e GRESSER 2005] compara-se o uso de sensores para medição de distância por meio de laser, ultra-som e radares de curto alcance. Neste trabalho, é apresentando um estudo que mostra as condições em que cada um não consegue medir bem. [ABAD e KAKINAMI 2007] estuda o uso de ultra-som em conjunto com outro tipo de medidor de distância para realizar a medição do espaço correspondente a uma vaga. Em [PARK et al. 2008] estuda-se a detecção de espaço para estacionamento usando sensores ultra-sônicos em conjunto. A característica da medição ilustrada pela Figura refFigPatt influencia

na detecção de bordas, problema encontrado ao se fazer medições para encontrar o espaço correspondente à vaga. Conseqüentemente, o espaço medido é diferente do real por causa dessa característica dos medidores. Existem vários sensores para este tipo de sistema de assistência, incluindo LRF (*Laser Range Finder*), câmera e SRR (*Short Range Radar*). O laser é capaz de obter informações precisas de distância angular até 180°. No entanto, faz-se apenas a varredura em um plano. O laser também não consegue detectar superfícies de cor preta por propriedade da luz. No caso de utilizar radares de curto alcance, é possível obter informações sobre os ângulos, distâncias e velocidade relativa. Mas, estes apresentam alguns problemas devido ao fenômeno do espalhamento nas bordas e seu preço é muito superior ao dos outros sensores.

1.5.2 Geração de Trajetória para Robôs

Um veículo autônomo pode ser estudado como um robô que apresenta algumas restrições quanto à sua movimentação. Para a tarefa de se estacionar um carro, é preciso que seja gerada uma trajetória que possa ser seguida pelo automóvel. [Dubins Curves, 2009] descreve um método em que são geradas curvas de comprimento mínimo compostas por arcos de círculos ligados por segmentos de retas. Entretanto, sua aplicação é voltada para robôs que se movem unicamente em um sentido.

[Reeds-Shepp Curves, 2009] estendem este algoritmo para o veículo que pode se mover tanto para frente como para trás. Uma característica das trajetórias geradas por estes algoritmos é que, para o caso da sua execução por algum robô, o mesmo deverá parar sempre quando for necessário mudar a orientação das rodas responsáveis pela direção. Para evitar este tipo de situação, Faichard e Scheuer [SCHEUER e FRAICHARD 1997][SCHEUER e FRAICHARD 2004] descrevem um método em que as novas trajetórias são compostas por arcos de círculos, arcos de curvatura constante e segmentos de retas. Assim, não é necessário que o veículo pare para reorientar as rodas dianteiras.

[OETIKER, BAKER e GUZZELLA 2009] mostram uma forma de orientar o veículo através de campos de navegação. Estes algoritmos utilizam um modelo cinemático do comportamento de um automóvel em movimento, fazendo-se simplificações sobre sua dinâmica. Em [PEPY, LAMBERT e MOUNIER 2006] é apresentado um método de geração de trajetória tomando-se como base o modelo dinâmico de um veículo.

1.6 Estrutura do Texto

Esta monografia está organizada em 5 capítulos. O primeiro apresentou a motivação e os objetivos da realização deste projeto. É também apresentada uma revisão bibliográfica discorrendo sobre os principais temas que abrangem este trabalho. O Capítulo 2 apresenta o processo para o qual é desenvolvido o sistema proposto.

O Capítulo 3 descreve o sistema de localização de vaga para estacionamento desenvolvido. São apresentados os sistemas que coletam os dados e a transformação desses dados em informações para serem aproveitadas posteriormente. O Capítulo 4 apresenta os algoritmos de geração de trajetória e de controle para estacionar o CADU. O Capítulo 5 traz os testes feitos em ambiente interno e externo e discussões sobre seus resultados. Por fim, no último capítulo, são apresentadas as conclusões sobre os resultados obtidos e são apresentadas sugestões para trabalhos futuros.

Capítulo 2

Visão geral do veículo autônomo

A implantação foi realizada em um veículo Chevrolet ASTRA 2003/2004 (Figura 2.1), pertencente ao grupo PDVA. Este é utilizado pelo grupo para a concepção de um automóvel de passeio autônomo. Alguns trabalhos anteriores implantaram sistemas de automação que permitem a condução do veículo sem a presença de um condutor.



Figura 2.1: Astra autônomo

O veículo, inicialmente, já dispunha dos seguintes recursos: acelerador eletrônico com sistema *drive-by-wire*, direção hidráulica, sistema de freios ABS e câmbio automático. Tais recursos simplificaram a automatização dos acionamentos da aceleração, câmbio e direção e freio.

Uma característica comum aos sistemas que foram implantados é que estes não deveriam impedir que um motorista conduzisse o automóvel normalmente, quando isso fosse necessário. A Figura 2.2 mostra o interior do automóvel.



Legenda:

- 1 - Direção
- 2 - Painel de Chaves
- 3 - Computador de Bordo
- 4 - Freio
- 5 - Câmbio
- 6 - Acelerador

Figura 2.2: Interior do veículo autônomo

Um painel de chaves que se localiza pouco à frente do câmbio (número 2 na Figura 2.2) permite que os acionamentos sejam colocados em modo manual, aquele em que o motorista é responsável por conduzir o veículo; ou automático, situação em que o computador comanda os acionamentos. É possível se alterar o modo da direção, do câmbio, do acelerador e do freio em separado.

Em modo automático, os controladores de cada sistema recebem de um computador central (Tablet PC) os respectivos valores de referência para suas variáveis controladas. O computador em questão é ainda responsável por comunicar a todos os sistemas de acionamento quando há ocorrência de uma eventual situação de emergência. A comunicação é estabelecida via interface USB¹ (entre o computador central e os microcontroladores PIC²) para as malhas de freio, aceleração e câmbio; e via protocolo serial padrão RS-232, no caso do controlador da direção.

Cada acionamento configura-se em uma malha de controle independente. Por meio de uma interface entre o computador e os sistemas de controle, valores de referência são enviados aos controladores responsáveis por efetivar os comandos recebidos. Outra forma de se gerar esses valores é através da execução de um programa com essa finalidade, como o usado neste trabalho.

Toda a instrumentação embarcada é alimentada pela bateria de 12 V do veículo, a mesma responsável por alimentar os circuitos elétricos e eletrônicos do carro. Os microcontroladores, entretanto, são alimentados por meio das conexões USB ou serial com o computador central, que fornecem suporte para tanto. O sistema de emergência, ativado por botões, é alimentado

¹Do inglês, *Universal Serial Bus*

²Microcontrolador PIC18F2550, da fabricante Microchip Technology Inc.

independentemente dos demais. Dessa forma, garante-se que este funcione mesmo em casos de falhas de comunicação.

A Figura 2.3 mostra um diagrama dos sistemas de automação atualmente implantados no veículo.

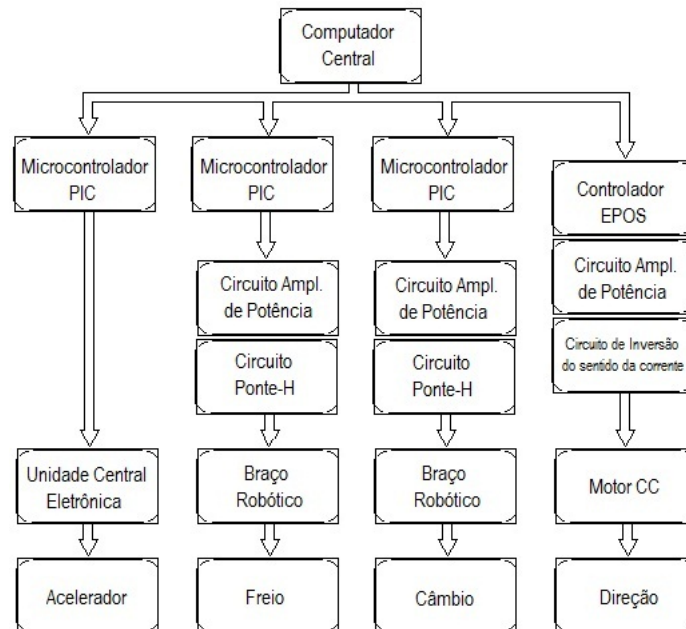


Figura 2.3: Diagrama dos sistema de automação implantados (adaptado de [Santos et al. 2008])

Para acelerar (número 6 na Figura 2.2), atua-se diretamente sobre o sistema eletrônico interno ao veículo. Este sistema é controlado por uma Unidade de Comando Eletrônico (UCE) que controla também a razão da mistura ar/combustível que entrará na câmara de combustão; o ventilador de arrefecimento do motor e o avanço da centelha de ignição. A posição do pedal do acelerador é traduzida por meio de um sensor e o sinal elétrico resultante é enviado à UCE. Em modo automático, conforme tem-se a geração e envio de um sinal PWM³ por um microcontrolador PIC18F2550 [PIC18F2550], que correspondente à aceleração desejada.

A troca de marchas (número 5 na Figura 2.2) é feita por meio do acionamento de um atuador linear acoplado à alavanca de câmbio, que pode mover-se em ambos os sentidos. Uma malha de controle de posição é implementada em um microcontrolador para o posicionamento da alavanca. O dispositivo é acionado por um sinal de tensão modulado em PWM pelo microcontrolador e amplificado por um circuito de potência Simple-H [SimpleH], da fabricante Robot Power. Configurado como uma ponte-H, este circuito permite a extensão ou retração do atuador. Um sinal de tensão de um potenciômetro interno ao atuador é usado como realimentação.

O acionamento do sistema automatizado dos freios (número 4 na Figura 2.2) é apresentado em [Baleeiro 2009]. Este sistema também utiliza um atuador linear, uma malha de controle

³Do inglês, *Pulse Width Modulation*

de posição implementada em um microcontrolador e um circuito do tipo ponte-H. O atuador está acoplado ao pedal do freio por meio de uma montagem mecânica que não atrapalha o motorista, caso o veículo esteja sendo guiado por um condutor humano. Em situações de emergência ou eventuais panes do sistema eletrônico, o controle do freio pode ser feito facilmente pelo motorista, por questões de segurança. O sistema de freios nativo do veículo é do tipo ABS de 4 canais, com um sensor de velocidade de relutância magnética variável em cada roda.

O sistema de acionamento da direção (número 1 na Figura 2.2) é composto por: um motor de corrente contínua Maxon RE40, acoplado ao volante por meio de uma corrente e catracas; um sensor para se medir posição angular do tipo encoder; e um controlador EPOS 24/5, também da fabricante Maxon. As posições angulares do eixo do motor, por meio de uma curva de calibração, foram associadas aos ângulos de esterçamento das rodas dianteiras do veículo.

Capítulo 3

Desenvolvimento do Sistema de Localização de Vaga

Para a tarefa de medir o espaço correspondente à vaga para o estacionamento foi construído um sistema multissensorial, composto por dois subsistemas. Um deles executa a tarefa de medição de distâncias entre o carro e os obstáculos com o uso de sensores de ultra-som. O outro é responsável por informar ao computador central dados de velocidade obtidos por meio do sensor do sistema de freios ABS.

Este capítulo traz uma breve apresentação do funcionamento de um microcontrolador, para que os itens apresentados posteriormente possam ser mais facilmente compreendidos. Em seguida, é mostrada uma explicação curta da técnica usada para medir distâncias por meio de sensores de ultra-som. Então, são apresentados os subsistema de medição de distâncias aos obstáculos e de medição de velocidade implementados.

3.1 Subsistema de medição de distância aos obstáculos

A região livre de obstáculos é encontrada com o auxílio de oito sensores de ultra-som, acionados por um microcontrolador PIC18F2550, que se comunica com o computador central. Este subsistema tem como objetivos:

1. Realizar medições para que:
 - (a) o carro parado atrás da vaga seja detectado;
 - (b) o carro parado à frente da vaga seja detectado;
2. Fazer medições para que a menor distância medida entre a ocorrência dos eventos (a) e (b) seja encontrada.
3. Observar o ambiente ao redor do veículo, enquanto este executa a manobra de estacionar.

Uma figura esquemática desse subsistema é mostrada na Figura 3.1. Ele é composto por:
- 1 sensor de ultra-som marca Polaroid, Série 6500, que tem capacidade de medir distâncias

- entre 17 cm e 10 metros [Polaroid 1994], quando usada a calibração padrão de fábrica;
- 7 sensores de ultra-som marca HG-M40DAII, que medem distâncias entre 32 cm e 4 metros [HAGISONIC 2009], quando usada a calibração de fábrica;
- 1 matriz de contatos, em que estão montados os circuitos de acionamento e alimentação dos sensores e do microcontrolador;
- 1 placa de circuito impresso em que está montado o microcontrolador PIC18F2550, responsável por acionar os sensores ligados a ele e comunicar ao computador as informações obtidas.
- Cabos de par trançado que fazem as ligações entre a matriz de contatos e os sensores.

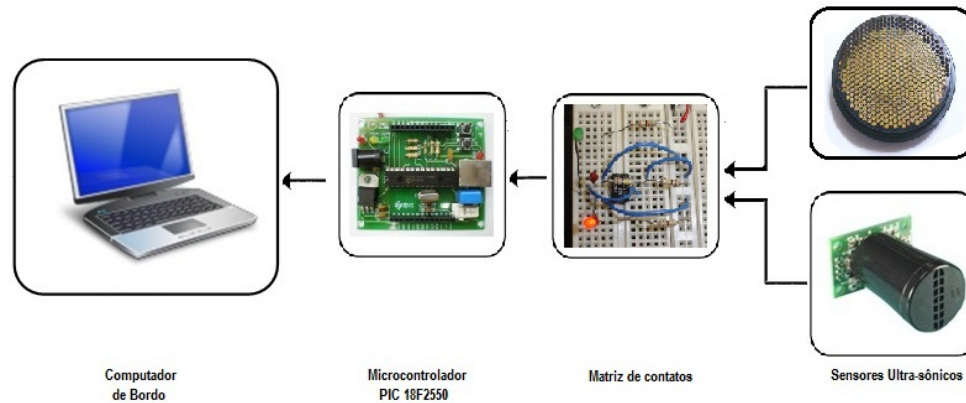


Figura 3.1: Subsistema de medição de distância aos obstáculos

3.1.1 Microcontrolador

Foi utilizado um microcontrolador PIC18F2550 para o acionamento e pré-processamento dos dados medidos pelos sensores. Este dispõe de periféricos úteis ao desenvolvimento de sistemas de controle tais como conversores A/D, temporizadores, interrupções, módulos PWM etc.

Para a troca de dados entre o PIC e um computador com sistema operacional Windows, a fabricante Microchip disponibiliza um pacote de funções (MCHPFSUSB Firmware[MCHPFSUSB]). A comunicação entre o microcontrolador e o computador de bordo é feita via conexão serial RS-232. Na prática, porém, o Tablet PC estabelece a comunicação a partir de uma conversão do protocolo RS-232 para USB.

O microcontrolador PIC18F2550 dispõe de 4 temporizadores, com diferentes características, que dão suporte às mais diversas aplicações. Um temporizador é um contador que incrementa um registrador interno a cada ciclo de instrução. Ao final da contagem de um período, o registrador interno é zerado e pode haver ou não a geração de uma interrupção. Existe um registrador que informa a próxima instrução a ser executada. Uma interrupção faz com que este aponte para a primeira de uma série de instruções responsável pelo tratamento das interrupções. Ao final da execução destas, este retorna ao valor que possuía antes do desvio e segue-se o fluxo normal do programa.

Uma interrupção externa é o evento gerado pela transição do nível de tensão em um terminal configurado como sendo de entrada do microcontrolador. Os terminais de entrada possuem um seguidor de tensão que considera nível lógico alto qualquer valor superior a 80% da tensão de alimentação do PIC. Os valores abaixo de 20% são considerados nível lógico baixo.

O microcontrolador provê também mecanismos para que o nível de tensão em um terminal configurado como sendo de saída possa ser manipulado (ora assumindo nível lógico alto, ora baixo), podendo ser usado para a geração um sinal retangular. A geração de ondas retangulares é possível através do uso de temporizadores e pela manipulação dos níveis de tensão nos pinos de saída. O período do temporizador é configurado de acordo com tempo em que o sinal deve permanecer em cada nível. A interrupção gerada pelo término da contagem de um período marca o momento da transição do nível de tensão. Para o presente trabalho, um temporizador foi utilizado também como o contador do tempo decorrido entre as transições de subida de nível em dois pinos.

A configuração e utilização dos recursos oferecidos é feita por meio das funções da biblioteca de compilação C18, fornecida pela Microchip para a programação de microcontroladores PIC da família 18 [C18].

Foi usada a placa desenvolvida por outros estudantes no laboratório CORO mostrada na Figura 3.2.

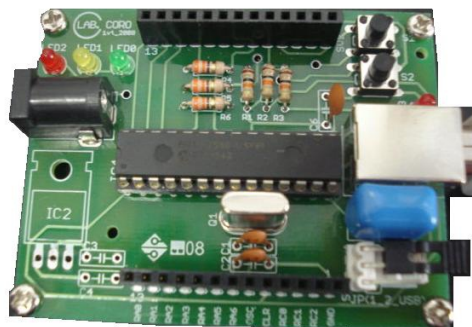


Figura 3.2: Placa de Circuito Impresso desenvolvida por estudantes do CORO

Esta placa consiste em conector e circuito regulador de tensão para alimentação do microcontrolador, *LEDs* ligados a alguns terminais de saída, locais para ligação de fios nos terminais configuráveis de entrada/saída e de terra do microcontrolador e demais componentes necessários ao funcionamento do PIC.

3.1.2 Ultra-som

Sensores ultra-sônicos são usados para se determinar a proximidade a objetos sem que haja contato físico. A implementação mais convencional para a realização desta tarefa baseia-se

no tempo-de-vôo entre um pulso e seu eco (*time-of-flight pulse-echo technique*). A Figura 3.3 apresenta cenas que ilustram esse tipo de medição.

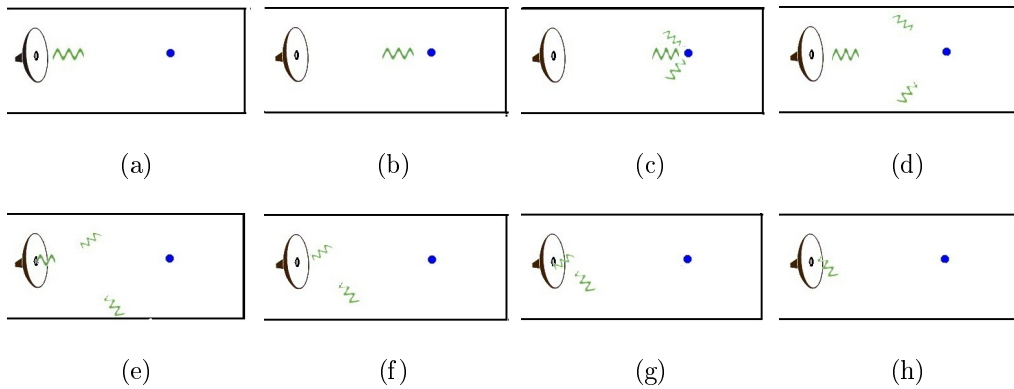


Figura 3.3: Diagrama da medição por tempo de vôo

A Figura 3.3a ilustra o momento em que o feixe de ondas é emitido. A Figura 3.3b mostra o instante anterior às ondas serem refletidas pelo obstáculo. A Figura 3.3c mostra que as ondas são refletidas em várias direções. A seqüência que vai da Figura 3.3d até a Figura 3.3h ilustra o fato de as ondas refletidas percorrerem distâncias diferentes até serem percebidas pelo sensor.

A distância ao obstáculo é proporcional ao tempo decorrido entre a emissão do feixe e a detecção do primeiro eco cuja amplitude exceda a um valor limiar. Em algumas aplicações, os demais ecos podem ser usados na caracterização da forma ou movimento do refletor. Em desenvolvimentos mais simples, esses são descartados. A Figura 3.4 apresenta o sinal resultante da captação dos ecos no tempo.

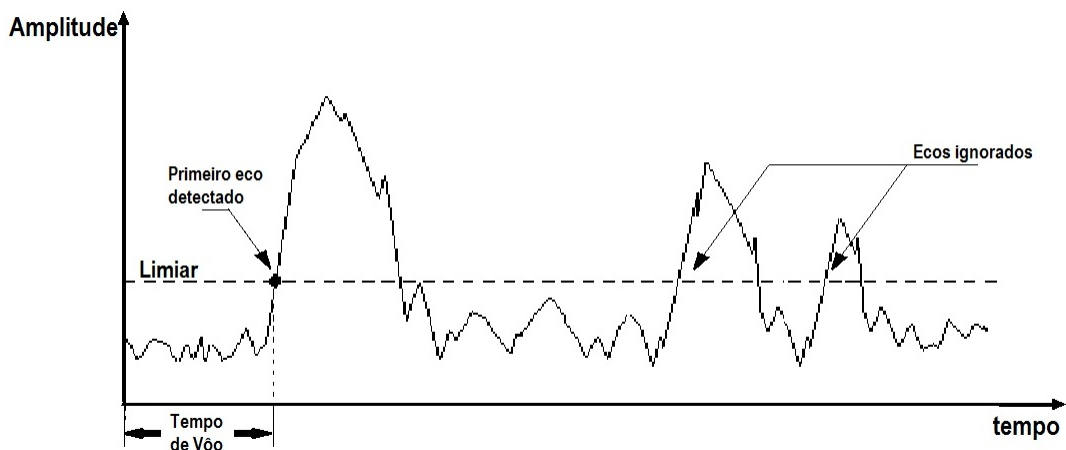


Figura 3.4: Gráfico das Amplitudes dos Ecos no tempo



Figura 3.5: Sensor Polaroid Série 6500

3.1.3 Funcionamento do sensor Polaroid

O acionamento do sensor Polaroid Série 6500 (Figura 3.5) é feito por meio de um sinal retangular em um terminal de entrada denominado INIT. Ao perceber uma transição de subida de nível de tensão neste terminal, o sensor inicia o processo de envio de feixe de ondas ultra-sônicas ao ambiente. O sinal INIT gerado pelo microcontrolador permanece 50 ms em nível baixo (aproximadamente 0V) e depois muda para o nível alto (aproximadamente 5V). O sensor, ao receber de volta as ondas ultra-sônicas refletidas pelo obstáculo, muda o nível de tensão em um terminal de saída denominado ECHO, gerando também um sinal retangular. Os demais ecos são descartados pelo circuito eletrônico do próprio sensor. A informação da distância está presente no tempo decorrido entre as transições de subida dos sinais INIT e ECHO. A subida do sinal ECHO marca o fim de um ciclo de medição. Neste momento, o PIC faz com que o sinal INIT retorne ao nível baixo. Assim, o sinal INIT tem seu tempo de nível baixo (50 ms) fixo, mas o de nível alto variável. Outro sinal de entrada, denominado BINH, é necessário para que o sensor meça distâncias abaixo de 50 cm. Este sinal é também retangular e tem sua subida um milissegundo após a subida do sinal INIT e retorna ao nível baixo no mesmo momento que este. Após atingirem o nível alto, os sinais INIT e BINH são mantidas até que o microcontrolador perceba a subida do sinal ECHO. O sinal ECHO é mantido alto até que uma nova subida no sinal INIT seja percebida pelo sensor. As características desses sinais são mostradas na Figura 3.6. Este sensor é alimentado com +5 V e atinge correntes de pico da ordem de 2 A, no momento em que o feixe de ondas ultra-sônicas é emitido.

3.1.4 Funcionamento do sensor HG-M40DAII

O acionamento dos sensores HG-M40DAII (Figura 3.7) também é feito por meio de uma onda retangular no pino de entrada denominado INIT. Este sinal, porém, fica em nível alto durante 1,9 ms e 20 ms em nível baixo, sempre. O sensor, ao receber de volta as ondas

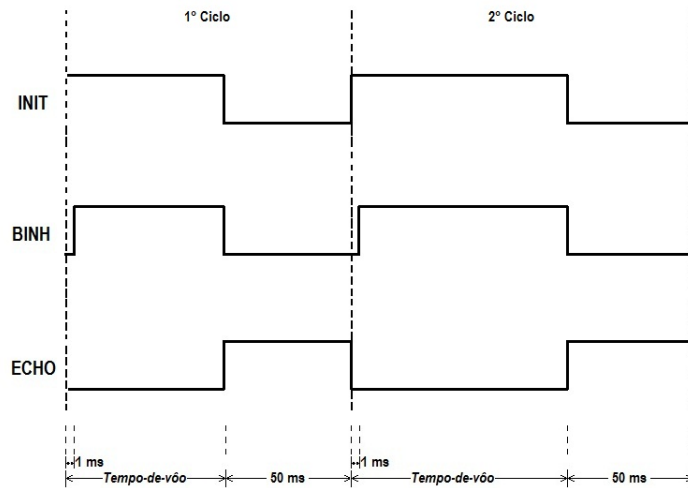


Figura 3.6: Diagrama de sinais do acionamento do sensor Polaroid 6500



Figura 3.7: Sensor HG-M40DAII

refletidas pelo obstáculo, muda o nível de tensão de um pino de saída também denominado ECHO, gerando um sinal retangular. O sinal ECHO mostra todos os ecos que apresentam amplitude maior que o limiar. A informação de distância está no tempo decorrido entre a subida da onda INIT e da primeira subida após esta do sinal ECHO. As características desses sinais são apresentadas na Figura 3.8. Este sensor é alimentado com 12 V e somente estes são os sinais necessários para que ele funcione. Neste subsistema são usados 7 destes sensores, instalados em locais estratégicos do veículo. Para evitar interferência de um sensor na medição do outro, eles funcionam em revezamento. Somente um sensor mede de cada vez.

3.1.5 Circuito de alimentação dos sensores

O circuito de alimentação dos sensores está montado em uma matriz de contatos. Neste circuito está presente o regulador de tensão LM323K[LM323K]. Este regulador foi necessário para baixar a tensão proveniente da bateria do automóvel de 12 V para 5 V, para o acionamento do Sensor Polaroid, que também necessita de uma corrente de 2 A de pico durante o tempo em que estiver funcionando. Foi interessante o seu uso porque as medições atingiam frequências maiores que 10 Hz e os outros testados não ofereciam o suporte necessário. Os demais sensores e a placa à qual está conectada o microcontrolador são alimentados com 12 V.

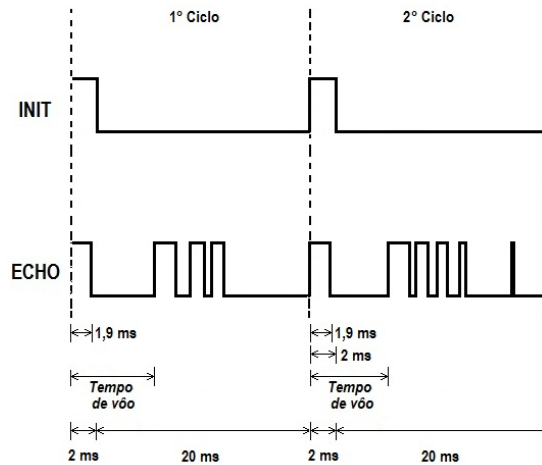


Figura 3.8: Diagrama de sinais do acionamento dos sensores HG-M40DAII

Os sensores HG-M40DAII funcionam em revezamento para evitar que o feixe de ondas emitido por um deles seja percebido por algum outro. Desta forma, foi necessária uma multiplexação no tempo para que todos pudessem medir de forma confiável. Então, enquanto um está medindo, todos os outros do mesmo modelo não estão sendo alimentados. O microcontrolador é responsável por gerar sinais que abrem e fecham chaves eletrônicas (MOSFETs¹ IRF630 [IRF630]) para que os sensores sejam acionados um de cada vez. Os terminais de alimentação positivos dos sensores estão ligados a um barramento de 12 V e os de referência estão ligados ao terminal dreno dos MOSFETs. Os terminais fonte dos MOSFETs estão ligados ao terminal de referência do circuito. Os terminais porta estão ligados ao PIC, que geram um sinais que "ligam" e "desligam" os sensores nos momentos apropriados, como mostrados na Figura 3.9.

3.1.6 Circuito de Acionamento e leitura dos Sensores

O microcontrolador gera os sinais INIT e BINH para acionar o sensor Polaroid através dos terminais configurados como sendo de saída RA4 e RA5. Estes sinais são transmitidos ao sensor através de fios que os ligam à placa de circuito impresso em que se encontra conectado o PIC. Ao receber o feixe de ondas refletidas pelo obstáculo, o sensor muda o nível de tensão do sinal de resposta ECHO. Este sinal também é transmitido através um fio ligado entre o terminal ECHO do sensor e o pino configurado como entrada RB7 do PIC. Há ainda um resistor de resistência 10 K Ohm ligado entre os terminais ECHO e de alimentação +5V, que funciona como resistor de polarização.

O terminal de INIT de todos os sensores HG-M40DAII está ligado direto ao pino RA3

¹Metal Oxide Semiconductor Field Effect Transistor

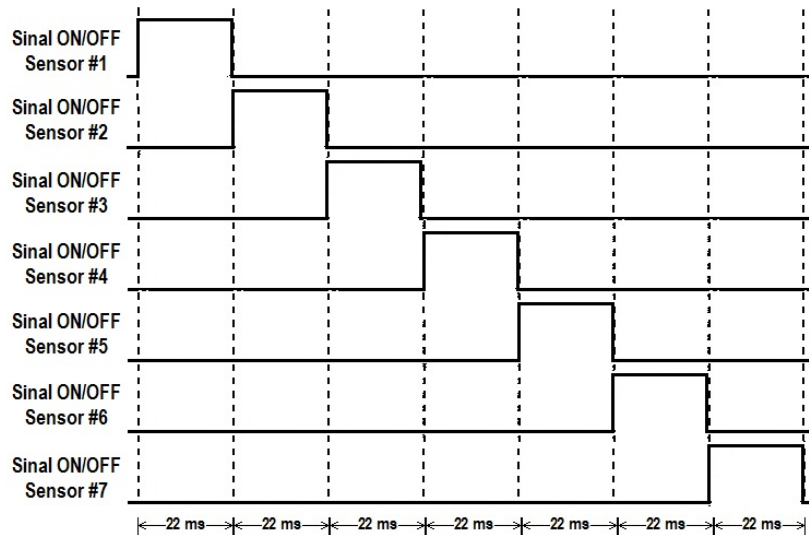


Figura 3.9: Diagrama de sinais gerados pelo PIC para o chaveamento dos sensores

responsável por gerar o sinal através do barramento. Todos estes, durante todo o tempo em que o sistema estiver acionando os sensores HG-M40DAII, estarão recebendo este sinal, mesmo que não estejam sendo alimentados. Devido à forma como foi montado o circuito de alimentação e a forma de operar dos próprios sensores, o terminal de ECHO destes apresenta nível de tensão de +12 V enquanto a chave eletrônica usada para interromper a alimentação estiver aberta. O nível de tensão de +12 V não pode ser ligado ao terminal de entrada do PIC, que suporta até +5 V. Para evitar que isso ocorresse, foi montado um circuito similar ao de alimentação, usando MOSFETs funcionando como chaves eletrônicas. Quando a chave está fechada, a diferença de potencial entre o terminal fonte e a referência varia ora sendo 0 V, ora 5 V. Para abrir e fechar estas chaves, são usados os mesmos sinais "liga-desliga" usados na alimentação. O sinal ECHO medido entre os terminais ECHO do sensor e de referência do circuito é de número 1 mostrado na Figura 3.11.

Os circuitos de alimentação e acionamento dos sensores é mostrado na Figura 3.10.

Assim, em cada fatia de tempo, um sensor tem seu circuito de alimentação fechado e o seu sinal de resposta ECHO, que será ou 0 V ou 5 V, estará sendo transmitido ao PIC. Com esta montagem, tem-se que o sinal ECHO do sensor que tiver o seu sinal de acionamento em nível baixo, estará em 0 V. Desta forma, e como somente um sinal ECHO varia a cada ciclo, pode ser feita uma operação de "OU LÓGICO" dos sinais dos sensores HG-M40DAII sem que haja perda de informação. O sinal resultante desta operação pode ser mostrado na Figura 3.11.

Tem-se que o sinal ECHO de cada sensor apresenta o resultado do multi-eco. Assim, para um mesmo pulso do sinal INIT que o sensor recebe, são retornados vários pulsos de ECHO. A informação de distância ao obstáculo mais próximo ao sensor pode ser obtida através do tempo decorrido entre a subida de nível do sinal INIT e a primeira subida de nível do sinal ECHO daquele sensor naquela faixa de tempo. O restante do sinal pode ser desprezado porque não

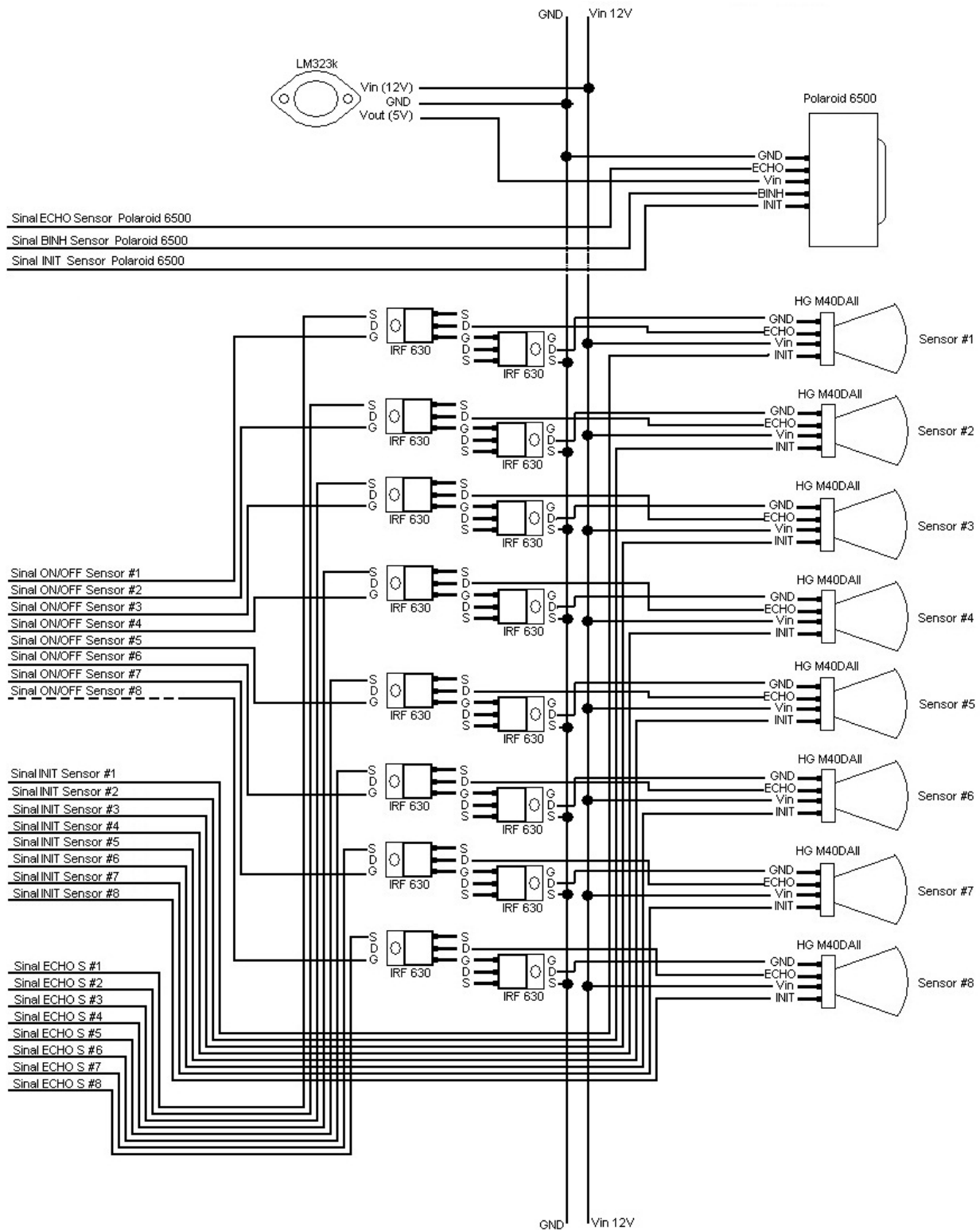


Figura 3.10: Circuito de alimentação e acionamento dos sensores

apresenta informação útil a este trabalho.

Assim, o sinal que contém informação útil compreende-se entre a subida do sinal INIT e a primeira subida do sinal ECHO. Para que o restante do sinal não causasse problemas quanto ao processamento da informação de distância, diversas soluções que objetivavam desprezá-lo ou anulá-lo foram estudadas e testadas. A solução implementada consiste na operação de "E

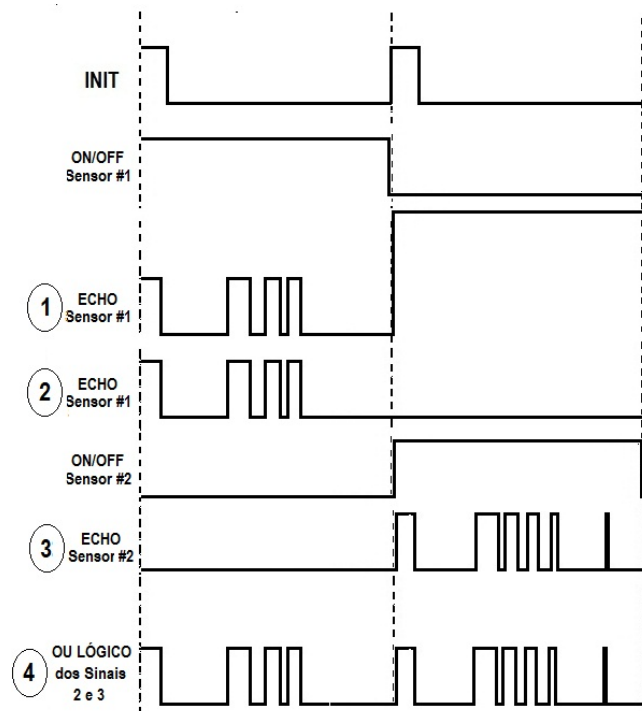


Figura 3.11: Diagrama de sinais que mostra a operação de OU LÓGICO feita com os sinais dos sensores HG-M40DAII

LÓGICO"do sinal ECHO com um sinal CONTROL que também é gerado pelo PIC. O sinal CONTROL tem nível alto do instante que o sinal INIT desce até que a primeira subida do sinal ECHO seja detectada pelo PIC. Neste momento, o sinal CONTROL muda seu nível para 0 V, fazendo com que a operação lógica resulte em um sinal de 0 V no terminal de entrada do microcontrolador. Dessa forma, é possível que o PIC detecte apenas a primeira subida do sinal ECHO e possa medir o tempo decorrido entre as subidas dos sinais INIT e ECHO. A Figura 3.12 mostra um diagrama de sinais referente a esta operação do sinal. A seguir, a Figura 3.13 traz um esquema da montagem feita no circuito que representa as operações de E e OU LÓGICAS dos sinais.

Assim, a informação de qual sensor está ativo em qual fatia de tempo e a respectiva informação de distância a algum possível obstáculo é encontrada no microcontrolador. Esta montagem foi realizada com o intuito de que o sistema seja facilmente expansível. Caso seja necessária a inserção de um novo sensor do mesmo modelo, seria necessário montar o pequeno circuito de chaveamento dos sinais e apenas que o microcontrolador gerasse o sinal de acionamento apropriado para que este operasse da mesma forma que os outros.

3.1.7 Funcionamento do programa executado no PIC

Foi desenvolvido um programa que tem como objetivos:

1. Gerar os sinais INIT e BINH para o sensor Polaroid 6500;

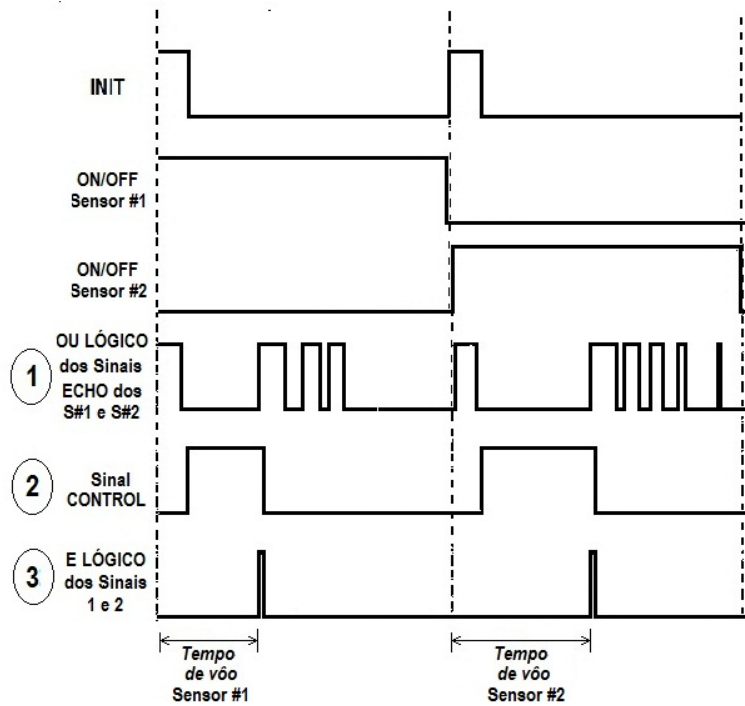


Figura 3.12: Diagrama de sinais que mostra a operação de E LÓGICO feita com os sinais dos sensores HG-M40DAII e o sinal CONTROL

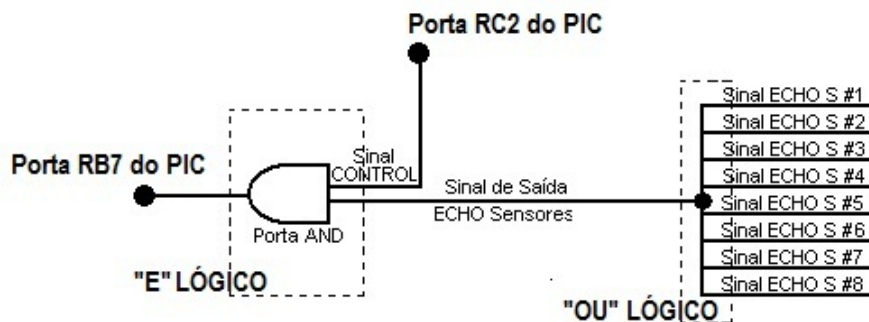


Figura 3.13: Diagrama que mostra o circuito eletrônico que realiza as operações lógicas com os sinais

2. Perceber quando o sinal ECHO apresentar transições de subida;
3. Medir o tempo decorrido entre as subidas do sinal ECHO e INIT e enviar ao computador central essa informação.
4. Gerar os sinais responsáveis pelo revezamento dos sensores HG-M40DAII, para que apenas um seja alimentado e tenha seu sinal ECHO sendo transmitido em cada período de tempo;
5. Gerar o sinal INIT para estes sensores.
6. Gerar o sinal CONTROL para que somente a primeira subida do sinal ECHO do sensor que estiver sendo acionado naquele período chegue ao microcontrolador;
7. Perceber quando o sinal ECHO apresentar transições de subida;

8. Medir o tempo decorrido entre as subidas do sinal ECHO e INIT e enviar ao computador central essa informação.

O terminal configurado como fonte dos sinais INIT para os sensores HG-M40DAII é o RA3. O sinal resultante das operações é percebido pelo terminal RB7. O sinal CONTROL é gerado no RC2. Para os sinais INIT, BINH e ECHO do sensor Polaroid 6500, foram configurados os terminais RA4, RA5 e RB6.

Para que o desvio da execução do programa quando há uma interrupção possa ser o mais breve possível, tem-se apenas atribuições de valores a poucas variáveis no seu fluxo de tratamento. As interrupções internas são geradas pelos contadores usados no auxílio às rotinas de geração dos sinais INIT e BINH para os sensores. As interrupções externas são geradas pelos sinais ECHO, que são as respostas dos sensores. A rotina de tratamento de interrupções é ilustrada pelo algoritmo a seguir:

Início da rotina de tratamento de interrupções

Se o motivo da interrupção for overflow do timer de geração do Sinal INIT

Então INITGenerationSignalTimerOverflow = Verdadeiro;

Se o motivo da interrupção for overflow do timer de geração do Sinal BINH

Então BINHGenerationSignalTimerOverflow = Verdadeiro;

Se o motivo da interrupção for a resposta do sensor Polaroid

Então PolaroidSensorRespond = Verdadeiro;

Se o motivo da interrupção for a resposta de um dos sensores HG-M40DAII

Então HGM40daIISensorRespond = Verdadeiro;

Fim da rotina de tratamento de interrupções

O programa desenvolvido para o PIC comunica-se com o computador central a partir da emulação de uma porta serial por meio de uma porta USB. O computador envia dados que são traduzidos em comandos para que seja efetuada troca de estados do sistema. O microcontrolador envia a confirmação de recebimento desses dados. Os outros dados que o microcontrolador envia são referentes aos tempos decorridos entre as subidas dos sinais INIT e ECHO do sensor que estiver sendo acionado naquele instante de tempo.

O programa é executado por meio de uma máquina de estados (mostrada na Figura 3.14) contendo os seguintes estados IDLE, CONNECTION, RUNNING, MEASURING, WATCH-PARKING e TERMINATE em que o microcontrolador executa funções distintas.

A rotina principal inicia-se no estado IDLE e são efetuadas a configuração dos terminais de entrada e saída e dos temporizadores do microcontrolador através das funções *ConfigureIOPorts* e *ConfigureTimers*. Feito isso, a execução do programa vai para o estado CONNECTION e iniciam-se as operações responsáveis por efetuar a conexão com o computador.

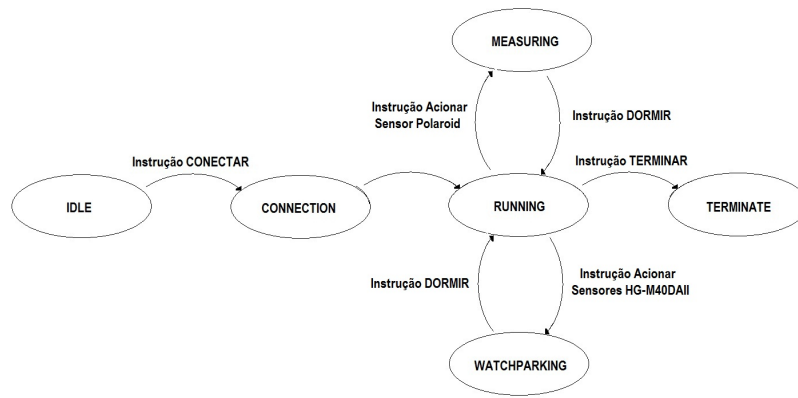


Figura 3.14: Máquina de Estados do funcionamento do programa do microcontrolador

A função *EstablishConection* está dentro de um laço que só é finalizado quando a conexão é estabelecida. Nela, após recebido o comando de conexão vindo do computador, é enviado de volta uma resposta a esse comando. A conexão entre o microcomputador central e o microcontrolador está, então, estabelecida.

Depois de estabelecida a conexão, a execução vai para o estado RUNNING. Um novo laço inicia-se por executar a função *ReadComputerNewStateChangeCommand*, que lê algum comando para mudança de estado que, porventura, tenha sido enviado pelo computador central. Após isso, são executadas instruções distintas, conforme o estado em que a execução se encontra.

Os outros estados possíveis são: MEASURING e WATCHPARKING, que são aqueles em que o microcontrolador estará acionando um dos dois tipos de sensores e lendo as suas respectivas respostas; e TERMINATE, que leva à finalização da execução do programa.

O algoritmo a seguir mostra a execução simplificada do fluxo principal do programa do PIC:

```

Início da rotina principal
Estado = IDLE;
ConfigureIOPorts(); (Configura os terminais de entrada e saída do microcontrolador)
ConfigureTimers(); (Configura os temporizadores usados para a geração de sinais)
Estado = CONNECTION;
Enquanto Estado é igual a CONNECTION faça
    EstablishConection(); (Tenta a conexão com o computador)
Fim Enquanto
Enquanto Estado é diferente de TERMINATE faça
    ReadComputerNewStateChangeCommand() (Lê o que foi enviado pelo Tablet PC)
    Caso Estado é igual a RUNNING faça
        DoNothing();
    Fim Caso
    Caso Estado é igual a MEASURING faça
        DoMeasuringActions();
  
```

```

Fim Caso
Caso Estado é igual a WATCHPARKING faça
    DoWatchParkingActions();
Fim Caso
Fim Enquanto
CloseConnection()
Fim da rotina principal

```

As instruções presentes nos estados se resumem na leitura dos valores das variáveis auxiliares à manipulação dos sinais INIT, BINH ou CONTROL, dependendo do estado; na leitura da variável responsável por avisar o acontecimento da subida do sinal ECHO do sensor que está sendo acionado naquele momento; e no conseqüente processamento e envio da informação do tempo de vôo.

No estado RUNNING, o microcontrolador não está acionando nenhum dos sensores. O microcontrolador apenas fica aguardando o envio de algum comando de mudança de estado vindo do computador central.

No estado MEASURING, o microcontrolador está acionando o sensor Polaroid. São gerados os sinais INIT e BINH e é percebida a subida do sinal ECHO. É obtido e enviado ao microcomputador o tempo decorrido entre as subidas dos sinais INIT e ECHO para que seja calculada a distância ao obstáculo. Os dados são enviados continuamente ao computador que os armazena em uma área de memória temporária até que eles sejam lidos, sem haver um mecanismo de confirmação de recebimento.

O algoritmo a seguir ilustra os comandos principais executados neste modo:

```

Início da Função DoMeasuringActions
Se INITGenerationSignalTimerOverflow = Verdadeiro
    Então ManipulateINITSignal()
Se BINHGenerationSignalTimerOverflow = Verdadeiro
    Então ManipulateBINHSignal()
Se PolaroidSensorRespond = Verdadeiro
    Então ReadTimeOfFlyghtAndSendToComputer()
Fim da Função DoMeasuringActions

```

A função *ManipulateINITSignal* é responsável por coordenar o nível lógico aplicado ao terminal de saída do microcontrolador, gerando o sinal INIT para acionar o sensor Polaroid. É responsável também por disparar a contagem do temporizador quando há a subida de nível de tensão do sinal INIT. A função *ManipulateBINHSignal* é responsável por coordenar o nível de tensão aplicado ao terminal de saída do microcontrolador, gerando o sinal BINH. A função *ReadTimeOfFlightAndSendToComputer* é responsável por parar o temporizador, ler o tempo (em número de pulsos do relógio interno ao microcontrolador) é indicado por ele neste

momento e enviá-lo ao computador para que seja aferida a distância ao obstáculo.

No estado WATCHPARKING, o microcontrolador está acionando os sensores HG-M40DAII. Neste estado, é feito o controle de qual sensor está sendo acionado em cada fatia de tempo. São gerados os sinais necessários à multiplexação do acionamento dos sensores, o sinal INIT e CONTROL e é percebida a subida do sinal ECHO vinda do sensor que está sendo acionado naquele instante. É obtido e enviado ao microcomputador o tempo decorrido entre as subidas dos sinais INIT e ECHO para que seja calculada a distância ao obstáculo.

O algoritmo a seguir ilustra os comandos principais executados neste modo:

```

Início da Função DoWatchParkingActions
  Se INITGenerationSignalTimerOverflow = Verdadeiro
    Então ManipulateINITAndCONTROLSignals()
    CoordinateMultiplexationSignals()
  Fim Se
  Se HGM40daIISensorRespond = Verdadeiro
    Então TurnOFFCONTROLSignal()
    ReadTimeOfFlyghtAndSendToComputer()
  Fim Se
Fim da Função DoWatchParkingActions

```

A função *ManipulateINITAndCONTROLSignals* é responsável por coordenar o nível lógico aplicado a dois terminais de saída do microcontrolador, gerando os sinais INIT e CONTROL para acionar e ler a resposta do sensor HG-M40DAII que estiver ativo neste momento. É responsável também por disparar a contagem do temporizador quando há a subida de nível de tensão do sinal INIT. A função *CoordinateMultiplexationSignals* é responsável por coordenar qual dos sete terminais de saída estará com nível alto, gerando o efeito de multiplexação dos sensores HG-M40DAII. A função *TurnOFFCONTROLSignal* é responsável por colocar o terminal de saída do sinal CONTROL em nível baixo, já que o primeira transição de subida do sinal ECHO foi detectada. A função *ReadTimeOfFlyghtAndSendToComputer* é responsável por parar o temporizador, ler o tempo (em número de pulsos do relógio interno ao microcontrolador) é indicado por ele neste momento e enviá-lo ao computador para que seja aferida a distância ao obstáculo. Ao se enviar o valor do tempo medido também é enviada ao computador a informação de qual sensor está sendo acionado neste momento.

Este laço que só termina quando o comando que leva a execução para o estado TERMINATE é enviado pelo computador.

3.1.8 Informações Gerais sobre o Subsistema de Medição de Distância aos Obstáculos

As medições de distância para a montagem do mapa da vaga são feitas usando-se somente o sensor Polaroid. Este está instalado próximo à roda dianteira direita do veículo (Figura 3.15). A frequência de tomadas de distância é variável. Isto é, assim que o sensor sobe o nível do sinal ECHO conclui-se um ciclo, os dados são enviados ao computador e um novo ciclo se inicia. Note que, dessa forma, para objetos posicionados perto do sensor tem-se uma frequência de tomadas de distância maior que para o caso de o objeto estar posicionado mais longe. Objetivou-se, com isto, que o maior número de medidas fosse feito, resultando na maior resolução possível para o escaneamento da vaga.



Figura 3.15: Local de instalação do sensor Polaroid 6500

No caso dos sensores HG-M40DAII, o ciclo de acionamento e leitura da resposta do sensor apresenta um tempo fixo. Cada sensor faz, aproximadamente, 3 medições por segundo, intercaladas com as medições dos outros sensores. Em ambos os casos, os dados são enviados continuamente e armazenados em uma área de memória temporária no computador, a cada leitura.

O computador faz a leitura desses dados sem nenhum tipo de mecanismo de sincronização entre a chegada de um dado na região de memória intermediária e a leitura desse dado. A forma de sincronismo existente é baseada em início e fim de medições, sendo que o computador envia para o microcontrolador as instruções de mudanças de modo. Assim, são acionados continuamente os sensores HG-M40DAII ou continuamente o sensor Polaroid, até que haja a mudança de modo.

O número 1 indica o local de instalação do sensor Polaroid Serie 6500. Os números de 2 a 8 indicam onde foram instalados os sensores HG-M40DAII. Estes foram instalados ao redor do veículo em pontos que pudessem auxiliar o sistema a detectar possíveis obstáculos próximos ao carro, conforme ilustrado pela Figura 3.16. Os sensores HG-M40DAII fazem tomadas de distância enquanto o carro se movimenta seguindo a trajetória traçada. Caso seja medida alguma distância que possa colocar em risco o veículo, uma instrução para que o carro acione o freio completamente é executada. Não foi objetivo instalá-los para cobrir os chamados

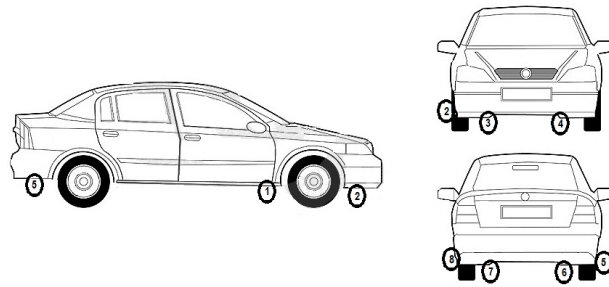


Figura 3.16: Diagrama do local de instalação dos sensores

pontos-cegos, para o caso de auxiliar um motorista a executar a manobra.

A Figura 3.17 mostra os locais em que foram instalados os sensores no veículo.



Figura 3.17: Locais de instalação dos sensores no veículo

3.2 Subsistema de medição de velocidade

A velocidade do carro é medida com base no sinal do sensor de velocidade do sistema de freios ABS da roda dianteira direita. Este sinal é tratado e, com o auxílio de um microcontrolador PIC, são obtidos os dados necessários para se conhecer a velocidade do automóvel. Os dados são enviados para o computador central via comunicação USB para que seja feito o cálculo.

Este subsistema tem por objetivos:

1. Realizar medições para que seja possível medir o tamanho da vaga encontrada para a realização dos cálculos necessários à geração da trajetória.
2. Realizar medições para que o posicionamento do veículo possa ser estimado enquanto este realiza a manobra do estacionamento.

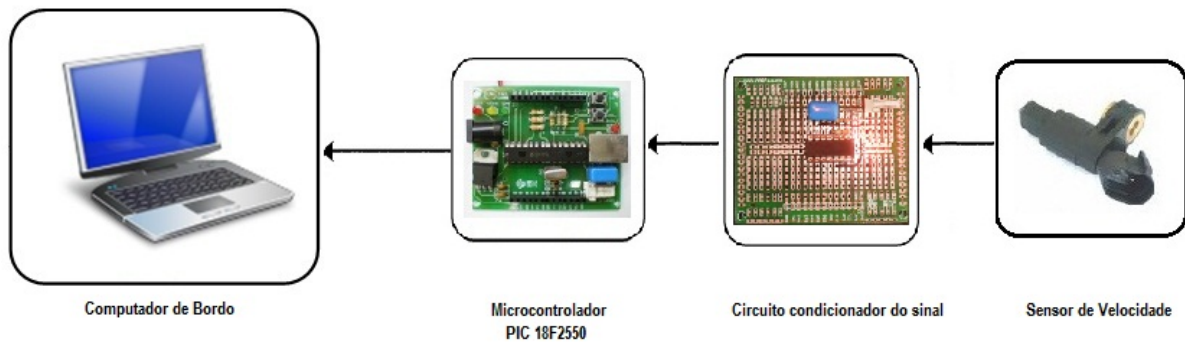


Figura 3.18: Subsistema de medição de velocidade

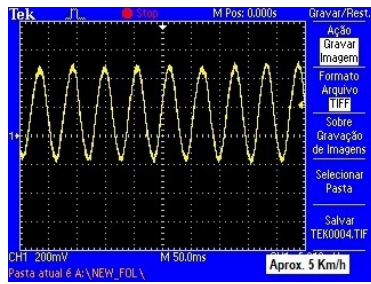
Uma figura esquemática desse subsistema é mostrada na Figura 3.18. Ele é composto por:

- 1 placa de circuito impresso em que está montado o microcontrolador PIC 18F2550, responsável por comunicar ao computador as informações capturadas.
- 1 placa de circuito impresso em que está montado o circuito condicionador do sinal do sensor de velocidade.
- Circuito condicionador do sinal do sensor de velocidade propriamente dito.

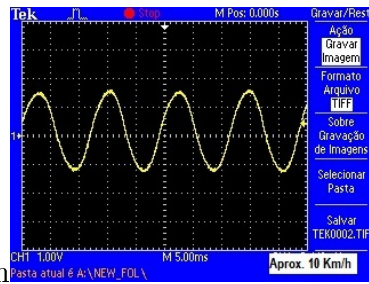
3.2.1 Funcionamento do sensor de velocidade

O sistema de freios ABS utiliza sensores de relutância magnética variável. O sinal resultante da medição é uma senóide em que amplitude e frequência variam com a variação da velocidade do automóvel. A frequência da onda é proporcional à velocidade. Foi usado um osciloscópio digital portátil para coletar este sinal enquanto o veículo se movimentava. Os sinais capturados, juntamente com a informação de velocidade mostrada no velocímetro do veículo foram usados para uma primeira calibração deste subsistema. A Figura 3.19 mostra os sinais e as correspondentes velocidades do carro no momento em que foram obtidos:

Para se estimar a velocidade, mediu-se o período de uma onda e, a partir dele, foram feitos os cálculos necessários.

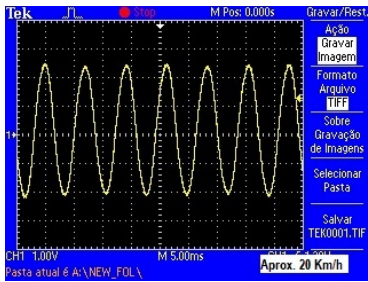


(a)

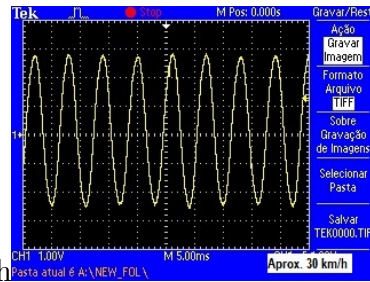


(b)

Aprox. 10 Km/h



(c)



(d)

Aprox. 30 Km/h

Figura 3.19: Sinais coletados a partir do sensor de velocidade do sistema de freios ABS

3.2.2 Circuito condicionador do Sinal

Para o tratamento do sinal foi utilizado um filtro passivo e um comparador, objetivando retirar o ruído presente no sinal medido e transformá-lo em uma onda retangular. O filtro, formado por um capacitor e um resistor, tem frequência de corte de aproximadamente 200 Hz. Como a informação que era buscada neste sinal útil a este trabalho estava presente nas frequências de até esse valor, esta montagem foi suficiente. O comparador é o circuito integrado LM339 [LM339].

O sinal filtrado é passado por um circuito comparador, resultando em uma onda retangular que apresenta nível alto quando a onda apresenta sinal de tensão positivo e nível baixo quando negativo. São comparados os níveis de tensão do sinal vindo do sensor no terminal de entrada positivo do comparador e do terra, no negativo. A Figura 3.20 apresenta o diagrama desse circuito.

A Figura 3.21 apresenta o processamento feito no sinal do sensor. A Figura 3.21a mostra o sinal ruidoso ($V_{\sim} - V_{ref}$). A Figura 3.21b mostra o sinal filtrado ($V_c - V_{ref}$) e a Figura 3.21c mostra o sinal resultante da comparação ($V_{final} - V_{ref}$).

Um microcontrolador PIC 18F2550 processa, então, o sinal retangular obtido pelo condicionamento.

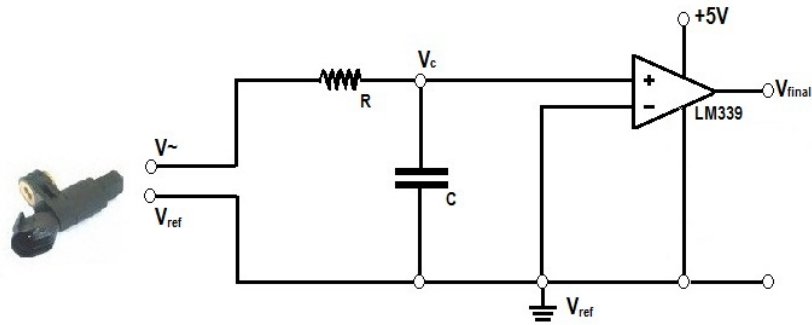


Figura 3.20: Diagrama do circuito de condicionamento do sinal do sensor de velocidade

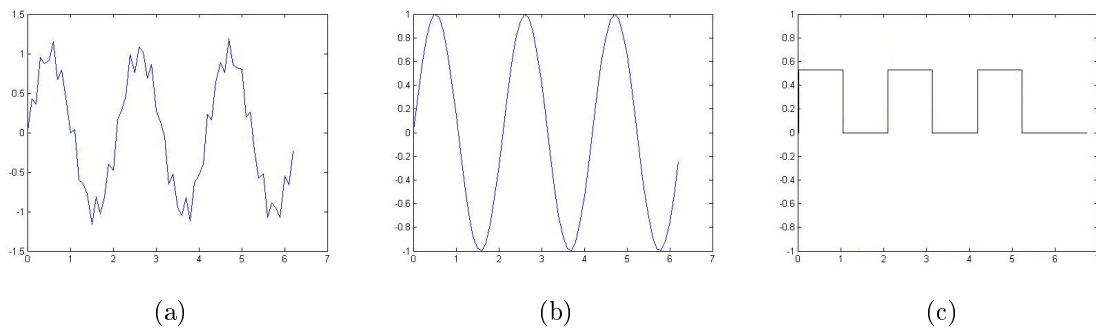


Figura 3.21: Sinais obtidos em diferentes pontos da Figura 3.20

3.2.3 Programa do microcontrolador

São utilizadas as funcionalidades de interrupção externa do microcontrolador e de temporizadores. Através da interrupção externa, é possível obter a informação de início de um período na onda do sinal do sensor de velocidade. Através do temporizador, é possível obter o tempo decorrido entre dois inícios de período consecutivos.

O computador envia o caractere que é traduzido no comando para que seja efetuada a leitura de um período. O microcontrolador envia a confirmação de recebimento desse dado e inicia a medição. Após medir, envia também o valor medido para o computador e aguarda outra requisição para medir novamente.

O programa é executado por meio de uma máquina de estados contendo os estados IDLE, CONNECTION, RUNNING, MEASURING e TERMINATE em que o microcontrolador executa funções distintas.

As instruções executadas para efetuar a conexão são iguais às do programa do sistema de medição de distância aos obstáculos.

Após estabelecida a conexão, a execução vai para o estado RUNNING. Um novo laço inicia-se por executar a função *ReadComputerNewStateChangeCommand*, que lê algum comando para mudança de estado que, porventura, tenha sido enviado pelo computador central.

Os outros estados possíveis são: MEASURING, em que o microcontrolador estará processando o sinal do sensor de velocidade; e TERMINATE, que leva ao final da execução do programa.

O algoritmo a seguir mostra a execução simplificada do fluxo principal do programa do PIC:

```

Início da rotina principal
Estado = IDLE;
ConfigurePorts(); (Configura os terminais de entrada do microcontrolador)
ConfigureTimers(); (Configura os temporizadores usados)
Estado = CONNECTION;
Enquanto Estado é igual a CONNECTION faça
    EstablishConection(); (Tenta a conexão com o computador)
Fim Enquanto
Enquanto Estado é diferente de TERMINATE faça
    ReadComputerNewStateChangeCommand()
    Caso Estado é igual a RUNNING faça
        DoNothing();
    Fim Caso
    Caso Estado é igual a MEASURING faça
        DoMeasuringActions();
    Fim Caso
Fim Enquanto
CloseConnection()
Fim da rotina principal

```

No estado RUNNING, o microcontrolador não está processando o sinal do sensor. A execução se resume em apenas aguardar o envio de algum comando de mudança de estado vindo do computador central.

No estado MEASURING, o microcontrolador está processando o sinal do sensor de velocidade. É obtido e enviado ao microcomputador o tempo decorrido entre as subidas consecutivas do sinal para que seja calculada a velocidade do carro. Diferentemente dos sensores de ultra-som, o dado é enviado uma única vez para cada solicitação de mudança de estado, caracterizando uma leitura sob requisição. Dessa forma, configura-se a execução como sendo uma forma de leitura sob requisição.

O algoritmo a seguir ilustra os comandos principais executados neste modo:

```

Início da Função DoMeasuringActions
ClearTimer()
Enquanto INITGenerationSignalTimerOverflow é diferente de Verdadeiro faça
    DoNothing()
Fim Enquanto
InitTimer()

```

```

INITGenerationSignalTimerOverflow = Falso
Enquanto INITGenerationSignalTimerOverflow é diferente de Verdadeiro faça
    DoNothing()
Fim Enquanto
StopTimer()
ReadTimerAndSendToComputer()
Estado = RUNNING
Fim da Função DoMeasuringActions

```

A função *ReadTimerAndSendToComputer* é responsável por parar o temporizador, ler o tempo indicado por ele neste momento e enviá-lo ao computador para que seja aferida a velocidade do carro.

O valor da variável *INITGenerationSignalTimerOverflow* é mudado para verdadeiro na função de tratamento de interrupções, toda vez que a subida do sinal vindo do sensor é detectada. Dessa forma, é possível se medir o valor de um período.

3.3 Montagem do mapa

Através dos dados fornecidos pelos dois subsistemas apresentados, foi possível construir um mapa da localização dos obstáculos em relação ao veículo. Para cumprir essa tarefa, as informações foram consolidadas em um programa executado no computador central. Uma das funcionalidades desse programa é ler os dados enviados pelos microcontroladores e transformá-los em informações úteis.

Esse programa executa a leitura dos dados e calcula a velocidade com intervalos de aproximadamente 50 ms. Como no computador central não é usado um sistema operacional de tempo real, esse intervalo era variava em torno desse valor. Esta leitura é feita através de uma rotina dedicada.

A leitura dos dados de distância aos obstáculos é executada em outra rotina. As medições são feitas com o veículo em movimento. As informações buscadas são: o momento (1) em que a distância acusada pelo sensor deixa de ser em relação ao carro parado atrás da vaga e passa a ser em relação à calçada; o momento (2) em que a distância medida deixa de ser em relação à calçada e passa a ser em relação ao carro parado em frente à vaga; e a menor distância medida em relação à calçada.

Para se obter o comprimento da vaga, foi calculada a velocidade média com base nos dados gerados pelo subsistema de medição de velocidade entre os momentos (1) e (2). Foi obtido também o tempo decorrido entre a ocorrência desses eventos. Através da multiplicação de velocidade média pelo tempo, foi obtida uma estimativa do comprimento do espaço correspondente a uma possível vaga. O menor valor de distância medido em relação à calçada

foi usado como sendo o valor da profundidade da vaga.

A Figura 3.22 ilustra o processo de medição das distâncias e a geração do mapa do ambiente. A Figura 3.22a ilustra o momento em que o sistema percebe que deixou de medir dados em relação a um automóvel e passou a medir distâncias em relação ao meio-fio. A Figura 3.22b mostra instantes antes de o sistema perceber que deixou de medir dados em relação à calçada e passou a medir distâncias em relação a um carro. A Figura 3.22c ilustra a posição em que o carro deverá parar após os dados para montagem do mapa serem medidos. A Figura 3.22d mostra o mapa gerado a partir dos dados medidos pelos dois subsistemas.

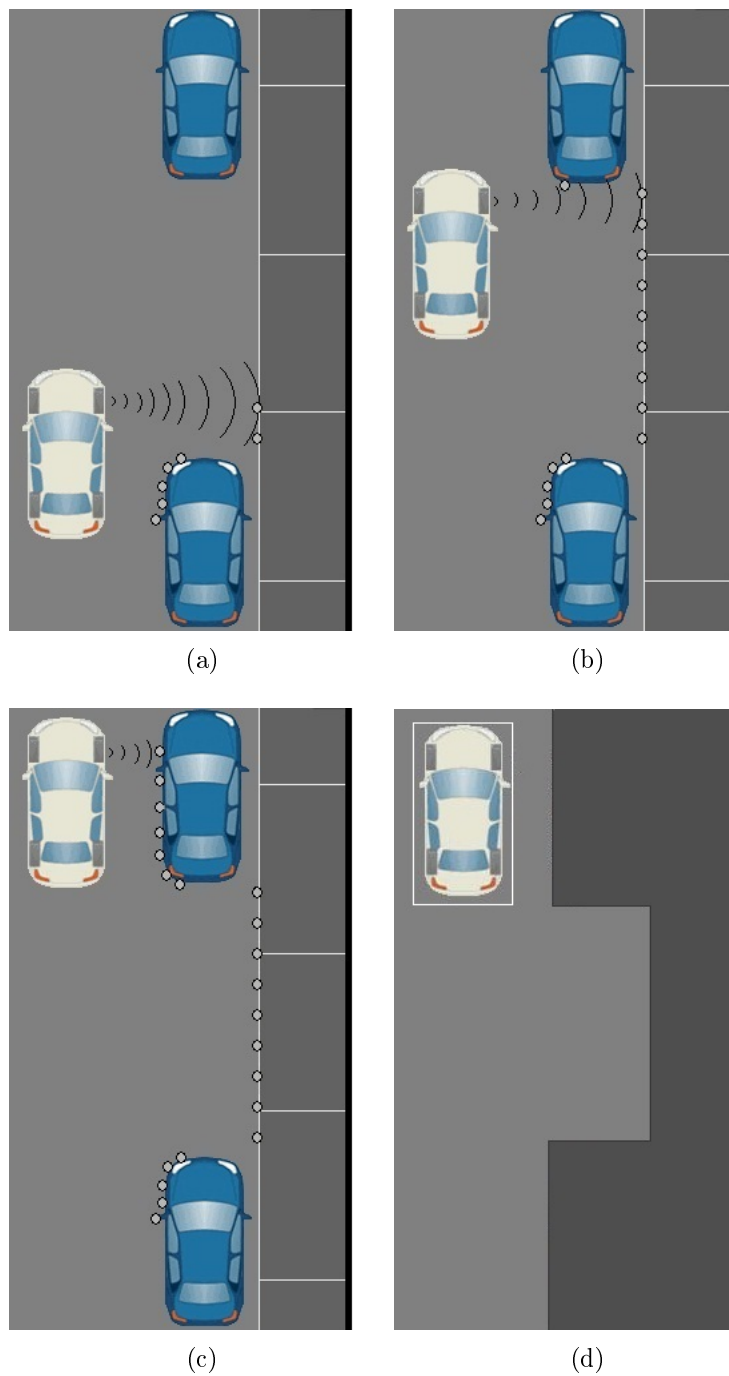


Figura 3.22: Medição das distâncias para geração do mapa

Capítulo 4

Algoritmos de Geração de Trajetórias e de Controle

Um carro é classificado como um robô não-holonômico, ou seja, que apresenta restrições quanto a sua movimentação. Sendo assim, nem todos os caminhos gerados podem ser seguidos por esse robô para estacionar. Outra restrição quanto ao caminho é que ele deve ser facilmente traduzido em comandos para os sistemas de controle embarcados no veículo.

Este capítulo, inicialmente, apresenta um modelo cinemático para o movimento de um carro. Em seguida, é mostrado o algoritmo para geração de trajetórias desenvolvido com base neste modelo. É apresentado também um algoritmo de controle que, a partir dos comandos calculados com base na trajetória, permite que o veículo seja estacionado sem colisão com obstáculos. Por fim, é mostrada a construção do programa executado no computador de bordo que permite a integração dos sistemas de localização da vaga e de atuação do veículo.

4.1 Geração da Trajetória

4.1.1 Modelo de Ackermann

O modelo utilizado neste trabalho é o modelo de Ackerman, descrito em [Choset et al. 2005]. Um veículo é representado por um volume sólido sobre quatro rodas, conforme mostrando na Figura 3.2. As rodas dianteiras são atuadas por meio do volante. A tração do veículo é dianteira e a velocidade dessas rodas é conhecida por meio de sensores. As rodas traseiras são fixas.

Na Figura 4.1, L é a distância entre eixos do veículo, Θ é o ângulo de esterçamento do volante e R é o raio do arco do círculo descrito pelo ponto central do eixo traseiro do veículo quando este se movimenta com o ângulo Θ mantido constante e em baixas velocidades.

O ângulo de esterçamento do volante é obtido por uma roda virtual localizada entre as duas rodas dianteiras, cujo valor é dado pela média dos ângulos das mesmas. A velocidade

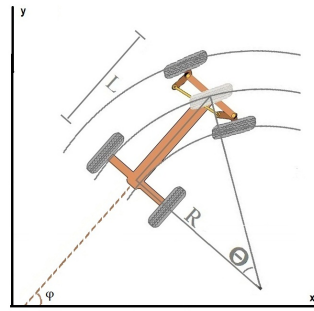


Figura 4.1: Modelo de Ackermann para um carro (adaptado de [Choset et al. 2005])

dessa roda também é dada pela média das velocidades das duas rodas dianteiras.

A orientação do veículo no plano horizontal é dada pela seguinte equação, em que $v(t)$ é a velocidade da roda virtual localizada entre as rodas dianteiras e $\Theta(t)$ é o ângulo de esterçamento do volante.

$$\dot{\varphi}(t) = \frac{v(t)}{L} \text{sen}(\Theta(t)) \quad (4.1)$$

4.1.2 Cálculo da Trajetória

O cálculo da trajetória inicia-se a partir das informações do comprimento e da profundidade da vaga em que se deseja estacionar. É assumido que a posição final do eixo traseiro do veículo coincide com a distância dada pelo comprimento da vaga. O primeiro passo é definir a posição em que o carro estará estacionado ao final da manobra¹. Ao final, a orientação do veículo em relação à calçada será a mesma do instante em que a manobra foi iniciada². A Figura 4.2 mostra as posições do veículo antes e após estacionar. Os retângulos tracejados têm comprimento igual à distância entre eixos e largura igual à largura do veículo, no desenho.

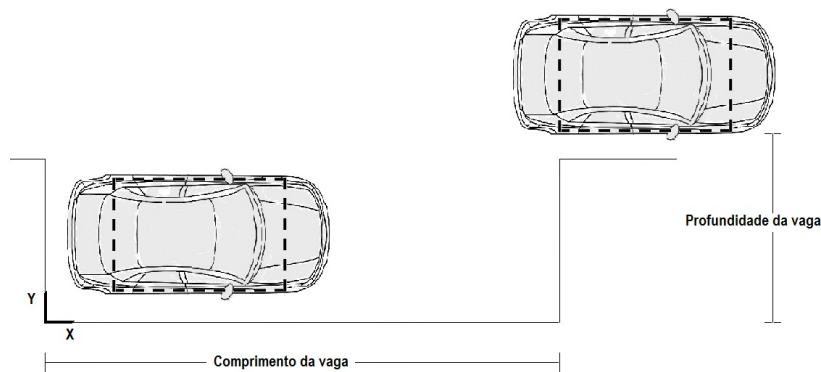


Figura 4.2: Início do cálculo da Trajetória - Definindo posição final do veículo

A partir do desenho esquemático apresentado na Figura 4.2 são estabelecidos dois pontos.

¹Definiu-se que a roda traseira esquerda deverá estar posicionada a 1,5 m do início da vaga e 0,5 m da calçada.

²É assumido que o veículo se encontra paralelo à calçada antes de iniciar a manobra.

Os pontos P1 e P2 são mostrados na Figura 4.3.

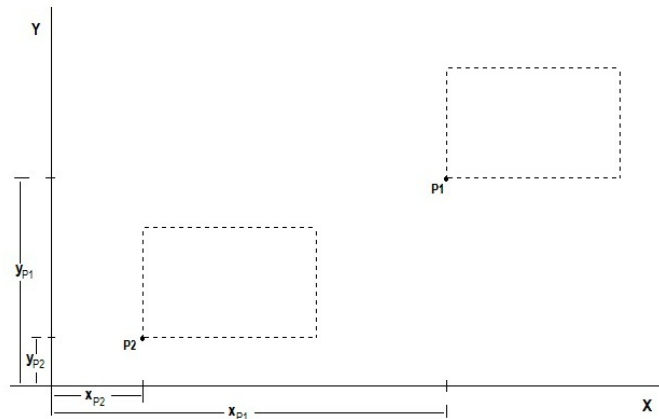


Figura 4.3: Cálculo da Trajetória - Definindo os pontos P1 e P2

Para simplificar o acionamento da direção do veículo na execução da manobra do estacionamento, o resultado do cálculo é uma trajetória formada por arcos de círculos, que serão descritos pelo ponto central do eixo traseiro do veículo. Para a manobra de estacionar em vagas paralelas e à direita do condutor, será necessário que o veículo descreva uma trajetória formada por arcos de dois círculos. Esses arcos são partes de dois círculos que se tangenciam, conforme mostrado na Figura 4.4.

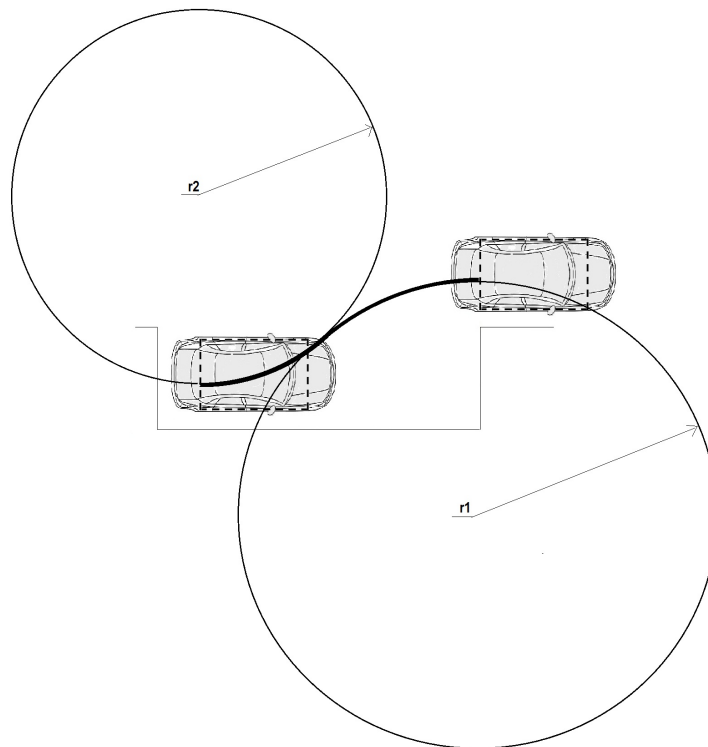


Figura 4.4: Trajetória para o Estacionamento

Para que os raios desses círculos sejam definidos, parte-se, inicialmente, de dois outros círculos de mesmo raio. Os centros dos círculos têm os mesmos valores de abscissas dos pontos

P1 e P2. O ponto P1 faz parte do círculo C1 e o ponto P2 faz parte do círculo C2. A Figura 4.5 mostra os centros dos círculos e o ponto de tangência P3. O ângulo entre o eixo das abscissas e reta tangente às circunferências que passa pelo ponto P3 indica a orientação em relação à orientação inicial que o veículo terá quando parar para reorientar as rodas dianteiras.

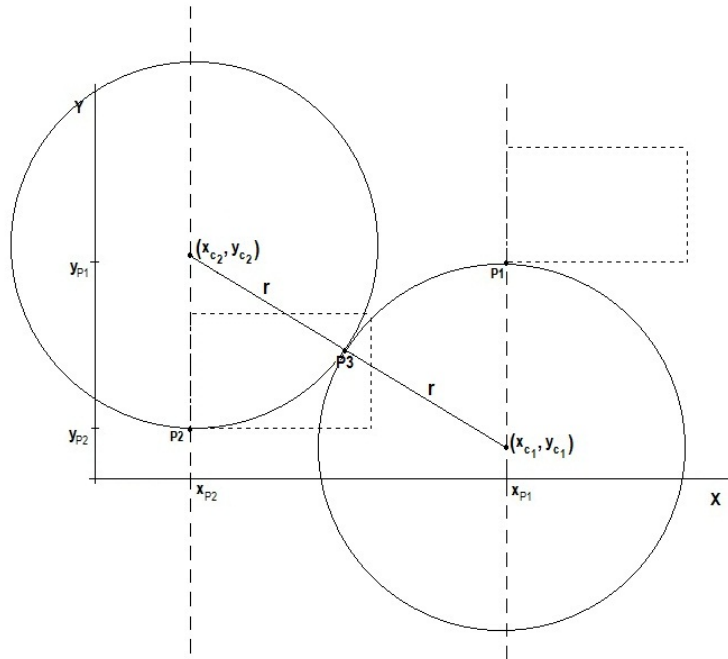


Figura 4.5: Cálculo da Trajetória - Ponto de Tangência

A Figura 4.6 mostra o triângulo retângulo extraído da Figura 4.5 para que seja feito o cálculo do raio das circunferências.

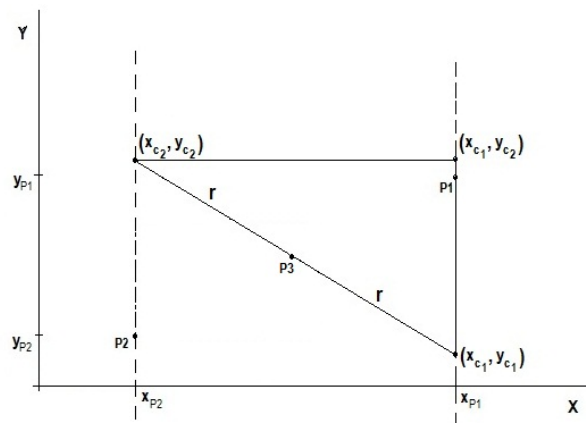


Figura 4.6: Cálculo da Trajetória - Triângulo Retângulo extraído da Figura 4.5

O cálculo do raio r é feito a partir do Teorema de Pitágoras, resultando em:

$$r = \frac{(y_{P2} - y_{P1})^2 + (x_{P1} - x_{P2})^2}{4(y_{P1} - y_{P2})} \quad (4.2)$$

A Figura 4.7 mostra a forma de se determinar os raios r_1 e r_2 mostrados na Figura 4.4 dos círculos que compõem a trajetória. A letra D indica a largura do carro.

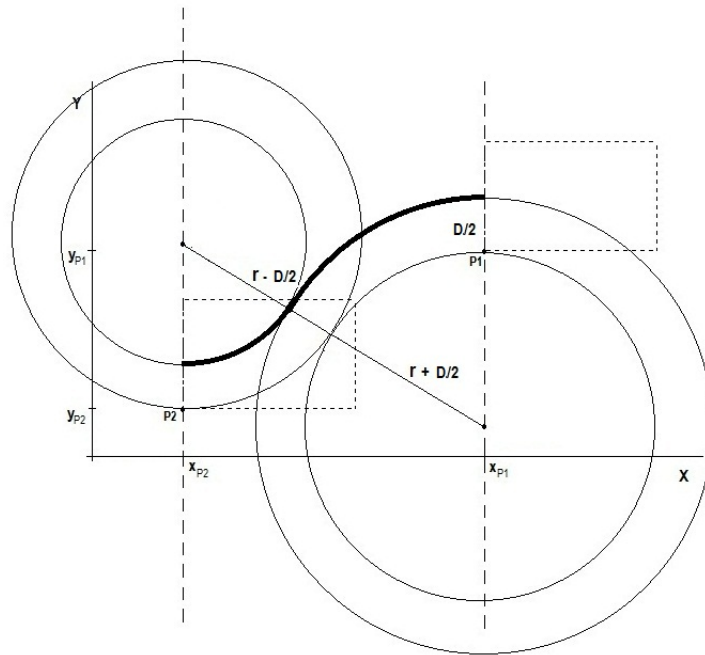


Figura 4.7: Cálculo da Trajetória - Círculos usados para gerar a trajetória

A orientação α que o carro terá (em relação à orientação inicial) quando tiver que parar para mudar o ângulo das rodas dianteiras é dada pela expressão:

$$\alpha = \arcsen \frac{x_{P1} - x_{P2}}{2r} \quad (4.3)$$

Para que o veículo estacione paralelamente à calçada, o condutor deverá passar com o carro também paralelamente a ela, pois o algoritmo prevê que a orientação final do veículo será a mesma que a inicial (em relação à calçada). Dessa forma, os comandos necessários para os acionamentos que fazem o carro descrever esse caminho são minimizados, e o algoritmo de controle é também simplificado. Assim, transformar este em um sistema de auxílio a um condutor humano se torna uma tarefa mais fácil.

4.2 Transformando a trajetória em sinais de comando

A Figura 4.8 ilustra a forma como calcular o comando que deverá ser enviado para o sistema de controle da direção para que o ponto central do eixo traseiro do veículo descreva uma trajetória circular, mantendo-se velocidades baixas. Nessa lustração, L é o comprimento da distância entre eixos do carro, R é o raio do círculo descrito pelo ponto central do eixo traseiro e Θ é o ângulo enviado ao sistema de controle da direção.

O comando para o sistema de controle da direção (ângulo Θ) para que o ponto central do eixo traseiro do veículo descreva uma trajetória circular de raio R é calculado segundo a seguinte equação:

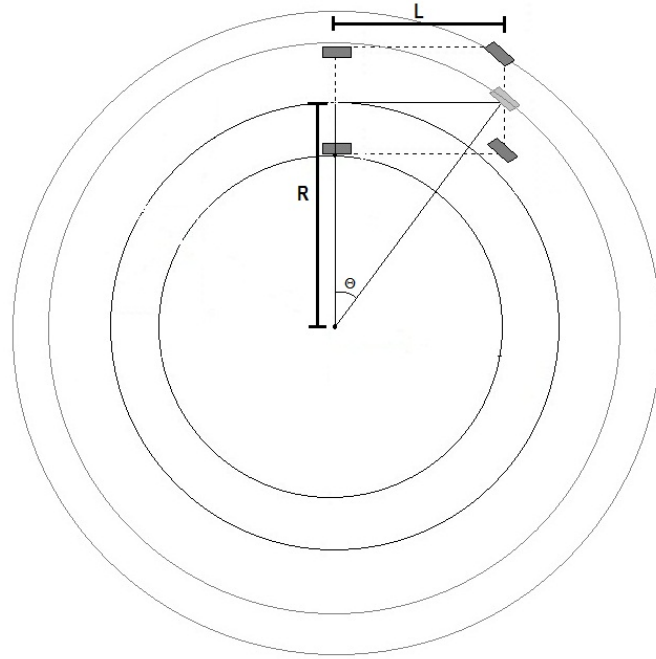


Figura 4.8: Calculando o comando para o sistema de controle da direção

$$\Theta = \text{arc tg} \frac{L}{R} \quad (4.4)$$

Assim, tem-se que, a partir do raio do primeiro círculo calculado pelo algoritmo gerador da trajetória, é calculado o comando de posição angular que será enviado ao sistema e controle da direção para que as rodas dianteiras sejam orientadas. Após a efetivação deste comando, os sistemas de troca de marchas e aceleração serão acionados, para que o carro entre em movimento e descreva o arco do primeiro círculo. O freio poderá ser acionado para manter a velocidade baixa. Após descrever este arco, o freio é acionado para que o carro pare.

Como o carro parado, o comando para orientar as rodas dianteiras, calculado a partir do raio do segundo círculo, é enviado. Após ser efetivado, acelerador e freio são comandados para que o carro descreva o restante da trajetória. Uma funcionalidade do programa executado no computador de bordo sincroniza o envio desses comandos para os controladores de cada atuador.

4.2.1 Algoritmo de Controle

O programa executado no computador de bordo calcula os valores dos comandos para que o veículo descreva a trajetória. São enviados, então, os comandos para que o veículo se movimente descrevendo o arco do primeiro círculo. É calculada, então, a orientação atual do veículo segundo a equação 4.1, em relação à orientação inicial. A partir das informações de ângulo do volante e velocidade da roda dianteira direita do carro calcula-se, a cada intervalo de tempo de aproximadamente 100 ms, a nova orientação do veículo. Caso esse valor se

aproxime³ do ângulo calculado para a inversão do sentido do movimento, o controlador envia um comando para o sistema de controle do freio para que o carro pare.

Depois de o carro parar, é enviada a instrução para que as rodas dianteiras sejam orientadas conforme o raio do segundo círculo. Após isso, o veículo é posto em movimento. Neste momento, novamente calcula-se a orientação atual em relação à do início do movimento. No momento em que a orientação do carro retorna a próximo a orientação inicial⁴, o veículo é freado. O objetivo final é que, após a efetivação dos comandos calculados a partir da trajetória, o carro esteja estacionado paralelamente à calçada sem haver colisão com qualquer obstáculo.

Evitando obstáculos

Enquanto o carro descreve a trajetória calculada, os sensores numerados de 2 a 8 na Figura 3.16 estarão fazendo a medição de distância e enviando-a ao computador central. Caso algum deles detecte um obstáculo a uma distância menor que 50 cm, que se traduz em perigo para a realização do estacionamento, a manobra será abortada. O veículo será parado onde estiver e não se tentará uma correção na trajetória para que este possa, ainda assim, parar no local desejado. Fatos desta natureza poderão ocorrer devido ao mapeamento do ambiente ter sido feito em velocidades superiores a 10 Km/h. Dessa forma, algum potencial obstáculo pode não ter sido detectado pelo sensor quando da medição da vaga e pode atrapalhar a descrição da trajetória calculada por parte do carro.

A implementação deste controlador é mais uma das funcionalidades do programa executado no computador de bordo.

4.3 Programa de Operação do Veículo

A Figura 4.9 apresenta a interface gráfica do programa desenvolvido. Foi utilizada a ferramenta de desenvolvimento Visual C++ 2008 Express Edition - versão para estudantes - fabricada pela Microsoft.

O programa possui os modos de operação automático e manual, que são escolhidos a partir dos controles com seus nomes. Em modo manual, é possível enviar comandos para os atuadores do acelerador, do freio, da direção e do câmbio usando-se os controles para selecionar o atuador que deseja operar e o botão 'Enviar'. O último valor enviado é mostrado no campo próximo ao controle de seleção de cada atuador. É possível, também, enviar comandos para que os sensores de ultra-som sejam acionados e tenham seus valores mostrados na tela. Na parte superior, é mostrada a distância medida até o obstáculo pelo sensor Polaroid (em cm).

³Em testes para uma determinada vaga, este valor foi de 80% do ângulo calculado para a inversão do sentido do movimento

⁴Em testes para uma determinada vaga, este valor foi de 20% do ângulo calculado para a inversão do sentido do movimento

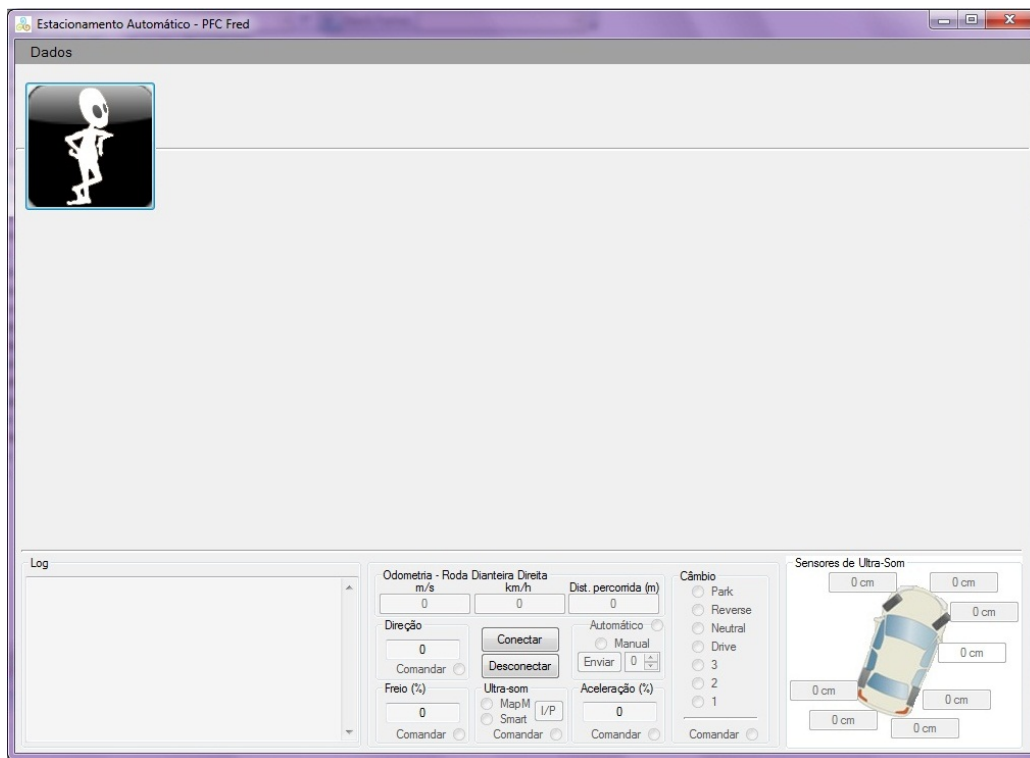


Figura 4.9: Interface gráfica do programa desenvolvido

Na parte inferior direita, são mostradas as informações medidas pelos sensores HG-M40DAII (em cm). O programa apresenta a figura de um automóvel e campos em que a informação é exibida, relacionando-a ao local onde o sensor foi instalado no carro. Em modo automático, o programa comanda o veículo para que a leitura do espaço correspondente à vaga e a manobra do estacionamento sejam executadas. Essas ações são disparadas pressionando-se o botão localizado na parte superior esquerda. São mostrados, no espaço central da janela do programa, uma estimativa da posição ocupada pelo carro em relação à posição de início da vaga (Figura 4.10) e, após encontrada a vaga, esta também é mostrada. O desenho vai se completando à medida que o carro descreve a trajetória calculada até que o carro ocupe posição final da manobra.

Outra funcionalidade é a visualização da velocidade medida (em m/s e Km/h) e da estimativa do espaço percorrido (em m), possível nos dois modos de operação. Os dados de velocidade e distância aos obstáculos enquanto o veículo procura pela vaga podem ser salvos em arquivo de texto através do menu "Dados", opção "Medidas do Mapa". O menu "Dados", opção "Salvar Dados Tela" permite que sejam salvos todos os dados que foram mostrados na tela durante a execução da manobra.

Na parte inferior esquerda da janela, um campo exibe as informações de estado das conexões com os microcontroladores responsáveis pela instrumentação.

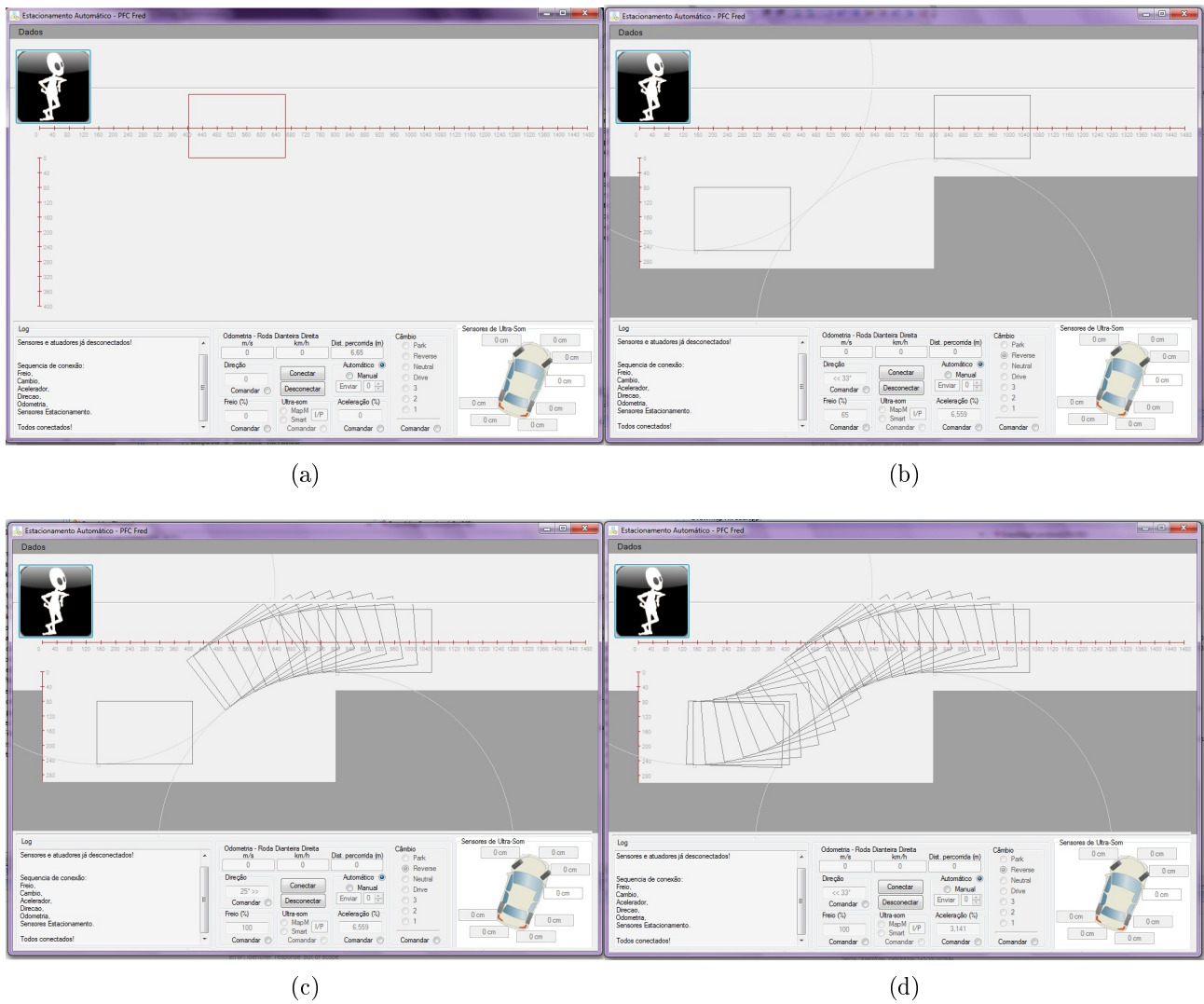


Figura 4.10: Desenho do carro completando a manobra do estacionamento

4.3.1 Desenvolvimento do Programa

Este programa foi implementado a partir de classes anteriormente desenvolvidas por outros membros do laboratório para a comunicação com os sistemas de instrumentação embarcados no veículo.

O diagrama UML dessas classes é mostrado na Figura 4.11.

Inicialmente, a classe CSerial disponibiliza funções para a comunicação serial com dispositivos periféricos conectados ao computador principal via RS232. A classe Módulo utiliza um objeto da classe CSerial e implementa um protocolo padrão de comunicação entre o computador central e os microcontroladores PIC (de mesmo modelo) utilizados nos trabalhos.

As classes Freio, Acelerador e Câmbio têm como base a classe Módulo e definem funções personalizadas de acordo com as operações que têm que executar.

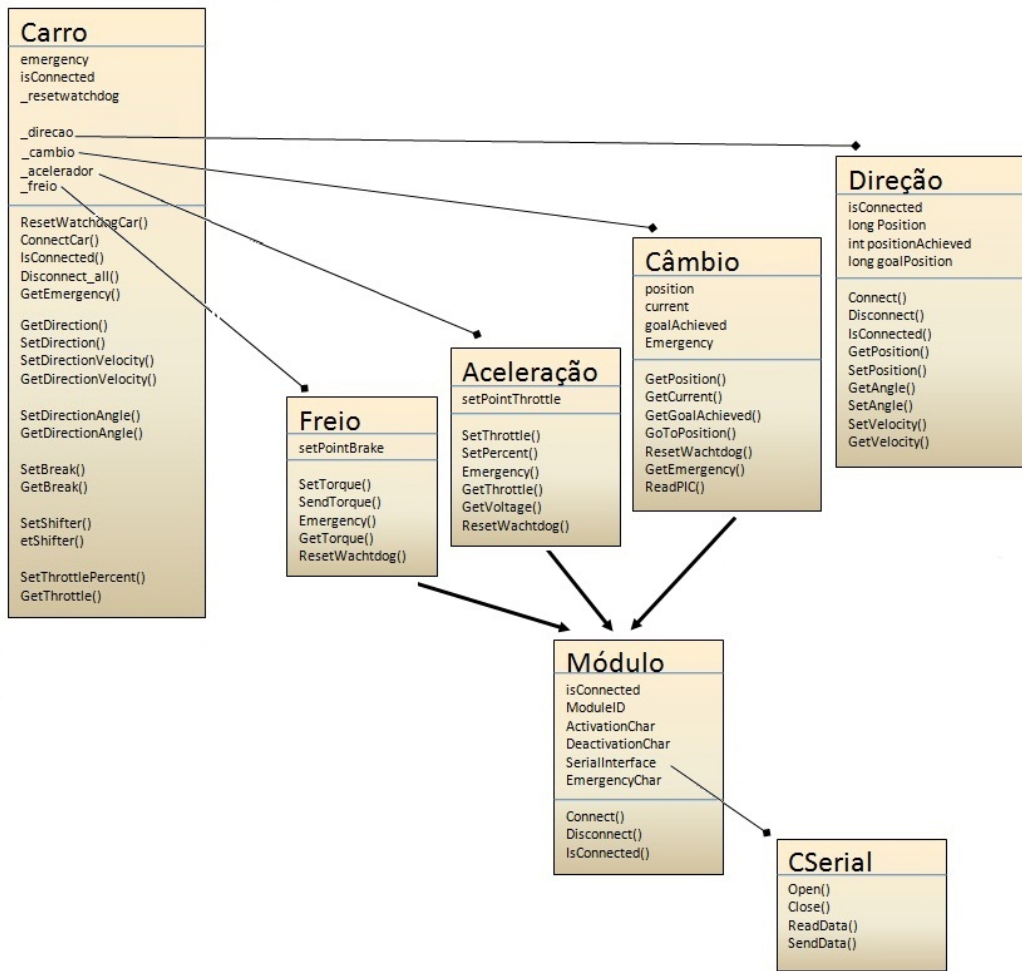


Figura 4.11: Diagrama UML simplificado da Classe Carro

A classe Direção tem implementação diferente. O acionamento da direção é feito a partir de um dispositivo proprietário, da fabricante Maxon. Um conjunto de funções é disponibilizado em uma biblioteca de ligação dinâmica e o subconjunto utilizado é reunido em uma classe. A partir das funções desta classe, é possibilitada a operação deste atuador.

As classes Propriedades e ResetWatchDog foram omitidas no diagrama. A classe Propriedades é usada para anular um ou mais sistemas do carro, permitindo que o programa seja executado com base somente no código dos sistemas ativados. A classe ResetWatchdog provê funções que permitem verificar, em intervalos de tempo mais ou menos constantes, se não houve a perda de comunicação com os sistemas de atuação de direção, freio e aceleração.

A classe Carro reúne a operação dos sensores/atuadores em uma interface única. Esta classe utiliza objetos das classes Freio, Direção, Câmbio, Acelerador, ResetWatchDog, Propriedades e pelas classes desenvolvidas neste trabalho. São disponibilizadas funções que permitem atuar ou receber informações enviadas pela instrumentação de forma padrão. Dessa forma, o desenvolvimento de programas em trabalhos que utilizam os sistemas embarcados no carro é simplificado.

O desenvolvimento do sistema partiu das premissas de que os microcontroladores deveriam ser os mesmos e o protocolo deveria ser mantido, visando a padronização dos trabalhos no

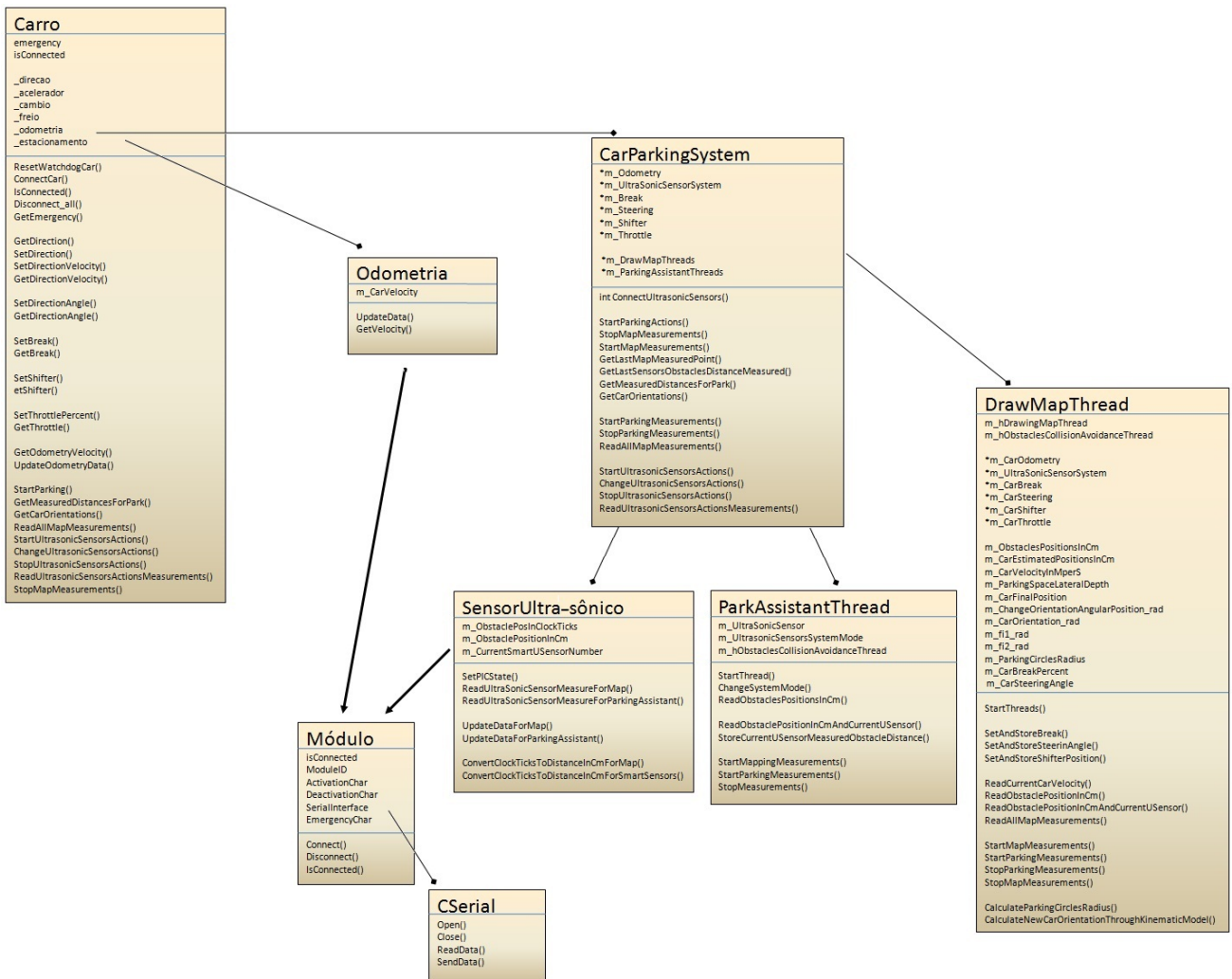


Figura 4.12: Diagrama UML simplificado das classes desenvolvidas neste trabalho

veículo autônomo CADU. Assim, as classes Odometria e SensorUltra-sônico são baseadas na classe Módulo. O diagrama dessa parte do sistema é mostrada na Figura 4.12.

A classe Odometria é uma adaptação da classe implementada para o mesmo fim, utilizada em trabalhos anteriores com o outro sistema de medição de velocidade existente. A comunicação com o microcontrolador foi personalizada para possibilitar a leitura da velocidade a partir do sistema desenvolvido no presente trabalho. Esta é usada na composição da classe Carro, sendo a responsável por disponibilizar a informação de velocidade.

A classe CarUltrasonicSensor foi desenvolvida para permitir a disponibilização da informação de distância a um obstáculo por algum dos sensores. Ela implementa a comunicação com o microcontrolador e o processamento dos dados enviados por ele.

A classe Drawmapthread implementa as funções de obtenção dos dados para a construção do mapa, a construção do mapa, a geração da trajetória e o controlador. Esta classe está relacionada com o modo automático de execução do programa. Nela, está encapsu-

lado todo o fluxo necessário nas tarefas de leitura da vaga e de efetuação da manobra do estacionamento. São implementadas também as funcionalidades de atualizar os dados e o desenho mostrados na janela do programa, bem como salvar esses dados em arquivos de texto.

A classe `ParkingAssistantThread` implementa as funções relacionadas à obtenção e disponibilização dos dados mostrados na tela, em modo manual.

A classe `ParkultrasonicSystem` disponibiliza uma interface padrão para que as funcionalidades relacionadas ao sistema de estacionamento sejam mais facilmente acessadas. É composta por objetos das classes `ParkingAssistantThread` e `Drawmapthread`, que reúnem as funções do sistema. Esta é usada na composição da classe `Carro` para fornecer funcionalidades ligadas à obtenção de dados via sensores de ultra-som (modo manual) e à confecção do mapa, da geração da trajetória e efetuação da manobra (modo automático).

Capítulo 5

Resultados Experimentais

Dois dos sensores do subsistema de medição de distância aos obstáculos foram testados em laboratório. Foi utilizada uma montagem contendo o computador de bordo do veículo e o subsistema de medição sendo alimentado por uma fonte de tensão. Em cada teste, o sensor foi posicionado a uma distância em relação ao solo parecida com a que ficariam se estivesse instalado no carro. Foi usada uma caixa de papelão como obstáculo.

5.1 Testes do Sensor Polaroid

A Figura 5.1 mostra fotos da montagem feita para o teste do sensor Polaroid. Como o sensor está posicionado muito próximo ao solo, foi necessário que este fosse instalado com uma pequena angulação em relação a este, para evitar que o sensor efetuasse leitura da distância em relação ao próprio pavimento. Este sensor tem como objetivos: medir a distância em relação aos outros automóveis parados antes e após a vaga; e, principalmente, medir a distância em relação à calçada, enquanto o sistema estiver realizando a leitura da vaga. Assim, como obstáculo, foi usada uma caixa de papelão com aproximadamente 12 cm de altura.

A caixa foi posicionada a diferentes distâncias do sensor. Uma fita métrica foi utilizada como referência. Foram efetuadas 50 medidas para cada posição.

Os resultados são apresentados na Tabela 5.1. Para cada posição testada, são apresentados o valor máximo e mínimo medido pelo sensor e a média das medições. Os valores são dados em cm.

Discussão dos resultados

Sensores de ultra-som fazem medições de distância em relação a um objeto que está posicionado em região à sua frente. Esta região é caracterizada por um ângulo de abertura característico de cada sensor. O sensor Polaroid foi calibrado de tal forma que esse ângulo fosse de 20 graus. Sendo assim, qualquer objeto que estiver dentro da região em que pode ser visto, será detectado. A resposta do sensor é uma simples medida de distância, sendo a

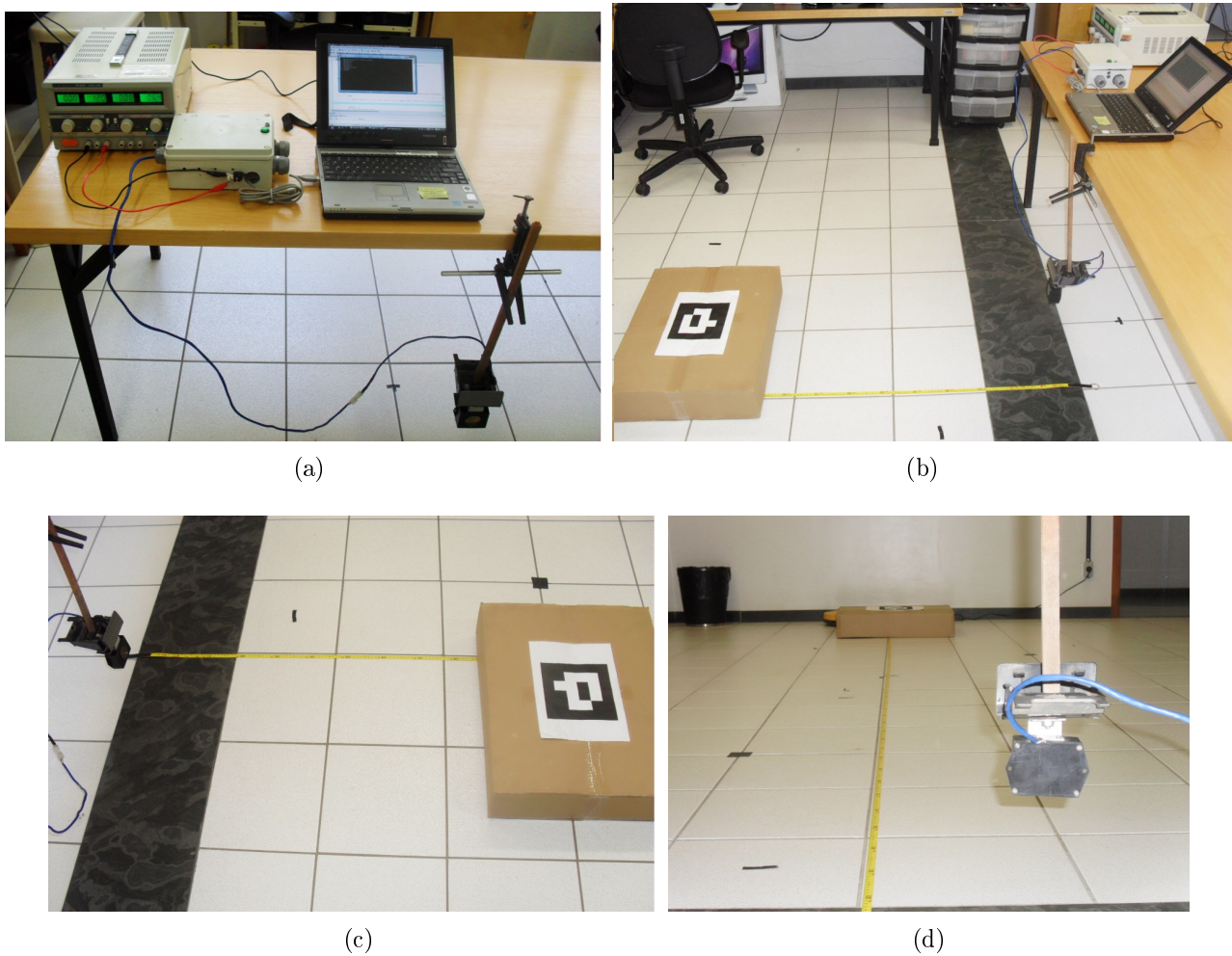


Figura 5.1: Testes em laboratório usando o sensor Polaroid 6500

distância ao objeto (ou parte dele) que estiver mais próxima ao sensor. No caso da calçada (ou caixa de papelão, conforme usado no teste), a menor distância medida será sempre em relação ao ponto localizado em frente ao sensor, desde que o carro esteja se movimentando paralelamente à calçada, premissa do uso deste sistema.

Dessa forma, o teste mostrou que as medições feitas pelo sensor são próximas às distâncias reais, mesmo para o caso em que o objeto esteja posicionado a uma distância maior que 3 metros do sensor.

5.2 Testes do Sensor HG-M40DAII

A Figura 5.2 apresenta a montagem feita para o teste do sensor HG-M40DAII. Estes sensores têm como objetivo medir a distância em relação a possíveis obstáculos enquanto o veículo estiver realizando a manobra do estacionamento. Logo, a caixa (obstáculo) foi colocada de forma diferente da forma do teste para o outro sensor, objetivando uma maior aproximação da realidade.

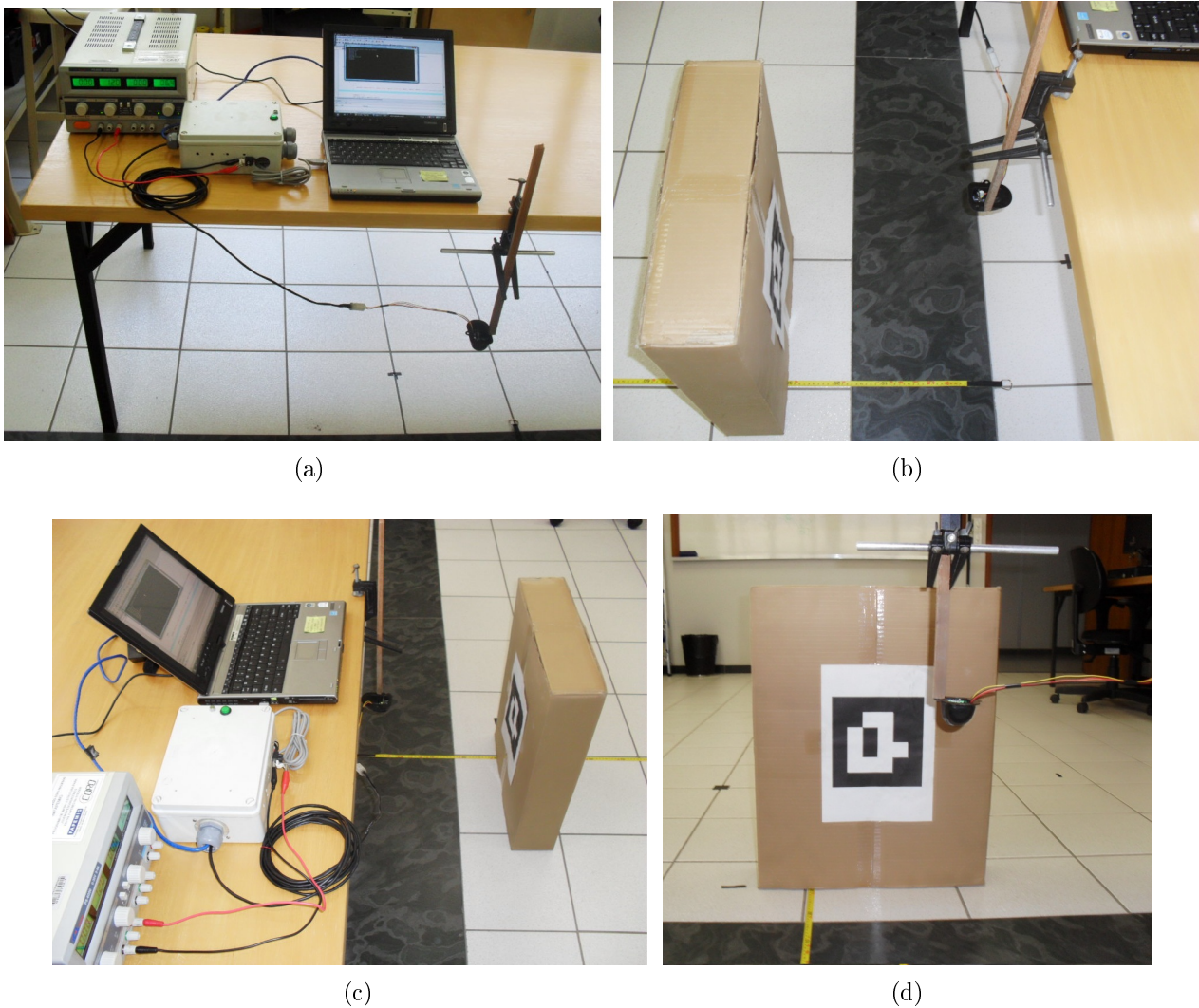


Figura 5.2: Testes em laboratório usando um sensor HG-M40DAII

Também foi usada uma fita métrica como referência e realizadas 50 medidas para cada posição. Os resultados são apresentados na Tabela 5.2.

Discussão dos resultados

Estes sensores foram calibrados para apresentar um ângulo de abertura de 40 graus e perceber objetos posicionados a menos de 2 metros de distância. O valor medido refere-se ao obstáculo (ou parte dele) que estiver mais próximo, não se tendo uma perfeita informação de localização deste nem de angulação que ele se encontra em relação ao sensor. Como, neste trabalho, não é necessário saber a posição exata do obstáculo em relação ao sensor, os dados medidos são suficientes para se determinar se existe ou não um obstáculo próximo ao carro.

Outros trabalhos que porventura vierem a necessitar da informação do posicionamento exato do obstáculo em relação ao sensor podem alterar o ângulo, diminuindo a área da região

em que o sensor percebe obstáculos. Assim, o objeto poderá ter sua localização identificada com maior precisão. Outra forma seria a de cruzamento da informação obtida através de sensores posicionados próximos um do outro. Dessa forma, poderá ser possível estimar a localização do obstáculo em relação aos sensores com maior eficiência ainda.

5.3 Testes do Sensor de Velocidade

A validação do subsistema de medição de velocidade foi feita em testes de campo. Pequenas distâncias foram percorridas em linha reta pelo carro e a estimativa do comprimento desse caminho apresentada pelo programa foi verificado. Este valor foi comparado com o valor real andado pelo carro, medido através de uma fita métrica. Assim, a melhor forma encontrada para validar o valor de velocidade lido pelo sensor foi utilizando uma integração. Não foi utilizada a informação do velocímetro como forma de validação porque a velocidade com a qual o carro trafega é muito baixa (menor que 5 Km/h). Para esta velocidade, o velocímetro não apresenta resolução suficiente para que fosse feita uma leitura confiável do valor medido.

Nos testes realizados, o veículo percorreu certa distância em linha reta com velocidades próximas a 1 Km/h. O valor estimado pelo programa (a partir da integração) e o medido por uma fita métrica foram comparados. Foram feitas poucas repetições, mas os resultados apresentavam erros menores que 3% do valor percorrido.

Discussão dos resultados

Apesar de o sistema operacional do computador de bordo não ser um RTOS¹, o programa conseguiu realizar a tarefa da integração com um nível de eficiência considerado bom para os testes. Houve pouca variação nas distâncias estimadas, mas que pôde ser apontada como problema relacionado à própria integração e não à calibração do sistema de medição da velocidade. Esses testes mostraram que a tarefa de integração na computação, quando não é usado um sistema operacional de tempo real, pode se tornar um problema de difícil solução. Neste caso, como as distâncias percorridas são pequenas, o erro intrínseco à integração não reflete em grandes problemas quanto à execução da manobra. Mas, para o caso de distâncias mais longas, a diferença entre o valor integrado e o realmente percorrido poderão ser muito grandes, acarretando em uma confiabilidade baixa para o valor resultante.

5.4 Estimação das medidas da vaga

Os testes realizados em campo tinham como objetivo validar a união das informações dos dois sistemas. O objetivo principal foi obter as informações de comprimento e profundidade

¹do Inglês, *Real-Time Operating System*

de uma vaga, após o veículo passar ao lado desse espaço. Para a realização destes testes, dois cones foram posicionados marcando o início e o fim do espaço correspondente à vaga, como se fossem parte de dois carros (semelhante à forma de como a baliza é feita nas auto-escolas). Para marcar a profundidade, foram usados mais dois cones e duas caixas.

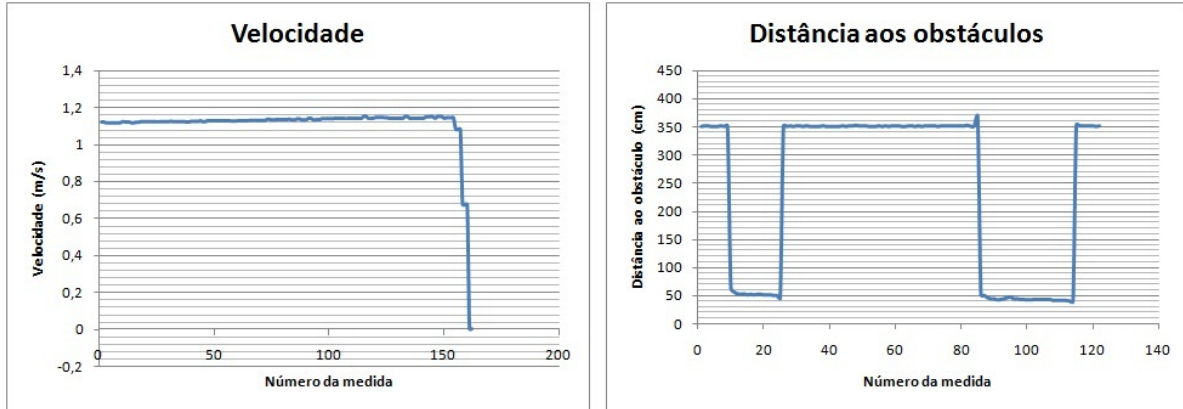


Figura 5.3: Dados dos sensores de velocidade e ultra-som em testes de campo

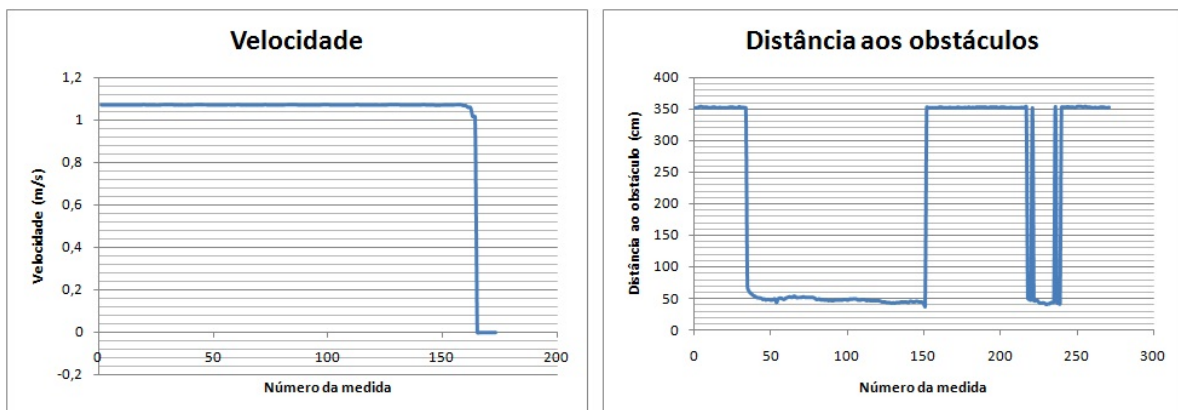


Figura 5.4: Dados dos sensores de velocidade e ultra-som em testes de campo

Os gráficos apresentados nas Figuras 5.3 e 5.4 mostram os dados medidos pelo sensor de ultra-som e os dados medidos pelo sensor de velocidade lado a lado. Os gráficos que apresentam a informação de velocidade mostram que o veículo mantém uma velocidade aproximadamente constante enquanto o sistema efetua as medições para encontrar a vaga. Os gráficos que apresentam a informação de distância aos obstáculos mostram que é possível obter os momentos em que o sensor está medindo distâncias em relação à calçada e o momento em que as distâncias medidas são em relação a objetos mais próximos ao veículo.

Nestes testes, o programa executado no computador de bordo utilizava o algoritmo baseado na integração do valor medido pelo sensor de velocidade para estimar o comprimento da vaga. No eixo Y, é mostrado o valor medido pelo sensor e no eixo X, o número da medida. No gráfico de distância ao obstáculo, não existe a informação de tempo decorrido entre as medidas.

No gráfico de velocidade, as tomadas são feitas em intervalos mais ou menos constantes de 50 ms.

Em outros testes, o algoritmo utilizado previa que o sensor de ultra-som efetuaria as medições e as enviaria na mesma frequência para o computador. O sensor de velocidade mediria, agora, com aproximadamente 10 Hz. O programa efetuaria a leitura do valor medido pelo sensor de velocidade, mas o descartaria até que os dados do sensor de ultra-som indicassem que o veículo passou pelo primeiro cone. A partir deste momento, os dados medidos pelo sensor de velocidade passariam a ser guardados em memória até que o segundo cone fosse detectado. A média das medições de velocidade guardadas em memória seria multiplicada pela diferença de tempo entre o veículo ter passado pelos cones.

Discussão dos resultados

Apesar de a integração do valor medido pelo sensor de velocidade não ser a mais apropriada para esta aplicação, este algoritmo apresentou alguns resultados em que o comprimento da vaga se aproximava do valor real. Em muitos casos, a união dos dados dos dois sensores apresentou problemas. A maioria deles em relação ao descompasso entre a escrita de dados na porta serial pelo sensor e a leitura desses dados pelo computador. O sensor de ultra-som efetua medições com a uma frequência de mais de 10 Hz. O sensor de velocidade, neste teste, mediu a aproximadamente 20 hertz. Como não foi implementada uma forma de sincronismo passível de ser executada pelo computador de bordo eficientemente, foi necessário procurar uma solução melhor.

O segundo algoritmo de cálculo do comprimento da vaga é mais simples que o primeiro. Reduzir a frequência das medições de velocidade não impactou no resultado final, visto que o veículo apresenta aproximadamente um movimento uniforme, como pode ser observado nos gráficos das Figuras 5.3 e 5.4 do conjunto de experimentos do primeiro algoritmo. Devido a isso, o fato de os intervalos entre as medições de velocidade não serem constantes não influenciou negativamente no cálculo da velocidade média. Não é feita também a integração do valor medido pelo subsistema de medição de velocidade. Assim, o custo computacional inerente à execução do segundo algoritmo é menor. Foram realizados poucos testes e os resultados apresentados foram melhores que os do primeiro algoritmo. Mas uma seqüência maior de teses é necessária antes de se afirmar que este algoritmo é melhor que o anterior.

5.5 Algoritmo para Geração de Trajetórias

O algoritmo de geração de trajetórias foi testado de três formas diferentes. Foram utilizados dois ambientes computacionais e foram realizados testes em campo. O programa desenvolvido neste trabalho desenha na tela o caminho descrito pelo carro durante a execução da manobra. O desenho é feito em escala e é baseado na geometria de Ackermann, tornando-se o mais

realístico possível. O carro é desenhado na tela conforme a trajetória gerada pelo algoritmo planejador. A Figura 4.10 mostra um possível resultado de uma baliza.

A segunda forma de validação utilizada foi a de testes em campo. O algoritmo foi implementado no programa e partiu-se do conhecimento prévio do comprimento e da profundidade da vaga. O programa, então, calculou a trajetória necessária ao estacionamento naquela vaga, gerou os comandos necessários para que o veículo seguisse a trajetória e os enviou para os sistemas de atuação. Nestes testes, pôde ser observado também se os comandos derivados da trajetória faziam com que o veículo seguisse o caminho desejado. A Figura 5.5 é retirada de um vídeo que mostra a realização deste teste. Nela, é possível ver o ponto de início da manobra e o ponto em que o veículo estacionou, ao final desta.

Uma terceira forma de validação da trajetória calculada foi utilizando-se um ambiente de testes desenvolvido com o programa MATLAB. Um programa possibilitou ajustes em seu código para que fosse feita a verificação da trajetória utilizada neste trabalho. Nele, são desenhados retângulos (simulando carros estacionados paralelamente à calçada) e um deles é redesenhado na tela obedecendo a orientação e o posicionamento calculados utilizando as equações da geometria de Ackermann. A Figura 5.6 mostra um resultado desta simulação.

Discussão dos resultados

Os resultados mostrados pelo primeiro método de avaliação mostram que a trajetória gerada pelo algoritmo é capaz de fazer com que o veículo estacione sem haver colisão com qualquer obstáculo desde que os comandos sejam corretamente efetivados pelos sistemas de acionamento. Como o desenho da localização do carro obedece às equações do modelo de Ackermann, pôde-se ter uma idéia da efetividade do algoritmo de geração da trajetória.

Os testes em campo foram realizados também na tentativa de fazer com que o veículo encontrasse a vaga e realizasse a manobra. O primeiro importante resultado observado foi que o veículo não colidiu com o cone posicionado à frente da vaga em nenhuma oportunidade. O algoritmo de geração da trajetória prevê que o raio do primeiro semicírculo seja maior que o do segundo justamente para evitar esta situação.

Inicialmente, o carro quase sempre colidia com as caixas utilizadas para simular a calçada e o carro parado atrás da vaga. Mas, neste caso, o problema estava relacionado ao envio dos comandos derivados da trajetória e não da geração da trajetória em si. Sendo assim, após uma calibração do controlador, o veículo passou a parar próximo ao meio-fio e ao cone usado para marcar o início da vaga, não mais colidindo-se com estes.

Após os testes em ambiente MATLAB, os resultados encontrados anteriormente foram confirmados, aumentando-se a confiabilidade do algoritmo de geração de trajetórias.

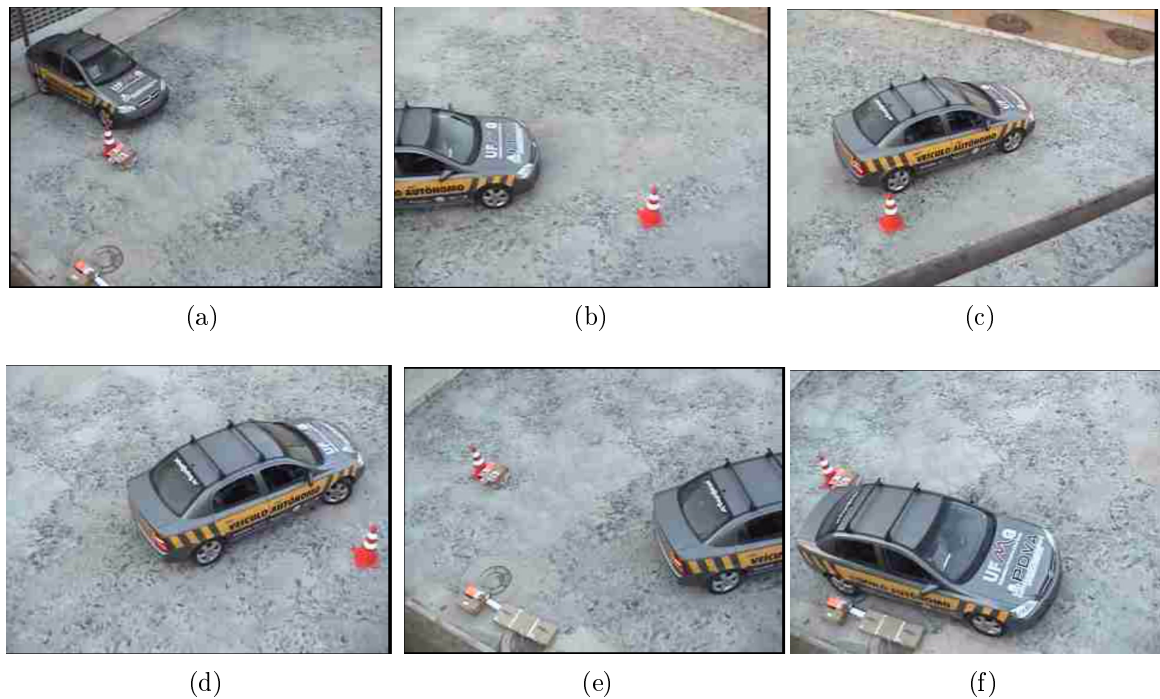


Figura 5.5: Imagens retiradas de um vídeo que mostra o veículo estacionando

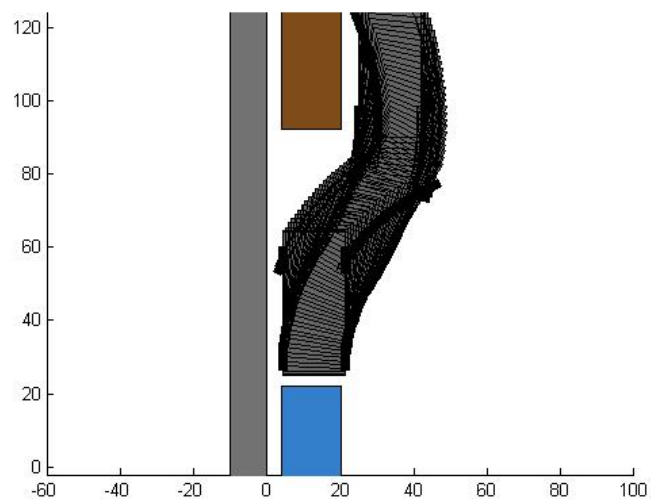


Figura 5.6: Resultado da validação do algoritmo de geração de trajetórias via MATLAB

5.6 Sistema de Estacionamento Automático

Foram realizados alguns testes de campo que almejavam validar o funcionamento de todo o sistema. Para marcar a vaga, foram usados cones e caixas, da mesma forma que nos testes anteriores. As caixas foram posicionadas paralelamente à calçada. Esta configuração foi montada na tentativa de simular um ambiente real. O veículo deveria passar pela vaga e parar, após encontrar o segundo cone, em uma posição favorável ao início da manobra. Deveriam ser

feitos os cálculos da trajetória e os comandos necessários para acionar os atuadores deveriam ser também gerados. O controlador enviaria os comandos e o veículo efetuaria a manobra. O objetivo final foi observar se o carro parava dentro da vaga em uma posição próxima à estimada pelo algoritmo.

Discussão dos resultados

Os primeiros testes foram marcados por falta de sincronismo entre a obtenção dos dados de velocidade e distância aos obstáculos. Neste momento, ainda era usado o algoritmo da integração. O mapa montado com os dados de comprimento e profundidade da vaga era, em alguns casos, muito diferente do que deveria ser. Este erro era propagado porque havia a geração de uma trajetória muito distante da ideal. Ao seguir a trajetória proposta, somando-se os erros inerentes à execução desta tarefa, os resultados eram muito ruins. Mas, quando as medições resultavam na montagem de um mapa próximo ao ambiente real, o carro parava mais perto da posição desejada.

O controlador foi calibrado para a vaga em que os testes estavam sendo realizados e, quando o mapa era montado com dimensões próximas às da realidade, o veículo estacionava bem perto do que foi desenhado. Com esta calibração, foi possível eliminar as situações em que o carro subia em cima das caixas que simulavam a calçada e o veículo parado antes da vaga.

O próximo passo foi, então, implementar uma nova forma de ler e unir os dados de velocidade e distância aos obstáculos. O objetivo foi conseguir uma maior repetição da situação em que o mapa refletia bem a realidade. Implementou-se, então, o algoritmo baseado na velocidade média. Com ele, esperava-se que o mapa montado pelo programa apresentasse boa fidelidade com relação ao ambiente com uma frequência maior que a do seu antecessor. Os resultados, nos poucos testes feitos, demonstraram que é possível se ter uma frequência maior de vezes em que o carro lê bem o tamanho da vaga. Com isso, novos testes poderão apresentar uma melhor performance que os realizados até o momento.

Foi observado que o sistema apresenta fragilidades, principalmente pela forma manual de construção dos circuitos eletrônicos. Isso resultou em muitos problemas de mau contato, que atrapalharam a realização dos testes. Houve dias em que estes problemas até impossibilitaram que o sistema fosse testado como um todo. Assim, não foi possível a realização de uma seqüência de testes para a obtenção de dados qualitativos para serem apresentados.

Tabela 5.1: Testes Sensor Polaroid

Distância (cm)	Menor medida (cm)	Maior medida (cm)	Média das medidas (cm)
50	49,63	50,37	49,90
60	58,15	58,52	58,17
80	76,67	77,04	76,81
90	88,52	89,26	88,73
100	99,26	100,00	99,35
120	118,89	119,63	118,97
130	128,15	128,89	128,52
140	137,78	138,15	138,08
150	147,41	147,78	147,60
170	168,15	168,52	168,29
180	177,78	177,78	177,78
190	187,04	188,15	187,87
200	197,78	197,78	197,78
210	207,41	208,15	207,77
220	217,78	218,15	217,86
240	237,78	238,89	238,20
250	246,67	247,04	246,72
260	256,30	256,67	256,44
270	266,67	267,04	266,90
280	276,67	277,04	277,02
290	287,04	287,41	287,37
300	297,67	298,15	297,43
320	317,41	319,26	318,15
340	337,41	337,78	337,63
360	357,15	358,15	357,89
370	387,78	368,15	367,89
380	377,41	377,78	377,59
390	387,26	388,15	387,93
400	397,04	397,41	397,11
410	407,41	407,78	407,67
420	416,30	418,15	417,04

Tabela 5.2: Testes Sensor HG-M40DAII

Distância (cm)	Menor medida (cm)	Maior medida (cm)	Média das medidas (cm)
30	30,14	30,50	30,34
40	39,79	40,14	39,88
50	49,43	50,50	49,90
60	59,07	59,79	59,28
70	68,71	70,50	68,98
80	79,07	80,14	79,73
90	88,36	89,43	88,65
100	98,36	99,79	99,11
110	110,86	110,86	110,86
120	119,07	120,14	119,70
130	128,36	130,50	129,08
140	138,71	139,79	139,01
150	148,36	150,86	149,39
160	159,79	160,86	160,29
170	169,07	171,21	169,99
180	178,00	179,79	178,96
190	188,00	188,36	188,23
200	198,36	198,36	198,36

Capítulo 6

Conclusão

6.1 Conclusão

Através da realização deste trabalho, foi possível obter uma experiência teórica e prática das etapas envolvidas na realização de um projeto de engenharia. Os resultados obtidos nos testes diante da dificuldade da construção do sistema foram considerados satisfatórios.

O subsistema de leitura de distâncias utilizando o sensor Polaroid Serie 6500 apresenta medições com erros percentuais baixos tanto em laboratório quanto em campo. Os circuitos montados na matriz de contatos apresentam pouca robustez e problemas de perda de conexão são recorrentes, mas não impossibilitaram que o sistema fosse embarcado no veículo para a realização dos testes. O subsistema de medição de velocidade pode ser usado para medir velocidades mais baixas, abrindo caminho para a realização de trabalhos que necessitem deste tipo de informação. Testes em que o valor da velocidade é integrado mostraram que a distância percorrida pelo carro e a indicada pela integração apresentam erro menor que 3%, evidenciando a qualidade do valor medido pelo subsistema. Os dois subsistemas trabalham bem separadamente. Apesar disso, os testes indicaram que a estratégia tentada para unir as informações obtidas por ambos apresentou um número pequeno de bons resultados. Outra forma de união das informações foi testada poucas vezes, mas não pode ter sua eficiência verificada. Sendo assim, a geração de um mapa da vaga que apresentasse dimensões próximas às da realidade foi possível somente em poucas tentativas. Dessa forma, o objetivo de detectar o espaço correspondente a uma vaga foi automaticamente foi parcialmente atingido.

O algoritmo de geração da trajetória foi avaliado na prática e os resultados apresentados tanto nas simulações, quanto nos testes em campo, mostraram que é possível estacionar o veículo dentro da vaga sem que haja colisão com obstáculos, desde que as informações das dimensões da vaga sejam de boa qualidade. A trajetória gerada é traduzida em poucos comandos pelo controlador, que consegue acionar os sistemas de atuação do carro. Dessa forma, o objetivo de desenvolver um algoritmo que gerasse trajetórias para estacionar pôde ser atingido.

Testes em campo mostraram que é possível percorrer o caminho calculado pelo algoritmo

planejador sem haver colisão com eventuais obstáculos. O sistema de controle automático desenvolvido foi capaz de efetivar os comandos calculados com base na trajetória gerada, atingindo o terceiro objetivo proposto. Como os comandos necessários para que a trajetória seja seguida corretamente são poucos e fáceis de serem efetivados, foi observado que o sistema poderia ser alterado para que o veículo fosse acionado tanto por sistemas robóticos, quanto por um condutor humano.

O desejo de se construir um sistema robusto para a realização das tarefas de leitura do ambiente e realização da manobra do estacionamento aproximou-se de sua realização. Os objetivos do Projeto Final de Curso, no que diz respeito à aplicação dos conhecimentos obtidos durante a graduação, foram cumpridos. Foi necessário, inclusive, buscar informações em áreas afins à Engenharia de Controle e Automação para que os objetivos desse trabalho fossem alcançados.

No entanto, apesar dos bons resultados, o sistema desenvolvido ainda está distante da sua forma ótima de operação. A comunicação com o computador central apresenta falhas que ainda não puderam ter seus motivos estudados. Nos testes, houve um grande número de vezes em que essa comunicação era perdida. Na montagem do mapa, quando estão sendo feitas medições da distância em relação à calçada, às vezes, o sensor de ultra-som apresenta valores que claramente não são verdadeiros, mas nenhum método de exclusão destes foi implementado. O circuito de alimentação e acionamento dos sensores de ultra-som foi montado manualmente. Por causa disso, os problemas de mau contato também são frequentes. Os sensores HG-M40DAII foram pouco usados. Então, falhas relacionadas ao seu funcionamento não foram observadas. Assim, eventuais aprimoramentos do sistema desenvolvido devem, inicialmente, considerar estes fatores.

6.2 Trabalhos Futuros

As sugestões iniciais são no sentido de melhorar a robustez do sistema desenvolvido. Existem falhas no circuito de alimentação e acionamento dos sensores de ultra-som que necessitam ser reparadas. Uma nova forma de acioná-los poderia ser também proposta, de forma a se obter uma maior frequência de medição para cada sensor, visto que alguns deles se localizam bem distante dos outros.

Até aqui, todos os trabalhos desenvolvidos para o CADU em que há a necessidade de comunicação entre o computador central e microcontroladores utilizam a porta USB. Esta forma de comunicação, neste trabalho não apresentou a robustez necessária à sua utilização. O conector utilizado na placa de circuito impresso do microcontrolador apresentou folga com o intenso uso, que resultava em queda na conexão com o computador central. Por este motivo, propõe-se que outra tecnologia para a comunicação entre os microcontroladores e o computador de bordo possa ser utilizada, como a rede Modbus apresentada em [Arruda 2009].

O desenvolvimento da instrumentação baseada em ultra-som abre caminho para que novos projetos que necessitem deste tipo de sensor sejam desenvolvidos. Uma sugestão seria projetar um novo algoritmo de geração de trajetórias, a citar campo de potenciais, para estacionar. A informação dos sensores poderia ser usada como *feedback* para que o carro desviasse de eventuais obstáculos e continuasse executando a manobra. Outra sugestão poderia ser o uso destes sensores para informar o momento em que o carro se encontra paralelo ao meio-fio, diminuindo a dependência do sensor de velocidade. Assim, a premissa de que o veículo deve passar pela vaga paralelamente à calçada não seria mais necessária. Outros trabalhos que necessitem medir distância a objetos ou simplesmente detectá-los sem que haja contato físico podem também utilizar o sistema de ultra-som desenvolvido neste projeto.

O sistema de medição para baixas velocidades pode ser usado na execução de manobras que necessitem melhor localização espacial. Dessa forma, novos trabalhos que utilizem controle espacial podem ser pensados. Neste contexto, existe a necessidade de que se disponha de um sistema operacional de tempo real, o que permitirá que as manobras sejam executadas com maior exatidão.

Referências Bibliográficas

- [ABAD e KAKINAMI 2007]ABAD, F.; KAKINAMI, T. Parking space detection. In: *14th World Congress on Intelligent Transport Systems*. Beijing (China): [s.n.], 2007.
- [Arruda 2009]ARRUDA, T. 2009. Rede de instrumentação para aplicação em controle embarcado. Projeto Final de Curso. Belo Horizonte, Brasil.
- [Baleeiro 2009]BALEEIRO, M. F. Sistema de acionamento elétrico para frenagem de um veículo autônomo. p. 1–12, dez 2009.
- [C18 2005]C18. 2005. Microchip Technology Inc. *MPLAB C18 C Compiler Libraries*, 2005.
- [CARULLO e PARVIS 2001]CARULLO, A.; PARVIS, M. An ultrasonic sensor for distance measurement in automotive applications. *IEEE Sensors Journal*, v. 1, n. 2, p. 143–147, ago 2001.
- [Choset et al. 2005]CHOSSET, H. et al. Principles of robot motion. *The MIT Press*, 2005.
- [DIRK 2002]DIRK, B. A novel ultrasonic sensing system for autonomous mobile systems. *IEEE Sensors Journal*, v. 2, n. 6, p. 597–606, dez 2002.
- [Dubins Curves, 2009]DUBINS Curves, 2009. [Http://msl.cs.uiuc.edu/lavalle/planning/node821.html](http://msl.cs.uiuc.edu/lavalle/planning/node821.html). Acessado em 01/10/2009.
- [HAGISONIC 2009]HAGISONIC. 2009. HAGISONIC, Data Sheet: *HG-M40DAII Anisotropic Sensor with Analog Circuit Data Sheet*.
- [HAUPTMANN, HOPPE e PUTTMER 2002]HAUPTMANN, P.; HOPPE, N.; PUTTMER, A. Application of ultrasonic sensors in the process industry. *Measurement Science and Technology*, v. 13, n. 8, p. R73–R83, ago 2002.
- [IRF630 1998]IRF630. 1998. STMicroelectronics. *NCHANNEL 200V-0.35W-9A-TO-220 FP MESH OVERLAY MOSFET Data Sheet*, 1998.
- [Laboratório CORO, 2009]LABORATÓRIO CORO, 2009. Disponível em <http://www.coro.cpdee.ufmg.br>. Acessado em 22/11/2009.
- [LM323K 1999]LM323K. 1999. National Semiconductor Corporation. *LM123-LM323A-LM323 3-Amp, 5-Volt Positive Regulator Data Sheet*, 1999.
- [LM339 1994]LM339. 1994. National Semiconductor Corporation. *LM139-LM239-LM339-LM2901-LM3302 Low Power Low Offset Voltage Quad Comparators Data Sheet*, 1994.

- [MAJCHRZAK, MICHALSKI e WICZYNSKI 2009]MAJCHRZAK, J.; MICHALSKI, M.; WICZYNSKI, G. Distance estimation with a long-range ultrasonic sensor system. *IEEE Sensors Journal*, v. 9, n. 7, p. 767–773, jul 2009.
- [MARIOLI, SARDINI e A 1988]MARIOLI, D.; SARDINI, E.; A, T. Ultrasonic distance measurement for linear and angular position control. *IEEE Transactions on Instrumentation and Measurement*, v. 37, n. 4, p. 578–581, dez 1988.
- [MCHPFSUSB 2007]MCHPFSUSB. 2007. Microchip Technology Inc. *MCHPFSUSB Firmware Users Guide*, 2007.
- [NAKAHIRA et al. 2001]NAKAHIRA, K. et al. Distance measurement by an ultrasonic system based on a digital polarity correlator. *IEEE Transactions on Instrumentation and Measurement*, v. 50, n. 6, p. 1748–1752, dez 2001.
- [OETIKER, BAKER e GUZZELLA 2009]OETIKER, M. B.; BAKER, P.; GUZZELLA, L. A navigation-field-based semi-autonomous nonholonomic vehicle-parking assistant. *IEEE Transactions on Vehicular Technology*, v. 58, n. 3, p. 1106–1118, mar 2009.
- [PARK et al. 2008]PARK, W. et al. Parking space detection using ultrasonic sensor in parking assistance system. In: *IEEE Intelligent Vehicles Symposium*. Eindhoven (Holanda): [s.n.], 2008.
- [PEPY, LAMBERT e MOUNIER 2006]PEPY, R.; LAMBERT, A.; MOUNIER, H. Path planning a dynamic vehicle model. *Information and Communication Technologies, ICTTA 06*, n. 2, p. 781–786, out 2006.
- [PIC18F2550 2007]PIC18F2550. 2007. Microchip Technology Inc. PIC18F2455/2550/4455/4550 Data Sheet: *28/40/44-Pin, High Performance, Enhanced Flash, USB Microcontrollers with nanoWatt Technology, 2007. DS39632D*.
- [Polaroid 1994]POLAROID. 1994. Polaroid, Data Sheet: *Technical Specifications For 6500 Series Sonar Ranging Module*.
- [RASSHOFER e GRESSER 2005]RASSHOFER, R. H.; GRESSER, K. Automotive radar and lidar systems for next generation driver assistance functions. *Advances in Radio Science*, v. 3, p. 205–209, 2005.
- [Reeds-Shepp Curves, 2009]REEDS-SHEPP Curves, 2009. Disponível em <http://msl.cs.uiuc.edu/lavalle/planning/node822.html>. Acessado em 01/10/2009.
- [Santos et al. 2008]SANTOS, M. M. et al. Automation and localization of a robotic car. In: *3rd Applied Robotics and Collaborative Systems Engineering (Robocontrol08)*. Bauru: [s.n.], 2008.
- [SCHEUER e FRAICHARD 1997]SCHEUER, A.; FRAICHARD, T. Continuous-curvature path planning for multiple car-like vehicles. In: *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. Grenoble: [s.n.], 1997.
- [SCHEUER e FRAICHARD 2004]SCHEUER, A.; FRAICHARD, T. From reeds and shepp's to continuous-curvature paths. *IEEE Transactions on Robotics*, v. 20, n. 6, p. 1025–1035, dez 2004.
- [SimpleH 2009]SIMPLEH. 2009. Robot Power, Washington, USA. *Simple-H User Manual*, 2009.