

Rafael Peixoto Derenzi Vivacqua

***Back Lane Marking Registry: uma Abordagem
de Localização e Seguimento de Caminho por
Veículos Autônomos em Via Sinalizada***

Vitória - ES

2017

Rafael Peixoto Derenzi Vivacqua

***Back Lane Marking Registry: uma Abordagem de
Localização e Seguimento de Caminho por Veículos
Autônomos em Via Sinalizada***

Universidade Federal do Espírito Santo – UFES

Centro Tecnológico

Programa de Pós-Graduação em Engenharia Elétrica

Orientador: Raquel Frizera Vassallo

Coorientador: Felipe Nascimento Martins

Vitória - ES

2017

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Setorial Tecnológica,
Universidade Federal do Espírito Santo, ES, Brasil)
Sandra Mara Borges Campos – CRB-6 ES-000593/O

V855b Vivacqua, Rafael Peixoto Derenzi, 1976-
Back Lane Marking Registry : uma abordagem de localização
e seguimento de caminho por veículos autônomos em via
sinalizada / Rafael Peixoto Derenzi Vivacqua. – 2017.
141f. : il.

Orientador: Raquel Frizera Vassallo.
Coorientador: Felipe Nascimento Martins.
Tese (Doutorado em Engenharia Elétrica) – Universidade
Federal do Espírito Santo, Centro Tecnológico.

1. Veículos autônomos – Direção de automóveis. 2. Visão por
computador. 3. Obstáculo – Localização. 4. Dead Reckoning
(Navegação). 5. Localização e mapeamento. I. Vassallo, Raquel
Frizera. II. Martins, Felipe Nascimento. III. Universidade Federal
do Espírito Santo. Centro Tecnológico. IV. Título.

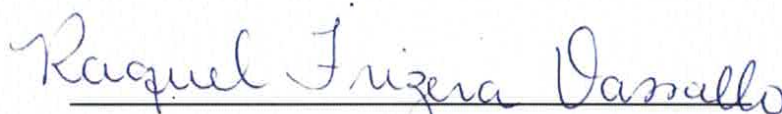
CDU: 621.3

Rafael Peixoto Derenzi Vivacqua

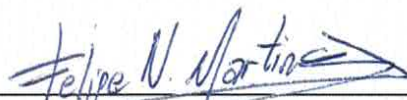
Back Lane Marking Registry: uma Abordagem de Localização e Seguimento de Caminho por Veículos Autônomos em Via Sinalizada

Tese submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Doutor em Engenharia Elétrica.

Aprovada em 23 de novembro de 2017.



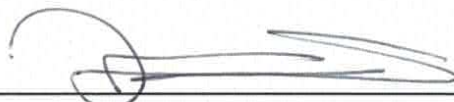
Dra. Raquel Frizera Vassallo - Orientadora
Universidade Federal do Espírito Santo



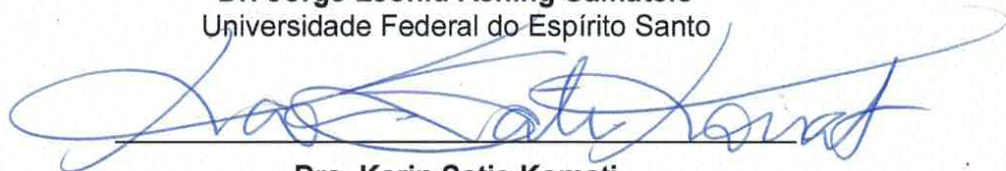
Dr. Felipe Nascimento Martins - Coorientador
Hanze University of Applied Sciences



Dr. Denis Fernando Wolf
Universidade de São Paulo



Dr. Jorge Leonid Aching Samatelo
Universidade Federal do Espírito Santo



Dra. Karin Satie Komati
Instituto Federal do Espírito Santo



Dr. Paulo Faria Santos Amaral
Universidade Federal do Espírito Santo

Vitória - ES
2017

*Este trabalho é dedicado às pessoas que se dedicam à ciência e engenharia com alegria,
amor e humildade.*

Agradecimentos

À minha querida e amada esposa Dagma, pelo apoio e incentivo ao meu trabalho e pela felicidade inspiradora de sua companhia.

A meus pais, irmãos e sobrinhos, simplesmente por vocês serem minha família.

A meu tio Paulo Augusto Vivacqua, que desde cedo, me despertou para os mistérios e maravilhas da ciência e da natureza.

A todos os meus queridos amigos, pela felicidade da convivência e renovação de energia.

À minha orientadora Raquel F. Vassallo por seu trabalho de orientação, por sempre ter acreditado em mim e nas minhas ideias, por ter defendido meu trabalho da falta de apoio que recebi de meu próprio programa de pós graduação.

Ao meu co-orientador Felipe N. Martins que sempre, prestativo e atencioso, ofereceu grande ajuda técnica e apoio emocional.

Ao professor Evandro O. T. Salles, pelo seu importante apoio ao meu trabalho em sua fase inicial, talvez sem ele eu nem tivesse começado.

Ao professor João Paulo A. Almeida e amigo de infância, que muito lucidamente me deu grande apoio técnico e emocional na fase final de publicação do trabalho.

Um agradecimento especial a todos que não atrapalharam.

Resumo

Esta tese teve o objetivo de contribuir para o desenvolvimento da tecnologia de direção autônoma. Mais especificamente, o objetivo desse trabalho foi projetar, construir e testar todo o *hardware e software* necessários à adaptação de um veículo comercial de modo que ele seja capaz de realizar com sucesso, repetidas vezes, uma missão de transporte pré-determinada, como se fosse um trem percorrendo uma ferrovia virtual. A questão dos obstáculos dinâmicos, igualmente importante e complexa, não foi tratada neste trabalho. Três abordagens amplamente utilizadas na literatura foram estudadas: (i) o seguimento de caminho direto por detecção das marcações de pista, (ii) o seguimento de caminho por localização global (GNSS), e (iii) o seguimento de caminho por mapas de marcadores de pista. Um veículo protótipo foi construído, respeitando a filosofia de baixo custo do projeto, para realização dos testes experimentais. As duas primeiras abordagens serviram de base para o desenvolvimento da abordagem final, baseada em mapas visuais de marcadores de pista. Esta abordagem produziu os resultados mais positivos do trabalho e contém a maior contribuição desta tese: o método denominado *Back Lane Marking Registry* ou BLMR. O BLMR permite construir uma percepção precisa, confiável e extensa das marcações de pista na vizinhança do veículo a partir de um arranjo de sensores de baixo custo. Esta extensa percepção combinada com o algoritmo rápido de filtragem e técnicas de *map matching* levou a uma precisão de localização alta o suficiente para aplicação em direção autônoma, o que foi demonstrado em testes realizados na Itália.

Palavras-chaves: Direção Autônoma, Veículos Autônomos, Carro auto-dirigido, Autopiloto Automotivo, Visão Computacional, Navegação Estimada, Detector de Faixas, Localização Visual, Mapeamento e Localização.

Abstract

This thesis aimed to contribute with the development of autonomous driving technology. More specifically, the aim of this work was to design, build and test all the hardware and software needed to adapt a commercial vehicle so that it is able to successfully perform a predetermined transport mission repeatedly, as if it was a train traveling on a virtual railway. The issue of dynamic obstacles was not addressed in this research. Three approaches widely used in the literature were studied: (i) direct path tracking by detection of lane markings, (ii) global localization tracking (GNSS), and (iii) path tracking by lane marking maps. A prototype vehicle was built, respecting the low cost philosophy of the project, to carry out the experimental tests. The first two approaches served as the basis for the development of the final approach, based on visual track marker maps. This approach has produced the most positive results of the work and represents the greatest contribution of this thesis: the method called Back Lane Marking Registry or BLMR. The BLMR allows the construction of accurate, reliable and extensive perception of lane markings in the vicinity of the vehicle from low-cost sensors. This extensive perception combined with a fast filtering algorithm and map matching techniques led to localization accuracy which is high enough for autonomous application, as demonstrated in tests conducted in Italy.

Key-words: Autonomous Driving, Autonomous Vehicles, Self-Driving Car, Automotive Autopilot, Computer Vision, Dead Reckoning, Lane Marking Detector, Visual Localization, Map Matching, Mapping and Localization.

Lista de ilustrações

Figura 1 – Evolução dos aviões e Automóveis em cem anos: (a) Bleriot 1909; (b) Airbus A380 de 2007; (c) Ford Modelo T de 1908; (d) Ford Fusion 2014.	18
Figura 2 – Definição do problema da direção totalmente autônoma.	19
Figura 3 – Evolução dos carros autônomos: (a) Propaganda do veículo autônomo na década de 60; (b) Projeto ARGO na década de 80; (c) veículo vencedor do desafio DARPA offroad em 2005; (d) Carro Autônomo da Google; (e) veículo italiano do grupo VisLab que dispensou o sensor LIDAR; (f) Mercedes S500 em fase de teste com direção automática.	22
Figura 4 – Princípio de operação de um sistema de posicionamento global. (VIRGINIATECH, ; LEARN, 2013)	27
Figura 5 – Carro Protótipo.	38
Figura 6 – Equipamentos instalados no carro protótipo.	39
Figura 7 – Módulo de interface.	40
Figura 8 – Detalhe da placa principal e da placa auxiliar.	41
Figura 9 – Diagrama esquemático simplificado do módulo de interface.	42
Figura 10 – Fluxograma do módulo principal de gerenciamento dos processos.	43
Figura 11 – Processo 1: Amostragem, controle de temperatura e geração de <i>PWM</i> .	44
Figura 12 – Operação do controle de temperatura do sensor inercial.	45
Figura 13 – Encapsulamento dos dados.	46
Figura 14 – Processo 2: Comunicação com o computador central.	47
Figura 15 – Processo 3: Medição de velocidade.	48
Figura 16 – Limitação da velocidade de giro do volante.	49
Figura 17 – Funcionamento do controlador do motor de passo.	50
Figura 18 – Processo 4: Controle do motor de passo (estado ativado).	51
Figura 19 – Classe <code>Car_Interface</code> .	51
Figura 20 – Exemplo do uso da API.	52
Figura 21 – Acionador do volante: detalhes de montagem.	53
Figura 22 – Circuito de controle do acelerador.	54
Figura 23 – Montagem da câmera.	56
Figura 24 – Receptor GPS.	56
Figura 25 – Arquitetura do programa seguidor de faixas.	59
Figura 26 – Transformação para monocromático: (A) imagem original; (B) conversão padrão; (C) conversão proposta, com ênfase na cor laranja.	60
Figura 27 – Convolução e detecção dos pontos da faixa.	61
Figura 28 – Manipulação das máscaras para cálculo rápido de convolução.	62
Figura 29 – Construção dos fragmentos de faixa.	64

Figura 30 – Mapeamento de um fragmento e transformação para polinômio.	66
Figura 31 – Visão da perspectiva inversa dos fragmentos.	66
Figura 32 – Teste de alinhamento entre dois fragmentos: (a) Conexão estabelecida ($\delta \leq \delta_{min}$); (b) Conexão não estabelecida ($\delta > \delta_{min}$)	67
Figura 33 – Exemplo de seleção dos dois melhores caminhos.	68
Figura 34 – Modelo da pista.	70
Figura 35 – Efeito do movimento sobre a posição dos objetos.	73
Figura 36 – Modelo Cinemático do carro.	73
Figura 37 – Rotação de uma parábola.	75
Figura 38 – Trajetos dos testes experimentais.	75
Figura 39 – Efeito da predição sobre os coeficientes do polinômio da linha central.	76
Figura 40 – Valor acumulado de EAH em um percurso de 100m.	77
Figura 41 – Caminho circular para atingir o alvo.	78
Figura 42 – Um quadro do vídeo demonstrativo do algoritmo de detecção visual operando: acima - imagem da câmera; abaixo Gráfico da qualidade do modelo; a direita projeção perspectiva inversa. Acessível em < http://youtu.be/mwSNEWhbJhg >	80
Figura 43 – Exemplo de reconstrução dupla de faixas.	81
Figura 44 – Percurso do teste com direção autônoma.	82
Figura 45 – Teste com direção autônoma, acessível em < http://www.youtube.com/watch?v=otWD6s8T3Js >.	83
Figura 46 – Arquitetura do sistema de direção autônoma por <i>dead reckoning</i> e GNSS.	86
Figura 47 – Transição de Estado.	87
Figura 48 – Influência da temperatura no <i>bias</i> do giroscópio.	89
Figura 49 – Erro causado pelo <i>bias</i> em um trajeto retilíneo.	89
Figura 50 – Efeito de variação do ganho de <i>update</i> no espalhamento dos pontos de referência.	91
Figura 51 – Caminho usado no teste de circuito fechado.	92
Figura 52 – Desvio acumulativo.	92
Figura 53 – Marcas para auxiliar o posicionamento inicial do veículo.	93
Figura 54 – Princípio de funcionamento do controlador do volante.	93
Figura 55 – Caminho dirigido em modo autônomo.	94
Figura 56 – Erros laterais: (a) Erro do controlador no corte A, (b) Erro real no corte A, (c) Erro do controlador no corte B, (d) Erro real no corte (B).	95
Figura 57 – Distorção geométrica (erro acumulativo) em nosso esquema de mapeamento baseado em <i>dead reckoning</i>	97
Figura 58 – (a) típica detecção frontal com alcance limitado pela precisão visual e (b) nossa proposta de detecção e reconstrução por <i>back registry</i> com o alcance limitado pela precisão do <i>dead reckoning</i>	97

Figura 59 – Mapeamento <i>online</i> : (a) A arquitetura do <i>software</i> . (b) Montagem experimental.	100
Figura 60 – Estrutura de dados do mapa e do BLMR: Caminho de referência e marcações de pista.	101
Figura 61 – O detector de marcação de pista de curto alcance operando: A informação visual entre 6,0m e 8,4m é usada para detectar os marcadores de pista.	103
Figura 62 – Exemplo da geração de mapa <i>online</i> : (a) foto de satélite do trecho mapeado; (b) marcações de pista resultantes.	104
Figura 63 – A arquitetura do <i>software</i> de localização <i>online</i>	105
Figura 64 – Estratégia de medição de postura: encontrar a transformação de corpo rígido que resulta na melhor sobreposição entre as marcações de pista do BLMR e do mapa.	107
Figura 65 – Seleção dos <i>Head points</i> candidatos: utilizar o mais próximos a posição atual do veículo. As marcações de pista não estão mostradas aqui para simplificar o desenho.	107
Figura 66 – Ajuste de orientação: cálculo do ângulo de rotação β necessário para alinhar os vetores MA e MB.	109
Figura 67 – Ajuste lateral: (a) calculo do desvio lateral; (b) após efetuado o ajuste.	110
Figura 68 – Erro de correspondência normalizado para diferentes secções de nosso trajeto de teste: (A) o BLMR contém uma curva acentuada; (B) o BLMR contém uma curva média; (C) o BLMR contém um trecho reto.	110
Figura 69 – Parâmetro de confiabilidade para a componente longitudinal da medição de postura.	111
Figura 70 – Direção autônoma baseada no seguimento de um ponto alvo no caminho de referência.	115
Figura 71 – Estradas escolhidas para a avaliação experimental da precisão.	116
Figura 72 – Registro de <i>dead reckoning</i> usado para estimativa do erro lateral do ponto alvo (δ_ξ).	117
Figura 73 – Precisão de localização: Gráficos violino (histogramas verticais) do erro lateral do ponto alvo (ELPA) das duas estradas testadas. A cruz corresponde ao valor médio.	118
Figura 74 – Veículos de teste do laboratório: (a) Deeva; (b) BRAIVE; (c) PORTER.	121
Figura 75 – Adaptações feitas no veículo PORTER: (a) Unidade de interface e giroscópio; (b) Encoder; (c) Montagem do encoder no carro.	122
Figura 76 – Testes de caminho fechado: (a) D.R. registrado pelo veículo PORTER; (b) Posições finais PORTER e BRAIVE.	123

Figura 77 – percurso mapeado para o teste de direção autônoma: (a) Mapa rodoviário; (b) mapa construído por <i>dead reckoning</i> ; (c) foto de satélite da curva 2; (d) marcações detectadas na curva 2.	124
Figura 78 – Fotos de um teste de direção autônoma.	126
Figura 79 – Exemplo de um trecho de uma estrada com diferentes características que afetam a localização longitudinal: (a) baixa incerteza, (b) media incerteza, (c) alta incerteza.	132

Lista de tabelas

Tabela 1 – Recursos utilizados do microcontrolador.	40
Tabela 2 – Características da câmera utilizada.	55
Tabela 3 – Efeito do uso da predição na redução do erro EAH.	77
Tabela 4 – Desempenho Computacional para um Quadro de Vídeo	80
Tabela 5 – Comparação de resultados reportados por trabalhos similares de localização por visão em veículos automotivos ($a, b, c, d, e, f =$ (HATA; WOLF, 2014; ZIEGLER et al., 2014; BADINO; HUBER; KANADE, 2011; LATEGAHN; STILLER, 2012; CUI; XUE; ZHENG, 2016; LYRIO et al., 2015)).	120
Tabela 6 – Desempenho Computacional: tempo gasto em cada processo do sistema de localização.	127

Sumário

1	INTRODUÇÃO	16
1.1	Motivação	16
1.1.1	Automóveis x aviões	16
1.2	Contextualização	19
1.3	História	20
1.3.1	Brasil	24
1.4	Estado da Arte	24
1.4.1	Localização por GNSS	26
1.4.1.1	Técnicas de melhoramento	28
1.4.2	Detecção de Marcadores de Pista	30
1.4.2.1	Detecção por Visão Computacional	31
1.4.3	Localização por Marcadores de Pista	33
1.5	Objetivos e Contribuições deste Trabalho	35
2	CONSTRUÇÃO DO VEÍCULO PARA TESTES	38
2.1	Introdução	38
2.2	Módulo de Interface	39
2.2.1	<i>Hardware</i>	40
2.2.2	<i>Firmware</i>	42
2.2.2.1	Processo 1: Amostragem de Sinais e PWM	43
2.2.2.2	Processo 2: Comunicação com o Computador Central	46
2.2.2.3	Processo 3: Medição de Velocidade	47
2.2.2.4	Processo 4: Controle do Motor de Passo	48
2.2.3	<i>User API</i>	51
2.3	Acionador do volante	53
2.4	Acelerador Eletrônico	54
2.5	Câmera	55
2.6	Receptor GPS	56
3	DIREÇÃO AUTÔNOMA BASEADA EM SEGUIMENTO DE MARCADORES DE PISTA	58
3.1	Introdução	58
3.2	Sistema de Visão	59
3.2.1	Conversão RGB-Mono com Filtragem de Cor	59
3.2.2	Binarização da Imagem	60
3.2.2.1	Algoritmo rápido de convolução	61

3.2.3	Associação dos Pontos	63
3.2.3.1	Construção das cadeias conectadas por 8	63
3.2.3.2	Projeção Perspectiva e transformação em polinômios	64
3.2.4	Construção da Melhor Representação	66
3.2.4.1	Conexão entre polinômios	67
3.2.4.2	Escolha dos melhores caminhos	68
3.3	Filtragem	69
3.3.1	Modelo da Pista	69
3.3.2	Atualização do Polinômio	70
3.3.3	Atualização da Largura	71
3.3.4	Atualização da Qualidade	71
3.3.5	Predição da Posição da Linha Central	72
3.3.5.1	Centro de Rotação e Deslocamento Angular	72
3.3.5.2	Rotação da Parábola	74
3.3.5.3	Avaliação Experimental	75
3.4	Controle do Volante	77
3.5	Testes e Resultados	78
3.5.1	Testes com Vídeo Gravado	79
3.5.2	Testes de Direção Autônoma	80
3.6	Conclusões	83
4	DIREÇÃO AUTÔNOMA BASEADA EM FUSÃO GNSS E <i>DEAD RECKONING</i>	86
4.1	Introdução	86
4.2	Sistema <i>Dead Reckoning</i>	87
4.2.1	Medição do Incremento Angular	88
4.3	Fusão com GNSS	89
4.4	Testes e Resultados	90
4.4.1	Desempenho do Sistema <i>Dead Reckoning</i>	91
4.4.2	Direção Autônoma Baseada em Fusão GNSS e <i>Dead Reckoning</i>	92
4.5	Conclusões	95
5	DIREÇÃO AUTÔNOMA BASEADA EM MAPAS DE MARCADORES DE PISTA	96
5.1	Introdução	96
5.2	Mapeamento <i>Online</i>	99
5.2.1	A estrutura do mapa	100
5.2.2	Dead reckoning	102
5.2.3	Detector de Marcações de Pista	102
5.3	Localização <i>Online</i>	104

5.3.1	O <i>Back Lane Marking Registry</i>	105
5.3.2	Medição da Postura	106
5.3.2.1	Translação e Rotação	108
5.3.2.2	Ajuste Fino Lateral	109
5.3.3	Filtro	111
5.3.3.1	Modo 1 - Localização Desconhecida	113
5.3.3.2	Modo 2 - Localização Aproximada	113
5.3.3.3	Modo 3 - Localização Precisa	114
5.4	Testes e Resultados	114
5.4.1	Precisão de localização	114
5.4.1.1	Metodologia para o <i>Ground truth</i>	116
5.4.1.2	Resultados	117
5.4.2	Direção Autônoma	119
5.4.2.1	Preparação dos Testes	120
5.4.2.2	Viagem de Mapeamento	123
5.4.2.3	Viagens com Direção Autônoma	125
5.4.3	Desempenho computacional	125
5.5	Conclusão e Discussão	127
6	CONCLUSÃO	130
6.1	Trabalhos futuros	131
6.2	Considerações sobre o Desempenho	132
	Referências	134

1 Introdução

1.1 Motivação

Todos os anos mais de um milhão de pessoas morrem vítimas de acidentes de trânsito no mundo. Em sua maioria, tais acidentes são provocados por falha humana (ASIRT, 2013). Neste cenário, tirar o controle do veículo das mãos do motorista humano e transferi-lo para um sistema automatizado tem o potencial de promover um ganho de segurança inestimável que salvaria uma grande quantidade de vidas e evitaria altos prejuízos financeiros com o tratamento dos acidentados e perdas materiais. Mas esse não é o único problema: o tempo gasto pelos motoristas com a direção poderia ser mais bem aproveitado com atividades de lazer (leitura, descanso, filmes) ou trabalho (telefonemas, leitura de emails, tele-reunião, estudos etc). Por exemplo, todo dia, para ir de casa para o trabalho dirigindo, o motorista tem de ficar totalmente dedicado à direção desde que entra no carro. Primeiro para tirá-lo da garagem evitando colidir com o portão; segundo, para acessar a via pública de transporte; terceiro, para conduzir o veículo no centro da pista nas vias públicas de transporte correspondente à rota desejada, ao mesmo tempo em que observa a sinalização e evita colisão com outros carros. Por último, para encontrar uma vaga e estacionar o veículo. Considerando-se, ainda, que o transporte terrestre é o que tem a maior quantidade de veículos no mundo e que a operação de cada um desses veículos requer a atenção total de um adulto produtivo, percebe-se então a gigantesca demanda por força de trabalho intelectual humano (homem-hora) requerida por esse modo de transporte de ampla escala.

Desta forma, a principal motivação deste trabalho de doutorado está no fato de se acreditar que ele poderá contribuir para o desenvolvimento de uma tecnologia que modificará profundamente, para melhor, a maneira como a sociedade se relaciona com o automóvel. Esta modificação poderá trazer ganhos enormes à sociedade como a redução de acidentes, redução do estresse e ganho de tempo livre.

1.1.1 Automóveis x aviões

Se for feita uma comparação entre evolução dos meios de transporte terrestre e aéreo no século XIX (Figura 1), vê-se que houve em ambos um tremendo avanço tecnológico. Mas se o foco for dado ao modo como os veículos são dirigidos ou pilotados, nota-se a diferença gigantesca entre os dois casos. Enquanto na aviação os pilotos contam com a ajuda dos sofisticados pilotos automáticos, os motoristas continuam a dirigir seus carros do mesmo modo como se fazia há 100 anos. O transporte terrestre ainda é o único que é

absolutamente dependente da atenção do motorista humano em todos os momentos.

No caso da aviação (Figura 1-a-b), pode-se ver os ganhos notáveis em todos aspectos. Por exemplo, em 1909 o avião Frances Bleriot XI (FACTORY, 2003) entrou pra história sendo o primeiro avião a atravessar o canal da Mancha, pesando 230kg, com um motor de 25CV (18,4kW) e velocidade máxima de 75km/h. Quanto aos sistemas de navegação, já em 1912 o primeiro piloto automático de um avião foi desenvolvido pela Sperry Corporation (WINGS, 2012). O piloto automático usava um giroscópio indicador de direção e um altímetro para hidraulicamente controlar o leme e o elevador de modo a permitir que o avião voasse em linha reta e nivelado no curso da bússola sem a atenção do piloto.

Atualmente o Airbus A380 (A380, 2014) é o maior avião de passageiros do mundo, com capacidade para transportar até 853 passageiros, possui quatro turbinas que geram 35 ton (343,2 kN) de empuxo cada, velocidade de cruzeiro de 900km/h com autonomia de 15700km. Os sistemas de piloto automático modernos mais complexos que equipam esse tipo de avião (ADVANCED..., 2009), conhecido como CAT-IIIb, já são capazes de assumir o controle do voo em todos os estágios: decolagem, subida, voo nivelado, descida, aproximação e aterrissagem.

No caso dos automóveis (Figura 1-c-d), em 1908 havia o Ford modelo T (T, 1908), considerado o primeiro automóvel produzido em grande escala. Este carro possuía um motor de combustão interna de 2,9 litros que desenvolvia 20 Hp (14,9kW) e o câmbio era manual de duas marchas. Passados mais de cem anos, temos o exemplo de um automóvel moderno, o Ford Fusion (CARROS, 2014), que possui um motor de 2,0 litros com 240CV (178,9kW) de potência, câmbio automático de 6 marchas, freios ABS, Airbag, sistema de navegação por GPS (apenas para indicação). Considerando-se apenas para o desempenho mecânico do automóvel moderno, pode-se afirmar que houve um imenso avanço tecnológico. Mas olhando-se com mais cuidado, percebe-se que esta evolução não aconteceu de forma tão profunda como no caso da aviação. Especialmente por duas razões: (a) ainda se utiliza na maioria esmagadora de veículos leves produzidos a mesma concepção de motor que equipava o Ford 1908, o motor de combustão interna de quatro tempos. (b) a direção do veículo ainda é totalmente dependente do motorista humano.

Recentemente, novas tecnologias estão sendo desenvolvidas e utilizadas comercialmente e tendem a reverter esse quadro, são elas: o automóvel elétrico e a tecnologia ADAS (Advanced Driver Assistance Systems). Por exemplo, a empresa americana de automóveis elétricos, Tesla, já fabrica e comercializa (MOTORS, 2014) para o mundo todo carros elétricos com autonomia superior a 400km e potência de 300 Hp (223,7 kW), superior a de muitos carros convencionais da mesma categoria. Em outro exemplo, temos a Mercedes Benz (INOVATION, 2014), considerada uma das líderes nas tecnologias de sistemas avançados de assistência ao motorista. Seus modelos mais luxuosos, já são equipados com

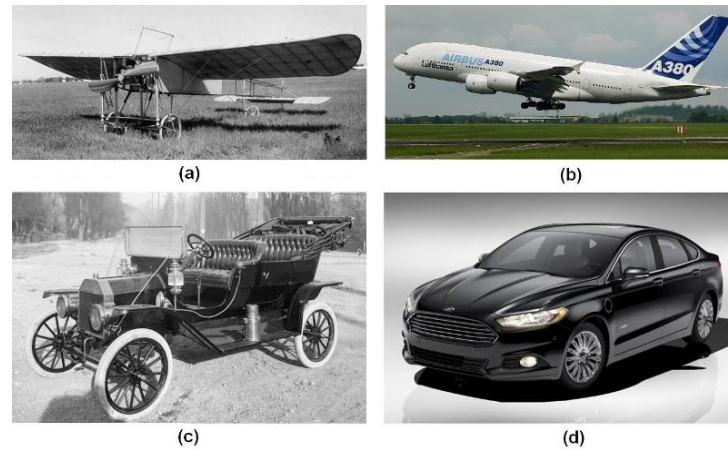


Figura 1 – Evolução dos aviões e Automóveis em cem anos: (a) Bleriot 1909; (b) Airbus A380 de 2007; (c) Ford Modelo T de 1908; (d) Ford Fusion 2014.

tecnologias como: o *Distronic plus with steering assist* que, além de ser um controle de velocidade adaptativo (se ajusta à velocidade do tráfego) também é um assistente de direção capaz de manter automaticamente o veículo no centro da pista, acionado quando o motorista tira as mãos do volante; o *Pre-Safe Brake With Pedestrian Detection*, um sistema de frenagem automática que usa câmeras estéreo e um sistema de radar para detectar pedestres e frear o carro a tempo de evitar ou minimizar os danos no caso de um atropelamento.

Esta grande diferença na história da evolução da pilotagem automática automotiva e aeronáutica pode ser justificada pelas seguintes razões: (a) a pilotagem automática automotiva exige um grau de precisão muito superior do que a aeronáutica onde um desvio na ordem de centímetros pode tirar o veículo da estrada; (b) os automóveis frequentemente trafegam em condições de tráfego denso onde a distância média entre os veículos é pequena; (c) os automóveis estão sujeitos a diversos obstáculos móveis como carros em cruzamentos, bicicletas e pedestres.

Os sistemas de localização global a base de satélites como GPS ou Galileu apresentam erros de localização na ordem de poucos metros. No caso da aviação, este desvio não prejudica a operação em modo automático porque a distância entre os aviões e a largura das aerovias é muitas vezes maior do que este erro. No caso automotivo, isso não é verdade e os automóveis não podem aproveitar os benefícios desta conveniente, confiável e barata tecnologia de localização. Isto fez com que a direção autônoma de um veículo em uma estrutura pública de transporte convencional fosse considerada um dos maiores desafios da automação moderna. Espera-se que, no futuro, em seu extremo de desenvolvimento, a direção autônoma deva ser capaz de conduzir de forma completamente autônoma um veículo do endereço A ao B com a mesma segurança de um experiente motorista humano. Nessas condições, o usuário torna-se apenas um passageiro.

1.2 Contextualização

A direção completamente autônoma de um veículo na estrutura pública de transporte ainda é considerada um grande desafio de automação. Apesar da grande capacidade dos computadores atuais a humanidade ainda não conseguiu resolver esse problema em uma escala comercial. O problema é que a direção totalmente autônoma exige alta precisão, confiabilidade e disponibilidade em um ambiente complexo e dinâmico.

A Figura 2 apresenta a definição do problema da direção totalmente autônoma. Acima tem-se o usuário que informa ao navegador o destino desejado. O navegador por sua vez usa um receptor GNSS para determinar sua localização atual, e uma base de mapas rodoviários (baixa precisão) para calcular uma rota eficiente. Esta rota então é enviada ao "motorista robô" que usa mapas de localização de alta precisão e sensores como LIDAR, cameras, INS, para efetivamente dirigir o carro no meio da pista, respeitando os semáforos e reagindo aos obstáculos móveis. É o "motorista robô" propriamente dirige o veículo atuando na posição do volante, acelerador, freio, seta, etc. Dentro deste contexto, o trabalho desenvolvido nesta tese está focado exclusivamente no problema da localização em nível de pista para o seguimento autônomo de caminho.

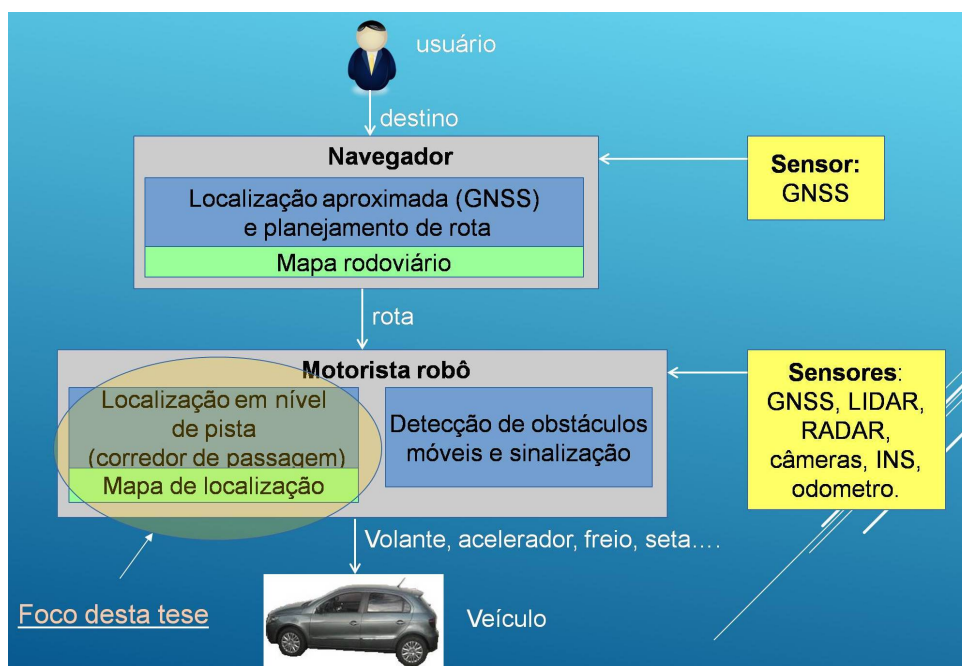


Figura 2 – Definição do problema da direção totalmente autônoma.

O cálculo de rota já é um problema bem resolvido, com a disponibilidade de diversos navegadores comerciais e aplicativos de *smartphone*. O navegador tem a função de, dado a posição atual e o endereço de destino, calcular a rota. Este tipo de aplicação não exige alta precisão de modo que a localização por GNSS e o uso de mapas rodoviários de baixa precisão não é um problema.

Entretanto, o controle do veículo, a direção propriamente dita, é a parte mais crítica do problema porque exige alto grau de precisão de localização (muito maior que de GNSS) e a capacidade de detecção e rápida reação aos obstáculos móveis e sinalização dinâmica (semáforos). É importante ressaltar que todas essas tarefas devem ser feitas com o mais alto grau de confiabilidade pois qualquer falha pode resultar em danos graves ou até mesmo morte. A localização em nível de pista e seguimento de caminho é um requisito fundamental para a direção totalmente autônoma. Esta tese dedica-se a contribuir para a solução deste problema.

A localização em nível de pista deve possuir erro lateral na ordem de poucos centímetros de modo a permitir que o controlador do volante possa manter o veículo se deslocando precisamente no centro da pista (corredor rodoviário). Esta tarefa, denominada seguimento autônomo de caminho, poderia ser resolvida de maneira relativamente fácil se o sistema GNSS tivesse precisão alta suficiente, porém não é o caso. Apesar de existirem sistemas GNSS de alta precisão (GNSS-RTK) que, em certas condições de operação, obtêm o nível de precisão suficiente, tais sistemas exigem uma infraestrutura de estações de base e não funcionam bem em ambientes urbanos densamente construídos. Por isso os projetos mais avançados têm usado as informações do ambiente rodoviário, tais como a sinalização pintada no chão, árvores ou construções do entorno, para localizar o veículo. Conforme será discutido mais amplamente nas seções que seguem, estes projetos usam câmeras e/ou LIDAR (sensor Laser) para detectar os objetos de localização (*landmarks*) e usam algum tipo de filtragem para estimar a localização do veículo no mapa. Uma vez que os mapas devem ser de alta precisão, e cada projeto utiliza *landmarks* específicos e uma estrutura de dados particular, estes mapas são construídos pelo próprio veículo usado em cada projeto.

Esta tese apresenta uma abordagem de localização e seguimento em vias sinalizadas. Ela utiliza como *landmarks* as marcações de pista pintadas no chão para delimitar cada pista de rodagem. Diferente dos outros trabalhos, esta abordagem primeiro acumula as *landmarks* em uma estrutura de dados denominada *Back Lane Marking Registry* (BLMR) para depois utilizá-la para localização. Esta estratégia permitiu que o algoritmo desenvolvido fosse leve computacionalmente ao mesmo tempo que preciso e robusto.

1.3 História

Apesar desta tecnologia estar amadurecendo apenas nos dias atuais, a ideia da direção automática de um veículo já é bem antiga. Em 1958 um sistema de direção autônoma desenvolvido pela RCA Labs foi apresentado em público com uma demonstração bem sucedida em uma estrada pública em um percurso de 120 metros. O veículo usava uma série de circuitos instalados sob o pavimento para se guiar. Na época, esperava-se

que o sistema fosse entrar em operação comercial em 1975 (CAR, 1960).

Na década de 1980 iniciou-se a era dos veículos autônomos guiados por visão computacional, com o professor Ernest Dickmanns da Bundeswehr University of Munich em parceria com a Mercedes-Benz. Considerado um dos pioneiros na direção autônoma, o professor Dickmanns demonstrou um sistema de direção autônoma baseado em visão capaz de atingir 63km/h. Entre os anos de 1987 e 1995, a então recém criada organização intergovernamental europeia, EUREKA, investiu 749 milhões de euros no projeto PROMETHEUS (Programme for a European Traffic System With Highest Efficiency and Unprecedented Safety) (EUREKA, 1987), com o objetivo de desenvolver novas tecnologias, como a do carro autônomo, que contribuíssem para aumentar a eficiência e a segurança no sistema de transporte terrestre. Em outubro 1994 ao final do projeto PROMETHEUS, foi feita a demonstração bem sucedida do veículo VaMP (VAMP, 1994), desenvolvido pela equipe do Prof. Ernest Dickmanns que dirigiu mais de 1000km em uma estrada de pista tripla em Paris a velocidade de até 130km/h.

A partir da década de 1980 a quantidade de projetos na área de veículos terrestres autônomos continuou a crescer e novos projetos surgiram em diversos países do mundo. Em 1995, o projeto NavLab (POMERLEAU; JOCHEM, 1996), da universidade americana Carnegie Mellon, realizou o desafio denominado “No Hands Across America”, onde o veículo dirigiu mais de 5000km com 98,2% do percurso feito em modo autônomo. Assim como nos projetos (SCHNEIDERMAN; NASHMAN, 1994), (BROGGI et al., 1999), e (LU et al., 2002), este veículo usava um sistema de visão computacional com base em uma câmera monocromática, capaz de calcular a posição da estrada em relação ao veículo e estimar sua curvatura. Apesar do uso de imagens monocromáticas ser mais leve computacionalmente, esta abordagem descarta parte da informação visual disponível, a cor. Em (CRISMAN; THORPE, 1989) a imagem colorida de uma câmera foi usada para classificar as regiões pertencentes à estrada ou não.

Em 2002 a agência de defesa de projetos de pesquisa avançados (DARPA), anunciou a primeira de uma série de competições de veículos robóticos, com o objetivo de impulsionar o desenvolvimento desta tecnologia. A sua primeira edição (DARPA, 2004) foi realizada em 2004, no Deserto de Mojave, na Califórnia - EUA. O desafio final era a direção autônoma por um percurso de de 240km no deserto de Mojave - Califórnia. Nenhuma equipe conseguiu completar o percurso, sendo a equipe que mais avançou completou apenas 11,8km. O prêmio de 1 milhão de dólares ficou acumulado. Em 2005 a agência de defesa de projetos de pesquisa avançados (DARPA), financiou a competição de veículos robóticos DARPA Grand Challenge (DARPA, 2005), que constituiu um marco importante na história da evolução dos veículos autônomos. O desafio final era uma corrida offroad num percurso de 212km, realizado no deserto de Mojave, sudeste do EUA. O percurso da prova foi fornecido aos participantes 2 horas antes do início da prova na forma de um

arquivo com uma coleção de pontos de referência (*waypoints*). Os veículos teriam que se auto-dirigir sem nenhuma intervenção humana ao longo desse caminho, mantendo-se no centro da estrada e desviando de obstáculos. Das 195 equipes que se inscreveram, 23 se classificaram para a final e 5 completaram o percurso. A equipe Stanford Racing Team, da universidade de Stanford, venceu a prova com o tempo de 6h53min seguida da equipe Red Team, da universidade Carnegie Mellon com o tempo de 7h05min. O veículo da equipe vencedora (THRUN et al., 2006) usou um arranjo multi-sensorial para percepção do ambiente, formado por 5 sensores LASER (de varredura simples), uma câmera colorida, um radar de microondas, receptor GPS, unidade de navegação inercial e odometria das rodas. A informação gerada por estes sensores foram combinadas para criar um mapa 2D do ambiente que permitisse ao veículo identificar o centro da estrada e a posição dos obstáculos. O estado de localização global (coordenadas, orientação e velocidade) do veículo era estimado por um filtro de Kalman a partir das informações recebidas do sensor GPS, da unidade inercial e do odômetro. O sistema de visão computacional do veículo usava um esquema probabilístico de classificação de pixels por cor (DAHLKAMP, 2006). Este sistema de visão permitiu aumentar o alcance de percepção do veículo para além do alcance dos sensores LASER permitindo ao veículo operar em velocidades mais altas.

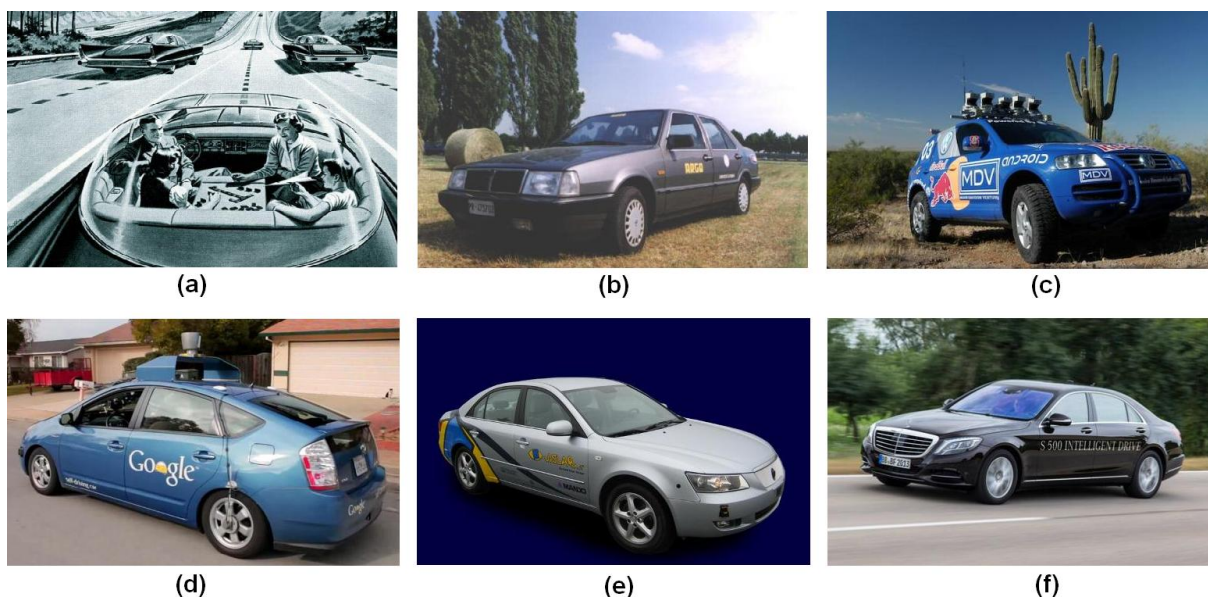


Figura 3 – Evolução dos carros autônomos: (a) Propaganda do veículo autônomo na década de 60; (b) Projeto ARGO na década de 80; (c) veículo vencedor do desafio DARPA offroad em 2005; (d) Carro Autônomo da Google; (e) veículo italiano do grupo VisLab que dispensou o sensor LIDAR; (f) Mercedes S500 em fase de teste com direção automática.

Outro marco importante na história dos veículos autônomos foi a demonstração da capacidade dos veículos operar com segurança em ambiente urbano na presença de outros veículos e respeitando a sinalização de trânsito. Isso foi demonstrado por diversas equipes na competição DARPA Urban Challenge realizada em 2007 ao longo de 96 km de

prova em um ambiente urbano controlado. Os cinco primeiros colocados na competição estavam equipados com um novo tipo de sensor LIDAR com varredura de múltiplos feixes que permite detectar diretamente objetos ao redor do veículo com cobertura de 360°. Nos anos subsequentes projetos que se destacaram nesta competição como ([SPECTRUM, 2011](#)) e ([LABS, 2011](#)) realizaram missões de navegação autônoma bem sucedidas em ambientes urbanos reais.

O progressivo aumento da capacidade de processamento e memória dos sistemas de computação permitiu que algoritmos de visão computacional cada vez mais complexos pudessem ser executados com velocidade suficiente para aplicação em tempo real. Um sistema binocular de visão (denominado visão estéreo) foi usado em ([BROGGI et al., 2010](#)) e ([LIMA; PEREIRA, 2011](#)) para detecção de obstáculos. Este tipo de sistema de visão permite que a informação de profundidade seja calculada a partir da informação combinada de duas câmeras. Em 2013, o laboratório VisLab da universidade de Parma fez uma demonstração pública do seu veículo de testes, chamado BRAiVE, se auto-dirigindo em condições de tráfego variadas, como estradas rurais, rodovias e trechos urbanos ([VISLAB, 2013](#)). Este veículo (Figura 3-e) abandonou o uso do sensor LIDAR e substituiu suas funções por um sofisticado sistema de visão estéreo. Também em 2013 a Mercedes Benz fez a demonstração pública "Autonomous Long-Distance Drive" onde um veículo protótipo (Figura 3-f) fez uma viagem autônoma com 100km de distância em rodovias e trechos urbanos ([S500, 2013](#)). A fábrica americana de veículos elétricos, Tesla, atualmente comercializa o modelo S. Este modelo usa câmeras e radar para dirigir automaticamente em *highways*, fazendo troca de faixa e ajuste de velocidade de acordo com o tráfego ([TESLAMOTORS, 2016](#)). A montadora Volvo, através do seu projeto "Drive Me", desenvolveu sua tecnologia de direção autônoma denominada IntelliSafe. Esta tecnologia está em seus últimos estágios do desenvolvimento da tecnologia IntelliSafe e a montadora anunciou que em 2017 entregará 100 unidades para clientes testarem na cidade de Gothenburg - Suécia ([VOLVOCARS, 2016](#)).

A retirada do sensor LIDAR do veículo parece ser a tendência natural da evolução dos veículos autônomos. Sem o LIDAR, os veículos autônomos mantêm sua aparência original e, mais importante, o custo da automatização fica bem mais baixo.

Em resumo, o que se observa ao longo da história da evolução dos veículos terrestres autônomos é que, nos primeiros projetos os veículos se comportavam basicamente como seguidores de caminho operando em estradas pavimentadas, e usavam visão computacional monocular para se guiarem. Na medida em que os projetos se sofisticaram, com o surgimento de novos sensores (LIDAR, GPS e IMU) e aumento da capacidade de computação, técnicas de mapeamento e localização começaram a ser empregadas de modo que o veículo tivesse uma compreensão bem mais ampla do ambiente ao seu redor. Por último, o aumento da capacidade de processamento tem favorecido a substituição dos

sensores LIDAR por sistemas mais baratos a base de visão.

1.3.1 Brasil

No Brasil este campo de pesquisa é relativamente novo, mas podemos destacar alguns projetos: o projeto CADU desenvolvido na UFMG (SABBAGH et al., 2010), foi um dos projetos pioneiros do Brasil, desenvolvido sobre um Chevrolet Astra devidamente equipado com sensores e atuadores; o projeto CaRINA 2, desenvolvido pelo Laboratório de robótica móvel (LRM) da USP - São Carlos (ICCM-USP, 2010) preparou um Fiat Palio Adventure com sensores e atuadores para o desenvolvimento de algoritmos de direção autônoma, também em 2015 LRM apresentou publicamente o projeto do Caminhão autônomo desenvolvido em cooperação com a empresa Scania (SCANIABRASIL, 2015); o projeto do laboratório de computação de alto desempenho da UFES (LCAD, 2010) utiliza uma plataforma de testes comercial (ByWire XGV) para testes e desenvolvimento de algoritmos de direção autônoma. Em 2015 fez uma demonstração de uma volta autônoma no campus da UFES e em 2017 uma viagem autônoma entre as cidades de Vitória e Guarapari;

1.4 Estado da Arte

A direção autônoma é o mais alto nível de automação para um veículo, que significa que o veículo pode se auto-dirigir de um ponto de partida até um destino sem intervenção humana. O problema pode ser dividido em duas tarefas principais. A primeira tarefa é a capacidade de manter o veículo se movendo ao longo de uma trajetória correta. Para realizar com sucesso esta tarefa, a determinação precisa da localização do veículo em relação à estrada é um requisito obrigatório. A segunda tarefa é a capacidade de perceber e reagir a eventos imprevisíveis, como por exemplo, a presença de outros veículos, pedestres, ou a condições da sinalização de tráfego.

Atualmente os projetos de direção autônoma mais bem sucedidos (PATON et al., 2017; KIM et al., 2017; ZIEGLER et al., 2014; LEVINSON; MONTEMERLO; THRUN, 2007; LEVINSON; THRUN, 2010; BERTOZZI et al., 2011) são baseados em localização precisa em ambientes mapeados. Alguns deles têm demonstrado o potencial desse método com a realização de viagens autônomas em ambientes reais e complexos. Em geral um mapa contém, no mínimo, um conjunto de *landmarks* usadas para localizar o veículo e um caminho de referência a ser seguido. Se a técnica de localização alcançar precisão suficiente em qualquer parte do mapa, o problema da direção autônoma pode ser reduzido a um problema de seguimento de caminho.

Algoritmos de localização usam marcações fixas (*landmarks*) no chão para localizarem o veículo. No espaço rodoviário, as marcações horizontais de tráfego, tais como

marcação de pista, faixas de parada, faixas de pedestre são opções bem convenientes para serem usadas como *landmarks*. De um modo geral, as marcações horizontais estão amplamente disponíveis, elas estão em todo lugar incluindo bem próximo ao veículo, são relativamente fáceis de serem detectadas, e sofrem pouca variação da aparência com o tempo. Os sensores amplamente usados para detectar as marcações horizontais são os sensores LIDAR e câmeras.

Sensores LIDAR são amplamente usados em projetos de direção autônoma. Eles são capazes de realizar medições diretas de distância e reflectância. Em (XUE et al., 2017; KIM et al., 2017; LEVINSON; MONTEMERLO; THRUN, 2007; DIETMAYER et al., 2005; HATA; WOLF, 2014) a informação de reflectância foi usada para detecção de marcações de faixas (*lane marking*). O LIDAR oferece grandes vantagens, tais como ângulo de detecção de 360° em torno do veículo e dados mais simples de serem processados quando comparado com visão computacional. Outra grande vantagem do LIDAR sobre as câmeras, é que ele emite sua própria radiação (LASER), e portanto, não fica dependente da condição de iluminação externa que, em certos casos, pode até impedir a detecção por cameras. A desvantagem do LIDAR é o seu alto preço e o fato de que precisa ser instalado no exterior do veículo, em posições específicas, o que requer modificações na forma externa do veículo.

Abordagens baseadas em visão computacional são amplamente usadas para detecção de estradas ou marcações de pista (XUE et al., 2017; KIM et al., 2017; YANG et al., 2017; BERTOZZI; BROGGI; FASCIOLI, 2006; GOPALAN et al., 2012; BORKAR; HAYES; SMITH, 2012; CUI; WANG; LI, 2014). As câmeras geralmente são compactas e fáceis de integrar ao veículo. Além disso, com a exceção de dispositivos especiais para detectar imagens no espectro infra-vermelho, as câmeras são equipamentos baratos. Outro ponto interessante é que a imagem obtida por uma câmera registra uma quantidade enorme de informação. De fato elas são capazes de registrar informação de textura e cor dos objetos capturados e ainda podem ser usadas para detectar a superfície de uma estrada. Todas estas características fazem das câmeras um sensor muito versátil que pode ser aplicado em diferentes tarefas. Por outro lado, a extração desta informação é um processo computacionalmente pesado. Desse modo, a escolha da máquina de processamento tem de levar em conta as tarefas extras de processamento para a visão computacional.

Para localização visual em locais sem marcações horizontais qualquer outro ponto fixo que tenha grande contraste visual pode ser usado como *landmark*. SURF (*Speed Up Robust Feature*) (BAY et al., 2008) é um algoritmo amplamente usado para detecção e descrição de tais pontos. Por exemplo, em (PATON et al., 2017) um algoritmo visual de seguimento de rota é apresentado para operação em ambientes *offroad* sujeitos a variação de iluminação lenta. O sistema de visão estéreo detecta pontos característicos (*features*) em múltiplos canais de dados (escala de cinza, croma) para estimar a localização do veículo

no mapa. Os testes de seguimento autônomo de rota, realizados no verão e no inverno, demonstraram que o uso de multi-canais de dados melhorou a robustez do sistema contra variação de iluminação. Em (ZIEGLER et al., 2014) um sistema duplo de visão é formado por um sistema de visão traseiro, que usa SURF para detectar pontos característicos visuais e, um sistema de visão frontal que detecta as marcações de faixa da estrada. Os testes realizados demonstraram que o uso de *landmarks* complementares contribuiu para melhorar a precisão da localização e a robustez.

Diversas são as estratégias para direção autônoma, desde as mais simples, como os detectores de saída de pista com atuação automática na direção, tratores autônomos baseados em localização por GNSS de precisão, até os mais sofisticados que operam em condições de tráfego real e usam sofisticados mapas com informações multisensoriais e detecção de obstáculos móveis.

No trabalho desenvolvido nesta tese, a abordagem adotada para direção autônoma é baseada na localização usando mapas de marcações de pista. O sucesso dessa abordagem depende fortemente do desempenho do detector de marcações de pista e do método de localização. Na sequência serão apresentados alguns trabalhos relacionados no campo de direção autônoma a base em GNSS, outros a base de detecção de marcações de pista para direção autônoma direta (seguidores de pista) ou indireta, ou seja, através de mapas e técnicas de localização.

1.4.1 Localização por GNSS

Uma maneira relativamente simples de fazer direção autônoma é por meio de um mapa rodoviário e um meio de localização. A localização poderia ser fornecida diretamente por um receptor GNSS (*global navigation satellite system*), porém mesmo os mais precisos sistemas de localização baseado em GNSS, tais como, GNSS com correção RTK, GNSS e IMU (*inertial measurement unit*), ou mesmo esquemas sofisticados de multi-medidas em veículos conectados em rede (SHEN et al., 2017) ainda não conseguem alcançar o nível de precisão e confiabilidade em ambientes urbanos densos. Em outras palavras, se o sistema GNSS tivesse precisão centimétrica, o tempo todo, os navegadores de GNSS já disponíveis nos carros, não apenas orientariam o motorista para dirigir na rota, eles dirigiriam o carro.

Os sistemas de posicionamento global (GPS americano, o GLONASS russo, o Galileu da União Européia e o Compass chinês) calculam a posição global (3D) de um objeto através da medição de sua distância em relação a no mínimo quatro satélites, conforme o esquema da Figura 4-a. O conjunto de 27 satélites (no caso do sistema GPS) fica orbitando a Terra de modo que para qualquer posição do globo sempre haja, no mínimo, seis satélites visíveis no céu.

A distância a cada satélite é medida através do tempo de propagação de um sinal eletromagnético emitido pelo satélite. Portanto, a questão crítica para a precisão é a correta medição de distâncias que estão na ordem de milhares de km. As principais causas de erro que afetam as medições de distância, e consequentemente a precisão da localização, são: condição da atmosfera que alteram a condição de propagação das ondas de rádio; condições do entorno tais como obstrução do sinal, e múltiplas reflexões de sinal *multipath*; disposição geométrica dos satélites e do receptor; limitações eletrônicas do receptor.

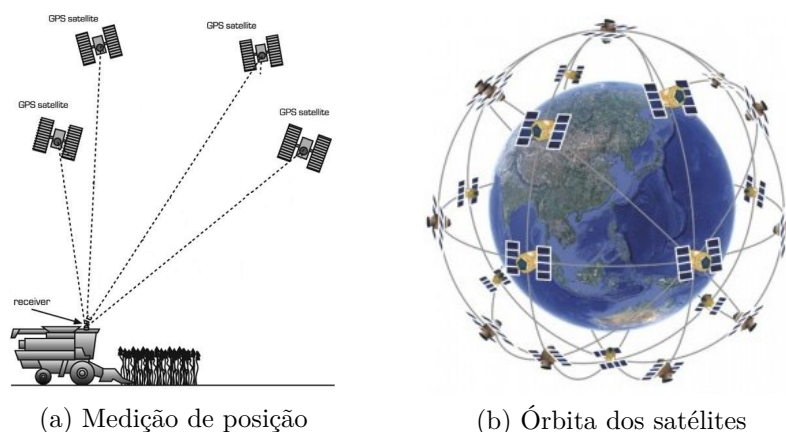


Figura 4 – Princípio de operação de um sistema de posicionamento global. (VIRGINIA-TECH, ; LEARN, 2013)

Como as causas de erro são variadas e aleatórias, obviamente o erro é um processo estocástico, apropriadamente caracterizado por uma função densidade de probabilidade. Normalmente na literatura leiga e, em alguns casos, na literatura técnica a precisão de um sistema GNSS é divulgado apenas como um valor, em metros. Por exemplo, o erro do sistema GPS é popularmente conhecido como algo entre 5m e 15m, sem que se compreenda que tipo de erro se trata (médio, percentil 95, erro máximo, etc). Na verdade o erro do sistema GPS, segundo documento oficial publicado pelo governo dos EUA que controla o sistema GPS (*Global Positioning System Standard Positioning Service Performance Standard*), é menor que 7,8 metros 95% do tempo. Para operação na Terra, obviamente este valor de erro aumenta devido à influência da condição atmosférica, que afeta o tempo de propagação das ondas de rádio. O uso de duas portadoras de sinal de frequências diferentes, como no caso dos receptores militares de GPS, permite compensar grande parte da distorção causada pela atmosfera, mas o máximo de precisão só é obtida com o uso de técnicas de melhoramento.

As técnicas de melhoramento, tais como cálculo de média, fusão com INS, WAAS, WADGPS, DGPS, RTK, melhoram muito a precisão básica do sistema GNSS, mas todos esses métodos não são adequados para operar em ambiente urbanos densamente ocupados por construções que degeneram os sinais de rádio. Por outro lado, em ambientes rurais,

devido à grande visibilidade do céu, o uso de GNSS melhorado já é explorado comercialmente na direção autônoma de máquinas agrícolas.

1.4.1.1 Técnicas de melhoramento

Em condições com boa visibilidade do céu e longe de construções altas, a precisão do posicionamento pode ser bastante melhorada usando o sinal GPS combinado com sistemas de melhoramento (*augmentation systems*). Algumas técnicas alcançam precisão de localização em tempo real de poucos centímetros, e medidas pós-missão em nível milimétrico. Por exemplo, o sistema WAAS (*Wide Area Augmentation System*), desenvolvido pela FAA (*Federal Aviation Administration*) para permitir a moderna pilotagem automática de aviões, usa um conjunto de estações de referência em terra para calcular um sinal de correção. Este sinal é enviado através de uma rede de satélites geostacionários para os receptores que possuem esta função. Dados coletados pela FAA em 2011 mostraram que a precisão horizontal ficou frequentemente próxima de um metro. Apesar de ser um valor bem baixo, ainda não é suficiente para operação de um carro em uma avenida, onde a precisão deve ser de poucos centímetros.

Outro sistema de melhoramento de precisão baseado em estações terrestres é o DGPS (diferencial GPS). Neste sistema, uma estação de referência transmite por uma antena terrestre um sinal de correção para os receptores na proximidade. Este sinal compensa os efeitos de erro devido à condição atmosférica, e quanto mais próximo da estação estiver o receptor, melhor será a precisão. Também usando este conceito de correção por estações base, o sistema RTK (*Real Time Kinematic*) usa a medição da fase da onda portadora, em vez do sinal modulante, para calcular as distâncias do receptor aos satélites. Como o comprimento de onda da portadora L1 do sistema GPS é de apenas 19 cm, e considerando uma precisão de medição de fase de 1% isso resultaria em um erro de 1,9 mm na medição de distância. Mas por questões práticas a precisão deste sistema fica na ordem de poucos centímetros. Com este valor de precisão, do sistema GNSS com correção RTK certamente seria suficiente para aplicação em direção autônoma, tendo em vista que o desvio lateral de poucos centímetros em relação a uma via de transporte é perfeitamente aceitável. Porém esta tecnologia tem a desvantagem de depender de uma densa rede de estações base pois quanto mais longe estiver o veículo de uma estação base, maior será o erro. Além disso esta tecnologia falha no caso de ambientes urbanos, pela razão a seguir.

Os sistemas de melhoramento de precisão por sinal de correção compensam o erro causado pela atmosfera através de estações de referência fixas e sinais de correção auxiliares. Mas esta tecnologia não consegue reduzir outra forte causa de erros de posicionamento: a condição do entorno próximo. Quando um receptor está operando dentro de uma cidade, onde há construções altas nas proximidades, o sinal recebido de alguns satélites pode ser temporariamente obstruído, sofrer refração ou ainda ser recebido com

múltiplas reflexões. Todos estes fatores contribuem para o aumento do erro de localização, que pode superar 50 m em certas posições em certos instantes. O grande problema neste caso, é que não se pode calcular um sinal de correção aplicável a uma grande área. Cada posição dentro de uma cidade sofre os efeitos de degradação do sinal de maneira diferente.

A forma mais simples (pois não depende de sinais de correção) de melhorar muito a precisão do posicionamento é o simples cálculo da média temporal. Devido à natureza aleatória dos erros, o cálculo da média converge para o valor correto. Porém esta técnica exige que o receptor fique imóvel durante longos intervalos de tempo, portanto não sendo adequado a operação móvel. Esta limitação pode ser superada se forem usados, junto com o GNSS, sensores de movimento incremental (*dead reckoning*). Esta técnica já é usada por fabricantes automotivos, como a Toyota, nos navegadores de seus automóveis, e permite a operação do navegador mesmo durante certos períodos sem o sinal GNSS (ex. túneis).

A fusão de GNSS com sensores proprioceptivos de *dead reckoning*, tais como unidades inerciais e/ou odometria, aumenta significativamente a precisão da localização global, mesmo sem o uso de sinais externos de correção. Isso acontece porque as melhores características de cada sistema se complementam. Por um lado temos a medição absoluta de posição pelo GNSS com erro alto, porém limitado, e o por outro lado temos a alta precisão de medição relativa de posição via *dead reckoning*, porém com erro absoluto ilimitado. A fusão normalmente é feita em um filtro que une as medidas dos dois sistemas de sensores através de operações de predição e de atualização. A predição corresponde ao cálculo de *dead reckoning* usando as informações de deslocamento incremental, e a atualização (ou *update*) é a correção da posição absoluta a partir da informação de GNSS.

A estrutura do filtro pode ainda ser probabilística de modo a considerar o erro de cada sensores. Este tipo de abordagem permite não somente calcular a posição atual do veículo, mas também o grau de confiabilidade desta estimativa, em cada instante. Entre as abordagens probabilísticas mais utilizadas temos o filtro de partículas e, para o caso de sensores com erro gaussiano, o filtro de Kalman. Por exemplo, em (THRAPP; WESTBROOK; SUBRAMANIAN, 2001), (THRUN et al., 2006) e (SPANGENBERG; CALMETTES; TOURNERET, 2007) foi usado um filtro de Kalman para fundir as informações dos sensores incrementais (odômetro, giroscópio e acelerômetros) com as medições do GPS. Soluções comerciais também usam esse tipo de estratégia. A empresa Ublox comercializa um *chip* de posicionamento global para aplicações automotivas em ambientes urbanos (UBLOX, 2014). Este chip possui um receptor duplo que opera simultaneamente com dois sistemas de posicionamento global (ex: GPS e GLONASS), giroscópio e acelerômetros de 3 eixos. O processador interno roda um algoritmo de fusão desta informação para implementar *Dead Reckoning 3D*, o que permite uma grande melhora na precisão da localização mesmo nos instantes em que o sinal de localização global não está disponível.

1.4.2 Detecção de Marcadores de Pista

A detecção de marcadores de pista pode ser usada diretamente para conduzir autonomamente um veículo, uma vez que fornece informação de orientação e localização lateral em relação à estrada. A principal desvantagem de um sistema de condução autônoma baseado puramente nas marcações de pista é a sua incapacidade de lidar com estruturas rodoviárias complexas, como trechos curtos de falha na marcação, rotatórias, bifurcações ou junções. No Capítulo 5 desta tese é apresentado um método para superar essas limitações com o uso de técnicas de mapeamento e localização. No Capítulo 3 apresenta-se o desenvolvimento de um detector visual de marcadores de pista que serve como sensor exteroceptivo para o algoritmo do Capítulo 5.

A detecção de marcadores de pista para localização lateral e estimativa de curvatura é um dos primeiros campos de pesquisa em visão computacional aplicada ao problema de direção autônoma, desde o início dos anos 90. Muitas publicações e pesquisas foram feitas (MCCALL; TRIVEDI, 2006; HILLEL et al., 2014; YENIKAYA; YENIKAYA; DÜVEN, 2013; KAUR; KUMAR, 2015) descrevendo a diversidade das técnicas que foram desenvolvidas para resolver este problema. Alguns dos primeiros projetos (BROGGI; BERTOZZI; FASCIOLI, 1999) já demonstraram na prática a capacidade de tais sistemas em seguir autonomamente uma estrada por mais de centenas de quilômetros. Na verdade, se a estrada tem uma pista bem pintada ao longo de todo o seu comprimento, sem desvios ou bifurcações, até mesmo um sistema relativamente simples é capaz de manter o veículo seguindo pela estrada até que o tanque do veículo se esvazie. Recentemente, detectores de marcadores de pista voltaram a ganhar mais atenção devido à sua aplicação comercial em sistemas avançados de assistência ao condutor (*Advanced Driver Assistance Systems* - ADAS) (HUANG; FU, 2005; MIAO; LI; SHEN, 2012; SEO; HWANGBO, 2015; LU, 2015) em aplicações como alarme de saída da faixa ou mesmo sistemas que atuam no volante para trazer o veículo automaticamente de volta à pista.

Diferentes tipos de sensores podem ser explorados para detectar as marcações de estrada e pista. Como exemplo, *scanners* a LASER são amplamente utilizados para detectar a forma da estrada ou a sua infra-estrutura como *guard-rail*. Sensores a base de LASER são capazes de medir diretamente distâncias e requerem menos processamento comparado às técnicas baseadas em visão. Alguns autores (LEVINSON; MONTEMERLO; THRUN, 2007; LEVINSON; THRUN, 2010; DIETMAYER et al., 2005; HATA; WOLF, 2014) testaram o uso de um *scanner* LASER de múltiplos feixes para detectar os marcadores de pista, explorando a diferença de reflexividade entre as faixas pintadas e a superfície da estrada. Geralmente, as abordagens baseadas apenas em *scanners* a LASER para detectar marcações de pista são mais difíceis de trabalhar e, portanto, geralmente não são exploradas. Além disso, os *scanners* a LASER são caros e difíceis de serem instalados no exterior de um veículo, frequentemente em posições específicas, portanto, a necessidade

de alterações do exterior do veículo.

1.4.2.1 Detecção por Visão Computacional

As abordagens baseadas em visão são amplamente utilizadas para a detecção de marcações de pista e/ou estrada (GOPALAN et al., 2012; BORKAR; HAYES; SMITH, 2012; CUI; WANG; LI, 2014; NASHMAN; SCHNEIDERMAN, 1993; BROGGI et al., 1999; AUFRERE; CHAPUIS; CHAUSSE, 2000; CHOI et al., 2012). Câmeras são geralmente pequenas e fáceis de integrar em um veículo sem a necessidade de alterar sua aparência externa. Além disso, com exceção de câmeras especiais, construídas para adquirir faixas espectrais específicas, como infravermelho, as câmeras são equipamentos muito baratos. Outro ponto interessante é que os dados capturados por uma câmera contêm uma enorme quantidade de informações. Câmeras são, de fato, capazes de capturar a textura e a cor dos objetos observados e também podem ser utilizadas para estimar a superfície da estrada (BROGGI et al., 2013; ONIGA; NEDEVSKI, 2010). Todas essas características fazem das câmeras um dispositivo de percepção muito versátil que pode ser empregado em tarefas diferentes. Por outro lado, a extração desta informação requer um processo computacional complexo e pesado. Portanto, a escolha do sistema de computação tem de levar em conta as tarefas de processamento extra para visão computacional. A grande maioria das abordagens estudadas resolvem o problema do seguimento visual de caminho em três etapas: (i) Detecção dos pixels pertencentes à faixa; (ii) Associação dos pontos; (iii) Atualização do modelo de estrada (filtragem).

A detecção dos pixels pertencentes à faixa é a primeira etapa da extração de características e normalmente a mais pesada computacionalmente pois deve ser executada pixel a pixel. Para reduzir a quantidade de dados, pode-se usar regiões de interesse (ROI - *region of interest*), que delimitam uma sub-região da imagem onde o processamento é aplicado. Estas regiões correspondem à região frontal do veículo onde espera-se encontrar as características visuais esperadas. Por exemplo, em ((NASHMAN; SCHNEIDERMAN, 1993) e (POMERLEAU; JOCHEM, 1996)) uma ROI com forma trapezoidal foi usada enquanto em (AUFRERE; CHAPUIS; CHAUSSE, 2000) e (CHOI et al., 2012) duas ROIs foram usadas separadamente, uma para a faixa esquerda e outra para a faixa direita. As regiões de interesse ainda podem se ajustar de acordo com o estado do modelo da pista, normalmente centralizada em um tipo de média das últimas posições detectadas. Em (AUFRERE; CHAPUIS; CHAUSSE, 2000) e (CHOI et al., 2012) a largura da ROI varia de acordo com o nível de incerteza do modelo, tornando-se mais larga quando a incerteza aumenta.

Outra vantagem do uso de ROI é a redução de informação inútil, que pode ser considerada como ruído, e com isso aumentar a robustez do processo. A escolha da largura da ROI ou como ela deve variar em função da incerteza do modelo (no caso dos filtros

probabilísticos) é uma questão bastante empírica. Se a ROI for pequena demais pode ser que o *tracking* se perca quando houver uma mudança de direção das faixas (entrada e saída de curvas) e se for grande demais inclui muita informação visual desnecessária.

Um método bastante utilizado para detecção dos pixels, calcula inicialmente o módulo do gradiente da imagem e em seguida escolhe os pixels correspondentes aos máximos locais acima de um valor de limiar ((NASHMAN; SCHNEIDERMAN, 1993), (AUFRERE; CHAPUIS; CHAUSSE, 2000) e (LU et al., 2002)). Este tipo de abordagem tem o inconveniente de marcar as bordas das faixas e não o centro propriamente dito, além de gerar uma quantidade muito grande de pontos. Em (BROGGI et al., 1999) e (CHOI et al., 2012) ao invés do gradiente, um procedimento similar à convolução da imagem com uma máscara tipo pulso é aplicado, e com isso as regiões claras rodeadas por regiões escuras se destacam. A largura da máscara é escolhida para produzir o melhor resultado com a largura padrão das faixas, o que significa que diminui para linhas horizontais de imagem mais próximas do ponto de fuga.

Na segunda etapa é feita a associação dos pixels. Este procedimento tem o objetivo de agrupar os pixels que pertencem à mesma faixa e abordagens bem variadas foram propostas. Em (NASHMAN; SCHNEIDERMAN, 1993) a associação dos pontos é feita analisando a direção do gradiente e a distância entre os pontos marcados e os pontos correspondentes definidos pelo modelo. Em (BROGGI et al., 1999) os pontos são primeiramente conectados diretamente com seus 8 vizinhos criando *clusters*, que são então aproximados por *polylines* e extrapolados. Os pontos extrapolados que se aproximam de outros *clusters* estabelecem então uma nova conexão. Ao final, a rede de clusters conectados é analisada e a melhor representação é escolhida com base em certos critérios, entre eles a proximidade com o modelo. Em (CHOI et al., 2012) os pontos já estão pré-agrupados por terem sido encontrados em diferentes ROIs. O agrupamento final é obtido após a retirada dos *outliers* usando o algoritmo RANSAC (FISCHLER; BOLLES, 1981). Se após o agrupamento a quantidade de pontos for pequena demais, o agrupamento é considerado inválido e não é usado para a atualização do modelo.

Na terceira etapa é feita a atualização do modelo que pode ser entendida como um processo de filtragem. Diferentes modelos e métodos de atualização foram propostos. Em (NASHMAN; SCHNEIDERMAN, 1993) o modelo de pista proposto foram dois polinômios do segundo grau, um para cada faixa. Cada polinômio é atualizado fazendo-se um ajuste polinomial recursivo ponderado exponencialmente (*exponentially weighted recursive least squares*). Este algoritmo se assemelha a um ajuste polinomial simples pelo método dos mínimos quadrados, porém considera na função de erro a informação dos pontos obtidos nos quadros anteriores, com ponderação menor. Desse modo o estado do modelo leva em conta a informação do passado. Em (AUFRERE; CHAPUIS; CHAUSSE, 2000) foi proposto um modelo de pista formado por um conjunto de 10 pontos (coorde-

nada x) em diferentes linhas horizontais de imagem e a atualização do modelo é feita por filtro de Kalman (KALMAN, 1960). Em (CHOI et al., 2012) o modelo usado descrevia a pista como duas hipérbolas. Os agrupamentos de pontos obtidos eram transformados em duas hipérbolas temporárias, e seus coeficientes usados para atualizar o estado do modelo através de um filtro de Kalman.

1.4.3 Localização por Marcadores de Pista

Os projetos mais recentes, e que têm demonstrado os resultados mais impressionantes, operam com base em mapas e algoritmos de localização. Alguns projetos como (ZIEGLER et al., 2014; LEVINSON; MONTEMERLO; THRUN, 2007; LEVINSON; THRUN, 2010; BROGGI et al., 2014) demonstraram o potencial deste método com a realização bem sucedida de viagens autônomas em ambientes reais complexos. No caso de (ZIEGLER et al., 2014; BROGGI et al., 2014) foram usados apenas marcadores visuais, detectados por câmeras, para a construção de mapas e localização.

A localização baseia-se na detecção e estimativa de posição destes pontos de referência em relação ao veículo (PINK, 2008; BADINO; HUBER; KANADE, 2011; LATEGAHN; STILLER, 2012), que podem ser objetos como marcações de pista, faixa de parada, faixa de pedestre, ou objetos como sinais de trânsito ou mesmo postes.

As abordagens são as mais variadas, como a detecção da refletividade das marcações por meio de sensor LIDAR (HATA; WOLF, 2014), uso de câmera estéreo para calcular a posição da pista no mundo (PINK, 2008; LATEGAHN; STILLER, 2012), mapas digitais e imagens de coerência (MATTERN; SCHUBERT; WANIELIK, 2010), uso de SURF para detecção de *features* verticais, uso mapas de *landmarks* e redes neurais (LYRIO et al., 2015; DESOUZA; KAK, 2002), uso de esquemas combinados de visão frontal e traseira com diferentes tipos de *features* (ZIEGLER et al., 2014). Todas estas estratégias são apresentadas em detalhes na Seção 1.4.3.

Uma localização visual precisa baseada em visão requer a construção de mapas detalhados que também contém marcos de referência visuais. Na verdade, o processo de localização baseia-se na detecção e estimativa de posição destes pontos de referência (PINK, 2008; BADINO; HUBER; KANADE, 2011; LATEGAHN; STILLER, 2012). Preferencialmente, marcos de referência não devem estar muito longe do veículo uma vez que o erro na estimativa da sua posição aumenta com a distância. Além disso, marcos de referência devem ser fáceis de serem detectados, a fim de minimizar a ocorrência de falsos positivos/negativos nas detecções. Por exemplo, artefatos comuns existentes em uma estrada padrão, como marcações de pista, faixa de parada, faixa de pedestre, ou objetos como sinais de trânsito ou mesmo postes são todas opções adequadas. Em (PINK, 2008) um algoritmo monocular simples é utilizado para detectar as marcações de pista nas imagens. A posição de marcações de pista no mundo é então calculada usando informações

3D fornecidas por um sistema de câmera estéreo usando *V-Disparity*.

Em (HATA; WOLF, 2014) um sensor *scanner* LASER foi utilizado para medir a refletividade da superfícies ao redor para detectar as marcas rodoviárias (marcações de pista e faixa de pedestre). Esta informação foi usada para construir um mapa. Em seguida, o método de localização Monte Carlo (MCL) foi usado para localizar o veículo dentro do mapa resultante. Apesar da boa precisão relatada (0,31 m), este sistema tem o alto custo do sensor LIDAR como uma desvantagem.

Em (MATTERN; SCHUBERT; WANIELIK, 2010) mapas digitais e imagens de coerência foram utilizados para estimar a localização do veículo. Os mapas são construídos a partir de imagens aéreas, onde todas as informações sobre as marcações da pista e da superfície da estrada são extraídas manualmente. O mapa é usado para gerar imagens artificiais a partir de um determinado ponto de vista. Um filtro de partículas é utilizado para estimar a posição do veículo, com a probabilidade de cada partícula calculada a partir do valor de coerência entre a imagem atual e a imagem artificial associadas a esta partícula. Este sistema relatou uma precisão de 0,35 m, mas tem a desvantagem de necessitar de um mapa com construção manual.

Em (BADINO; HUBER; KANADE, 2011) é apresentado um esquema de mapeamento e localização baseado na detecção de *features* visuais por SURF (*Speeded Up Robust Feature*). A abordagem propõem o uso combinado de um mapa métrico e um mapa topológico para alcançar resultados métricos precisos, mas sem perder a robustez da localização topológica. O método relatado tem um erro médio de localização de 2,70 m, que é insuficiente para direção autônoma.

Em (LATEGAHN; STILLER, 2012) *features* 3D, extraídas por visão estéreo, são usadas para criar um mapa em 3D no qual é possível se localizar usando as *landmarks* atuais, calculadas em tempo real, e as que estão armazenadas no mapa. O valor reportado de erro médio (0,34 m) é suficiente para direção autônoma.

Já, em (LYRIO et al., 2015), um esquema de localização baseada em imagem foi apresentada com base em *Virtual Generalizing Random Access Memory (VG-RAM)* (DESOUZA; KAK, 2002). Um mapa neural é construído a partir de *landmarks* 3D, detectados por um sistema de visão estéreo, e usado para a localização. O erro de lateral médio relatado foi de 1,12 m, que não é suficientemente baixo para aplicação em direção autônoma.

Em (CUI; XUE; ZHENG, 2016) é apresentado um método de localização baseado na detecção de marcadores de pista que alcança precisão na ordem de centímetros. As marcações de pista são detectadas por uma câmera frontal que possui uma IMU interna para estimativa de postura. A abordagem usa um mapa construído manualmente, o *track* atual do veículo (estimado por fusão GNSS/INS) e um algoritmo de *map-matching* para antecipar a posição esperada das marcações de pista a frente e melhorar a confiabilidade

em sua detecção. As marcações detectadas são transladadas e um algoritmo iterativo closest point (ICP) (BESL; MCKAY, 1992) é usado para calcular o deslocamento do veículo, em relação ao mapa, com precisão centimétrica.

Em (ZIEGLER et al., 2014), uma combinação de câmeras frontal e traseira é usada para detectar diferentes tipos de pontos de referência. O sistema de visão frontal detecta elementos pintados no chão da estrada (marcações de pista, faixas de parada e outros) para obter a chamada LFL (*Lane Feature based Localization*). O sistema de visão traseira detecta *features* que são comparadas com os dados adquiridos anteriormente, para se implementar a localização denominada PFL (*Point Feature based Localization*). Esta dupla abordagem fornece uma localização mais precisa e confiável e permite que o sistema de localização se adapte bem a diferentes cenários, como estradas rurais, auto-estradas, ou ambientes urbanos. Os mapas são construídos *off-line*, usando imagens, odometria e dados de GNSS coletados durante uma viagem de mapeamento preliminar.

1.5 Objetivos e Contribuições deste Trabalho

O presente trabalho tem o objetivo de contribuir para o desenvolvimento da tecnologia de direção autônoma. O trabalho está focado especificamente no problema de localização e seguimento de caminho, ou seja, na capacidade de manter o veículo se movendo, de modo confiável e preciso, no centro da via de transporte. A questão dos obstáculos dinâmicos, igualmente importante, não foi tratada neste trabalho. Mais especificamente, o objetivo desse trabalho é projetar, construir e testar todo o *hardware e software* necessários à adaptação de um veículo comercial de modo que ele seja capaz de realizar com sucesso, repetidas vezes, uma missão de transporte pré-determinada.

A filosofia deste trabalho objetivou uma solução de baixo custo para o problema. Por isso não foram utilizados sensores caros, como sensores LASER, receptores GPS de alta precisão, ou unidades inerciais (IMU) comerciais. O algoritmo, baseado em visão computacional, foi desenvolvido com especial cuidado para ser computacionalmente o mais leve possível, mas sem comprometer o desempenho. O desenvolvimento do carro protótipo a partir de uma arquitetura de baixo custo também contribuiu para manter o custo do *hardware* baixo. Esta arquitetura inclui o aproveitamento de sinais originais do carro e a construção de circuitos eletrônicos customizados para esta aplicação.

Três abordagens amplamente utilizadas na literatura foram estudadas e testadas com o carro protótipo especialmente desenvolvido: o seguimento de caminho direto por detecção das marcações de pista, o seguimento de caminho por localização global (GNSS), e o seguimento de caminho por mapas de marcação de pista. Para cada uma destas abordagens foi dedicado um capítulo desta tese.

A ampla diversidade de problemas tratados inclui questões como a escolha do

sensor, construção do circuito de interface, adaptações eletro-mecânica no carro protótipo, especificação e montagem da câmera, levantamento do modelo cinemático do carro protótipo, síntese e programação dos algoritmos de visão computacional, síntese e programação dos algoritmos de controle, síntese e programação dos algoritmos de mapeamento de localização, planejamento e realização de testes para medição de precisão e, por fim, planejamento e realização de testes de direção autônoma. A tese está organizada conforme descrito a seguir.

No Capítulo 2 é apresentado o carro protótipo que foi desenvolvido para realização dos testes na estrada. Neste capítulo estão detalhados o projeto da unidade eletromecânica de acionamento do volante, as adaptações elétricas para aquisição de sinais do veículo, o circuito eletrônico da interface do veículo, o *firmware* desenvolvido para interface do veículo e o sistema de desarme de segurança.

No Capítulo 3 é apresentado o detector visual de marcações de pista e sua utilização em direção autônoma, como seguidor de pista. Neste capítulo é apresentado o algoritmo de visão computacional, a estratégia de controle do volante, e os resultados experimentais. O detector visual de marcações de pista foi aproveitado na abordagem do Capítulo 5.

No Capítulo 4 é apresentado o sistema de navegação por *dead reckoning* desenvolvido para o carro protótipo, e sua fusão com a informação do recetor GPS. A precisão dos sistema de localização é avaliada por testes experimentais e suas limitações discutidas. O sistema *dead reckoning* foi aproveitado na abordagem do Capítulo 5.

No Capítulo 5 é apresentada a parte principal desta tese de doutorado. Este capítulo foi resultado do amadurecimento das ideias e experiências apresentadas nos dois capítulos anteriores. Neste capítulo são apresentadas e discutidas as abordagens de Direção Autônoma Baseadas em Mapas e Localização, mais especificamente localização baseada em marcadores de pista. Primeiro é feita uma introdução, depois o algoritmo de localização é apresentado em detalhes, e por último os resultados experimentais que incluem testes de direção autônoma são apresentados.

A principal contribuição deste trabalho é o método de localização denominado BLMR (*Back Lane Marking Registry*). Este método é baseado em sensores de baixo custo mas que permite construir uma percepção precisa, confiável e extensa das marcações de pista na vizinhança do veículo. Esta extensa percepção combinada com o algoritmo rápido de filtragem e *map matching* levou a uma precisão de localização alta suficiente para operação em direção autônoma.

As contribuições secundárias surgiram do esforço em desenvolver um algoritmo leve de direção autônoma por seguimento visual de marcadores de pista (Capítulo 3). São eles: a conversão para imagem monocromática rápida com ênfase de cor; método para classificação dos pixels pertencentes à faixas (binarização) usando uma técnica de

convolução rápida com duas máscaras de largura ajustável; conexão entre fragmentos de faixas por meio de polinômios do segundo grau; divisão de uma faixa por mudança brusca de direção (detecção de "ponto de quebra"); uso de até quatro faixas, escolhidas por um critério de qualidade, para determinação da posição do veículo; e por último, filtragem com predição da posição futura das faixas a partir das informações de odometria.

As contribuições desse trabalho também foram apresentadas em dois artigos em congresso nacional (VIVACQUA; MARTINS; VASSALLO, 2012; VIVACQUA; MARTINS; VASSALLO, 2013) e em dois artigos publicados em revistas Qualis A1 (VIVACQUA et al., 2017; VIVACQUA; VASSALLO; MARTINS, 2017).

2 Construção do Veículo para Testes

2.1 Introdução

O carro protótipo foi projetado e construído para servir como laboratório para testar as ideias e algoritmos desenvolvidos no trabalho de doutorado em condições reais de operação. Foi feita a opção por adaptar um veículo comercial VW Gol 1.6 com câmbio manual, para este propósito. A adaptação do veículo foi feita de modo que os equipamentos para direção autônoma pudessem ser colocados e retirados com facilidade pois o veículo deveria servir a dois propósitos: veículo de uso particular e veículo protótipo. É importante frisar que os equipamentos eram instalados apenas quando o veículo servia como plataforma de testes, sendo retirados em seguida. Dessa forma, as adaptações não provocaram alteração nas condições originais de funcionamento do veículo quando este é usado como veículo regular. Visto por fora (Figura 5-a) não é possível perceber nenhuma modificação já que todos os sensores são internos. A Figura 5-b mostra o interior do veículo e a disposição de alguns dos equipamentos: (1) o computador central (um *notebook*), (2) o acionador do volante conjugado com a interface de operação, (3) a câmera e (4) o receptor de GPS.



(a) Exterior



(b) Interior

Figura 5 – Carro Protótipo.

A Figura 6 mostra a visão lateral esquematizada do veículo, detalhando a localização de todos os equipamentos instalados. O computador central (1), um *notebook* com processador Intel Celeron 1,86GHz, fica localizado no banco do carona, preso a uma almofada de apoio. Ele se conecta com os demais dispositivos por meio de três portas USB, uma para a câmera (4), uma para o receptor GPS (7) e uma para o módulo de interface com os sinais do veículo (2). Seu suprimento de energia é feito por um inversor de frequên-

cia de 127V/250W localizado no módulo de interface. O módulo de interface, responsável por concentrar e processar os sinais do veículo, fica localizado sob o acento do carona. Ele se conecta diretamente com o acionador do volante (3), o *encoder* das rodas dianteiras (5) e com o acelerador eletrônico (6).

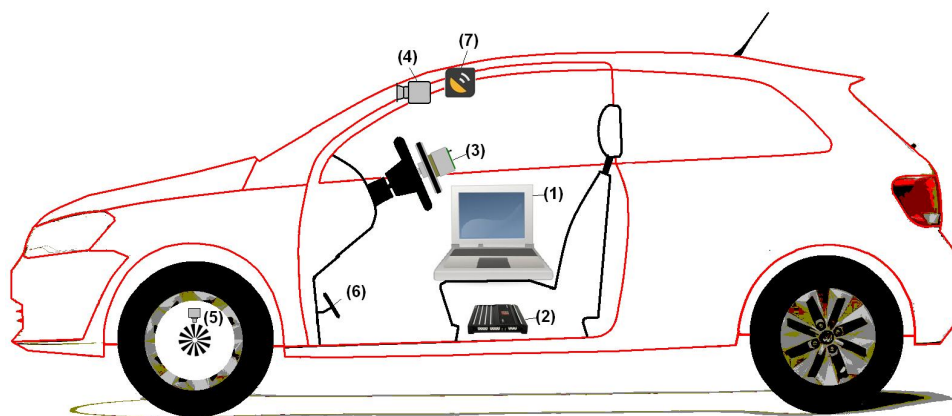


Figura 6 – Equipamentos instalados no carro protótipo.

Na secção seguinte será feita uma descrição detalhada de cada equipamento, onde serão apresentados detalhes construtivos, esquema de conexão elétrica, funcionalidades etc.

2.2 Módulo de Interface

O módulo de interface é o equipamento que concentra os sinais elétricos de controle do veículo e se conecta ao computador central por meio de uma porta USB e um protocolo próprio desenvolvido para esta aplicação. Este módulo roda o *firmware* de controle da interface do veículo, descrito na Seção 2.2.2, e implementa as seguintes funcionalidades: (i) Controlar o motor do acionador do volante; (ii) Controlar a interface homem-máquina (IHM); (iii) Receber e processar o sinal digital do *encoder* para calcular a velocidade interna; (iv) Gerar o sinal PWM para o acelerador eletrônico; (v) Receber e processar o sinal do sensor inercial.

O módulo de interface (Figura 7) fica localizado abaixo do assento do carona, fixado fortemente ao piso do veículo. Seus principais componentes são: (1) placa principal; (2) placa de acionamento de motor de passo; (3) sensor inercial em seu envólucro térmico.

A placa principal abriga o microcontrolador, o conversor USB para serial e circuitos adicionais de proteção e condicionamento dos sinais vindos do veículo. A placa do motor de passo contém os circuitos de potência para acionamento de motores bipolares de até 2,5A.

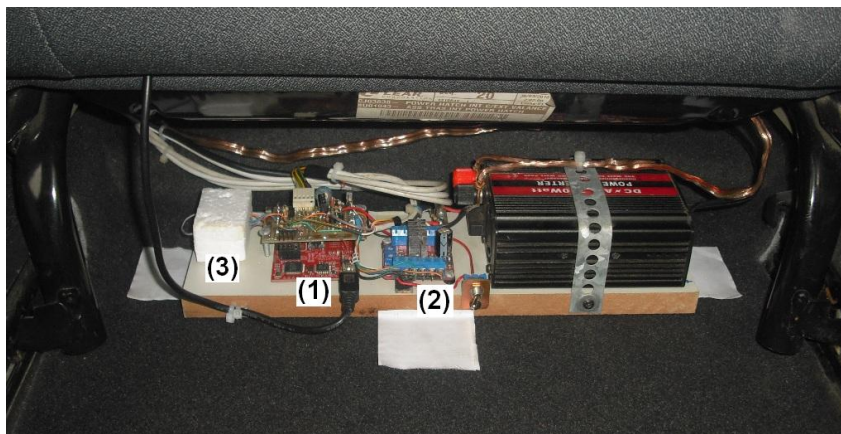


Figura 7 – Módulo de interface.

2.2.1 Hardware

A placa principal do módulo de interface, mostrada na Figura 8-a, foi construída a partir de uma placa de desenvolvimento "MSP-EXP430G2 LaunchPad Experimentier Board". O microcontrolador utilizado na versão atual é o MSP430G2553 com encapsulamento de 20 pinos. Este microcontrolador, de arquitetura RISC de 16-bits, possui 16kB de memória *flash*, 512 Bytes de memória RAM, dois *timers* com função de *capture and compare*, uma interface de comunicação universal (UART, *I²C* e SPI), 16 pinos de I/O digital, módulo comparador de tensão com 8 entradas, conversor AD com 8 entradas, cinco modos de operação de baixo consumo de energia. A Tabela 1 mostra a utilização dos recursos do microcontrolador na versão mais completa do módulo do carro.

Tabela 1 – Recursos utilizados do microcontrolador.

<i>Recurso</i>	<i>Função</i>
<i>clock</i>	interno 8MHz (50,0%)
memória <i>flash</i>	2,5 kBytes (15,6%)
memória RAM	146 Bytes (28,5%)
<i>timer</i> 0 (CCP0)	gerador de pulso de <i>clock</i> para o motor de passo
<i>timer</i> 0 (CCP1)	gerador de PWM para o acelerador
<i>timer</i> 0 (CCP2)	gerador de base de tempo <i>TickTime</i>
<i>timer</i> 1 (CCP0)	medidor de velocidade a partir da frequência do <i>encoder</i>
pinos de I/O	12 pinos utilizados (75,0%)
interrupções	TIMER0_CCP0, TIMER0_CCP2, TIMER1_CCP0, PORT2
USCI-A0	UART 9600bps para comunicação com o computador central
USCI-B0	<i>I²C</i> para comunicação com o sensor inercial
conversor AD	utilizado uma para controle direto via potenciômetro

Sobre a placa principal (Figura 8-a), foi montada a placa auxiliar (Figura 8-b) para abrigar os circuitos de condicionamento de sinal e os conectores externos. A placa auxiliar foi construída em uma placa universal e está eletricamente interligada à placa principal por dois conectores PCB de 10 pinos.

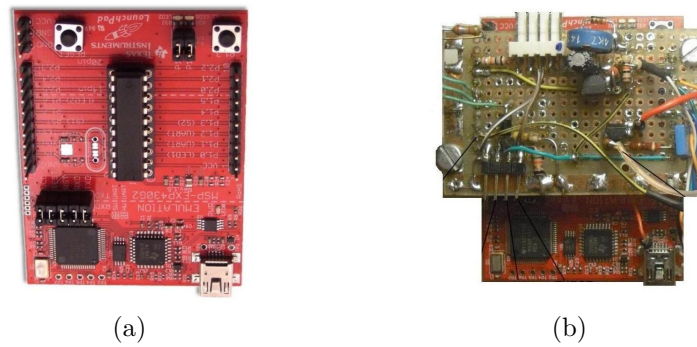


Figura 8 – Detalhe da placa principal e da placa auxiliar.

A Figura 9 mostra o diagrama esquemático resumido do módulo de interface. O conversor USB/serial está conectado ao microcontrolador por meio de dois sinais digitais TX - transmissão e RX - recepção. A placa de acionamento do motor de passo (Figura 7-2) se conecta ao microcontrolador por meio de três sinais de controle, *clock*, direção, e *reset*. Os sinais de *clock* e direção controlam a movimentação do motor e o sinal de *reset*, a energização. O sensor inercial (Figura 7-3) retira a alimentação de 3,5V da placa principal, e se comunica com o microcontrolador através de dois sinais digitais (SDA e SCL) de uma interface de comunicação I^2C .

A placa de acionamento do motor de passo (*Step Motor Driver*) opera em modo de corrente controlada PWM e usa dois resistores *shunt* para amostragem da corrente de cada bobina. Sobre um desses resistores é retirado o sinal I_{sense} correspondente ao valor instantâneo da corrente em uma das bobinas. Este sinal é filtrado pelo filtro passa baixa (R5, R6 e C1) e amplificado pelo transistor Q2 operando no modo base comum com o resistor de *pull-up* da entrada P2.4 atuando como resistência de carga. Este circuito possui um ganho alto suficiente para saturar com facilidade o sinal de saída e se comportar como um comparador de tensão. O sinal binário resultante ($I_{SensSat}$), é utilizado para detectar quando o volante é forçado pelo motorista, através da identificação de um padrão de pulsos rápidos que ocorre nesta condição. Sempre que esta condição ocorrer, o programa do módulo de interface desativa o motor de passo e isso libera o volante para direção manual.

O transistor Q1 é usado para aumentar a capacidade de fornecimento do pino de saída P1.0 que controla o acendimento do LED da IHM. O transistor Q3 controla em modo liga/desliga o resistor de aquecimento do envólucro térmico do sensor inercial para manter a temperatura interna constante. O resistor R7 em conjunto com os diodos internos da entrada TA1.0 funcionam como um circuito limitador de tensão que ajusta a amplitude dos pulsos gerados pelo *encoder* original do veículo (+12V) para o nível correto de operação das entradas do microcontrolador. Por último, o circuito formado pelo diodo D1, resistor R1 e o *trimpot*, geram a tensão de referência para o amplificador

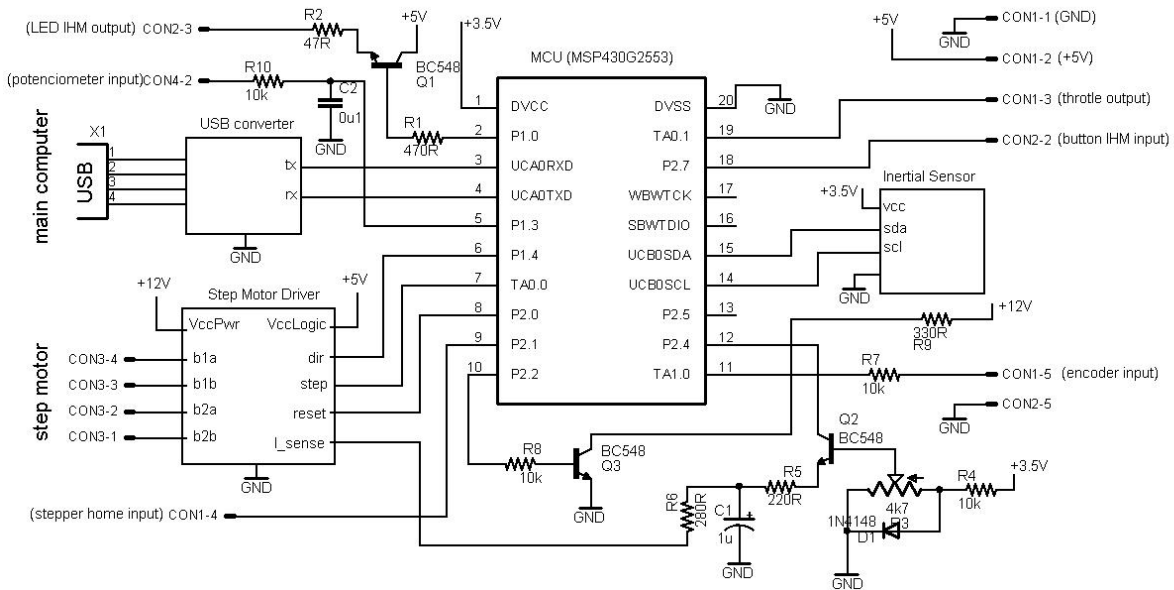


Figura 9 – Diagrama esquemático simplificado do módulo de interface.

do sinal de corrente. O ajuste deste trimpot influencia na sensibilidade da resposta do sistema de desarme do volante e deve ser feito experimentalmente até que a resposta desejada seja alcançada. O sinal *stepper home input* é gerado por um circuito de barreira óptica montado diretamente sobre o volante. Este circuito possui um LED infravermelho e um fototransistor que corta toda vez que o volante está na posição central (posição de referência neutra). Opcionalmente pode-se utilizar a entrada P1.3 para controlar diretamente a posição do volante por meio de uma tensão analógica de 0,3 a 3,5V, produzida por um potenciômetro externo.

2.2.2 Firmware

O *firmware* do módulo de interface foi desenvolvido em linguagem C, através da ferramenta de desenvolvimento e depuração integrada gratuita IAR Embedded Workbench IDE que permite a compilação de 4kB de código. Esta ferramenta é um ambiente de desenvolvimento integrado que permite depurar em modo simulado ou diretamente no microcontrolador alvo. Neste segundo caso é possível, por exemplo, acessar o nível lógico dos pinos de entrada a qualquer instante e com isso também depurar o *hardware*.

A arquitetura do *firmware* foi desenhada na forma de um módulo principal, que gerencia quatro diferentes processos: 1 - Amostragem de sinais e controle de temperatura, 2 - Comunicação UART, 3 - Odometria e cálculo de velocidade e 4 - Controle do motor de passo. A Figura 10 apresenta o fluxograma do módulo principal.

Cada um desses processos apresenta características diferentes em relação ao atendimento pelo processador. Por exemplo, a amostragem dos sinais pode ser feita em intervalos fixos de tempo enquanto o processo de geração dos pulsos para o motor de passo exige

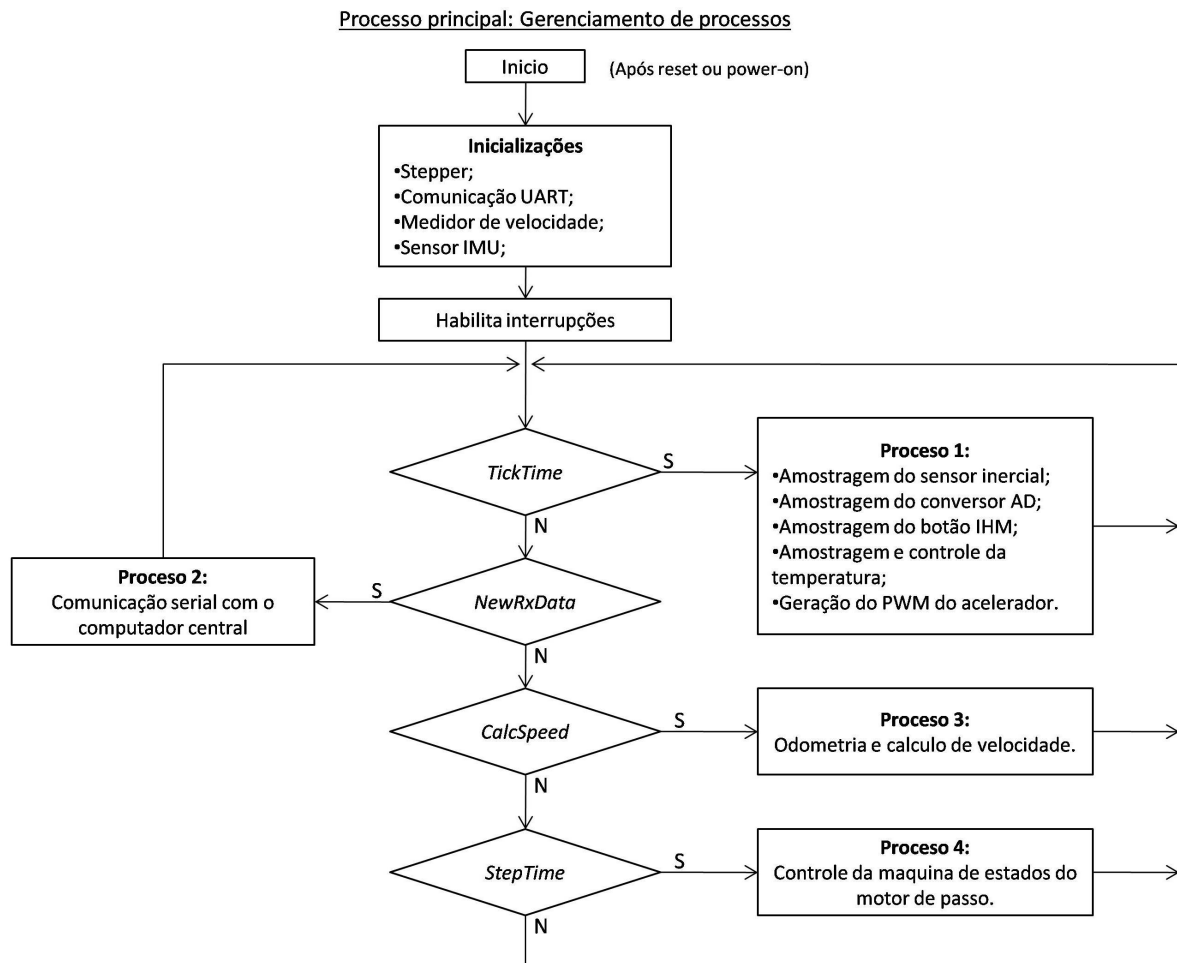


Figura 10 – Fluxograma do módulo principal de gerenciamento dos processos.

uma resposta imediata. A comunicação entre os processo é feita por variáveis globais e os *flags* *TickTime*, *NewRxData*, *CalcSpeed* acionados por interrupções. Na inicialização é feita a configuração dos recursos internos do microcontrolador tais como direção dos pinos, frequência de *clock*, *timers*, velocidade de comunicação, módulos de captura, módulos de comparação.

2.2.2.1 Processo 1: Amostragem de Sinais e PWM

O processo 1, apresentado na Figura 11, executa as ações que operam com frequência fixa mas que toleram um atraso no início da execução, como a amostragem de sinais, controle de temperatura e geração do sinal PWM de controle do acelerador. O *flag TickTime* controla a execução do processo 1 e é disparado a cada 8ms dentro da rotina de interrupção do módulo de captura e comparação (CCP2) do timer0.

A amostragem do sinal do sensor inercial se inicia quando o mestre da conexão *I²C*, neste caso o microcontrolador, envia para o escravo (o sensor inercial MPU6050) um comando de leitura de registrador de 16 bits. Este sensor possui um giroscópio de 3 eixos, um acelerômetro de 3 eixos, com tecnologia MEMS (*Micro Electro-Mechanical*

System). Sua operação é controlada por meio de um conjunto de 104 registradores, alguns de 8 bits e outros de 16 bits. A maioria desses registradores é usada para configurar o modo de operação do dispositivo, como por exemplo a sensibilidade do giroscópio, frequência dos filtros passa baixa internos e parâmetros de comunicação I^2C . Os endereços 0x3B a 0x47 foram reservados para se acessar os valores dos giroscópios, acelerômetros e temperatura. O sinal de velocidade angular do eixo Z do giroscópio, acessado no endereço 0x47, corresponde ao eixo perpendicular ao plano de movimento do veículo (*heading*). Após a recepção correta deste sinal, ele então é acumulado (integrado) em uma variável auxiliar que armazena a variação total de direção ocorrida desde a última transferência para o computador central, que ocorre em uma taxa menor, a cada 25ms. Toda vez que a variação total é transmitida seu valor é zerado para iniciar um novo ciclo de integração.

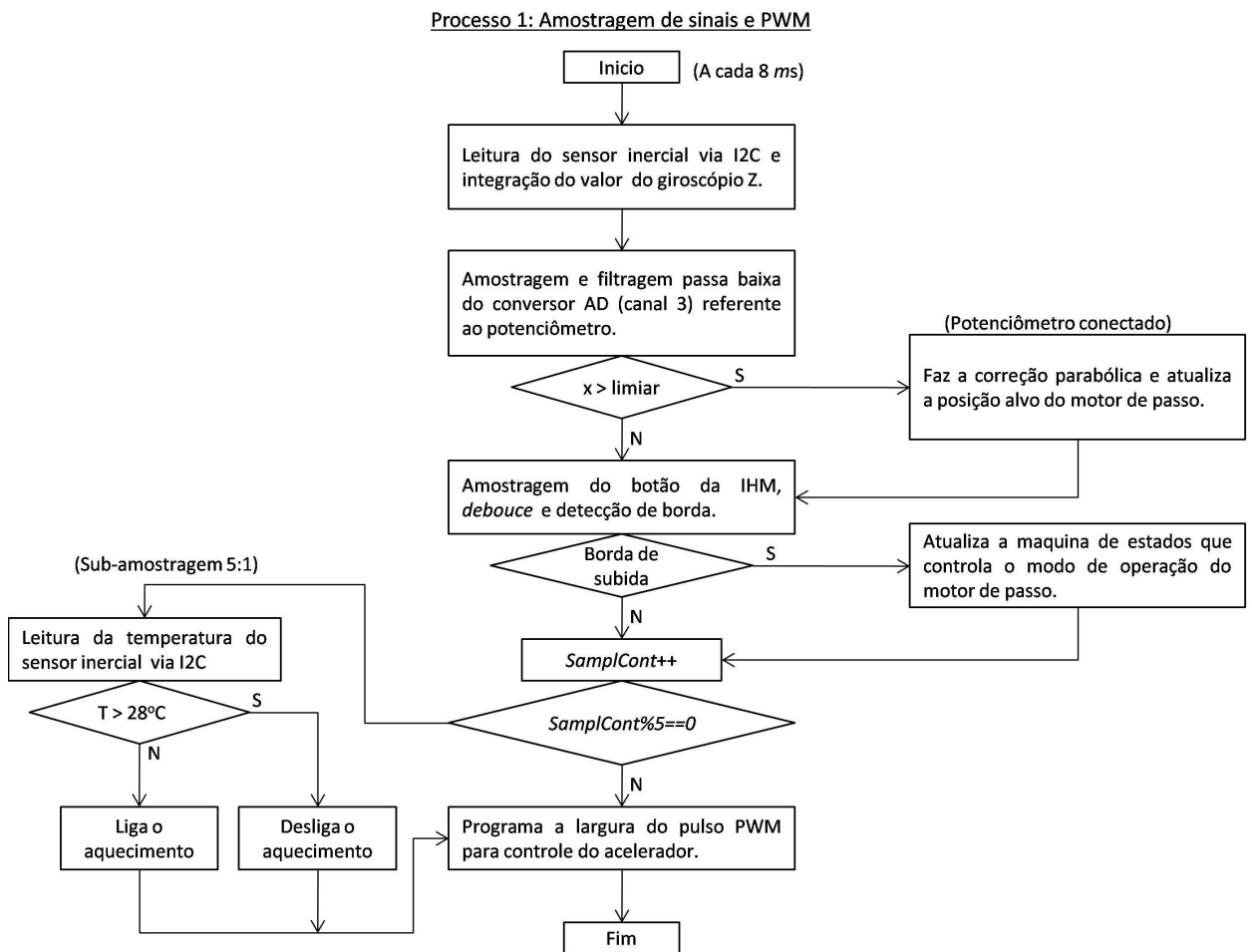


Figura 11 – Processo 1: Amostragem, controle de temperatura e geração de PWM .

O conversor AD realiza a amostragem da tensão do pino P1.3 e compara com um valor de limiar para verificar se há um potenciômetro conectado, e caso sim, qual é a posição do cursor. Um resistor de *pull down* interno garante que a tensão seja nula quando não houver um potenciômetro conectado. O sinal amostrado, u_i , passa por um filtro passa baixa IIR, descrito pela Equação 2.1, para aumentar a imunidade a ruído. O sinal filtrado x_i é aplicado a uma função não-linear descrita na Equação 2.2, que reduz a sensibilidade

da resposta na faixa próxima à posição neutra e com isso facilita o controle do veículo em velocidades maiores. O sinal resultante $x_{i,corr}$ atualiza o valor da variável $SetPos$, que é usada pelo processo 4 para posicionar o volante na posição desejada respeitando a dinâmica do sistema mecânico.

$$x_i = x_{i-1} + 0,2(u_i - x_{i-1}) \quad (2.1)$$

$$x_{i,corr} = x_i \left(1 + \frac{|x_i|}{2}\right) \quad (2.2)$$

O sinal amostrado do botão da IHM primeiramente é tratado por uma rotina *debouncing* para evitar múltiplos disparos devido à vibração dos contatos e em seguida aplicado a um detector de borda de subida. Quando uma borda de subida é detectada o estado de operação do motor de passo (ativar/desativar) é alterado.

O controle de temperatura do sensor inercial MPU6050 foi implementado com o propósito de compensar o *bias* do giroscópio. A não compensação do bias leva a uma estimativa de orientação bastante errada, conforme explicado mais adiante no Capítulo 4. A temperatura é mantida quase constante por um processo de controle tipo liga/desliga, em que a temperatura do sensor é lida via comunicação I^2C (registrador 0x41) e comparada com o valor de limiar, fixado em 28°C . Quando a temperatura estiver abaixo deste valor, o resistor de aquecimento, acionado pelo pino P2.2, é ativado. Caso contrário, o resistor é desligado. Fica evidente que para este sistema de controle funcionar corretamente a temperatura interna do veículo deve ser mantida abaixo de 28°C . A Figura 12-a mostra o funcionamento do controlador de temperatura na fase de aquecimento e após entrar em regime. Na Figura 12-b pode-se notar o pequeno desvio oscilatório da temperatura em torno da referência.

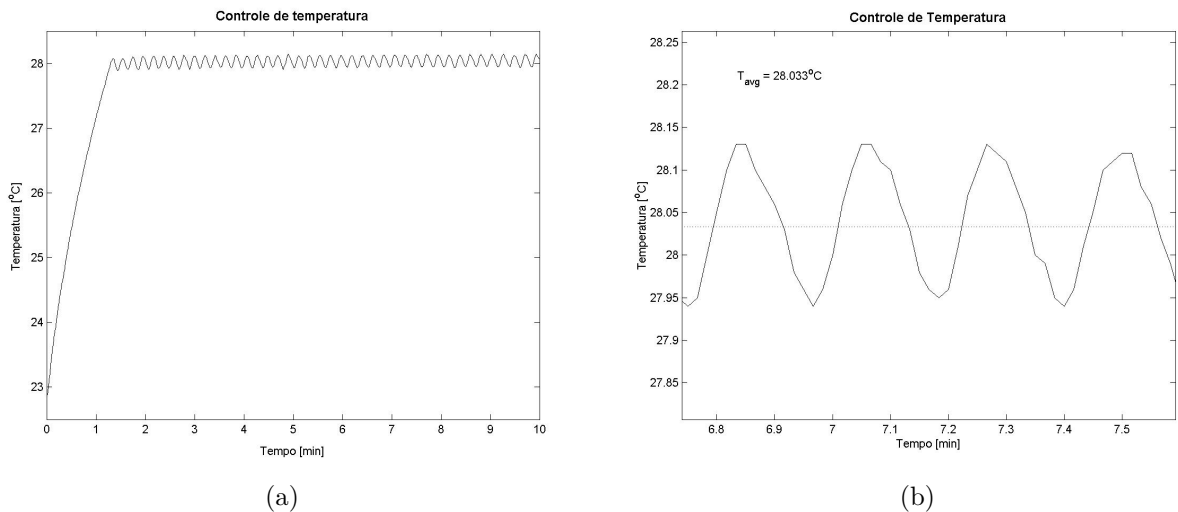


Figura 12 – Operação do controle de temperatura do sensor inercial.

A geração do pulso PWM usado para controlar o acelerador do veículo é feita com o auxílio do módulo CCP1 do timer0. Toda vez que o processo 1 é executado, o pino de saída TA0.1 é forçado para nível alto e o módulo CCP1 programado para forçar para nível baixo este mesmo pino automaticamente (sem a ação do processador) após decorrido o tempo definido pela variável global *ThrottleVal* que controla a razão cíclica do sinal.

2.2.2.2 Processo 2: Comunicação com o Computador Central

A transmissão e recepção dos dados entre o computador central e o módulo de interface é feita através de uma porta serial operando em 9600bps, 8 bits de dados, sem paridade e um *bit* de parada. Os dados seriais são convertidos pelo conversor serial/USB embarcado na placa principal. O protocolo de comunicação desenvolvido para esta aplicação é do tipo mestre-escravo, onde o mestre é o computador central e o escravo é o módulo de interface. A cada 40 ms o mestre transmite um *frame* (*MasterMsg*) ao escravo que responde imediatamente na forma de outro *frame* (*SlaveMsg*). Ao final de cada *frame* são adicionados dois *bytes* de controle, o *checksum* calculado sobre todos os *bytes* anteriores, e o terminador de frame (EOF) cujo valor em hexadecimal é 0x55. A Figura 13 mostra a construção dos *frames*. No caso de haver algum erro, a mensagem é simplesmente descartada, pois, conforme verificado experimentalmente, a frequência desse tipo de erro é muito baixa.



Figura 13 – Encapsulamento dos dados.

Os campos de dados do *frame* (*MasterMsg*) são: (Res.) - espaço reservado; (SetPos) - valor de 16 bits correspondente ao ângulo de posicionamento do volante em unidades de passos do motor; (Thro) - controle do acelerador. E os campos de dados do *frame* (*SlaveMsg*) são: (State) - estado de operação do controlador do motor de passo; (Res.) - espaço reservado; (Vel) - velocidade em km/h; (Odom) - Incrementos do *encoder* desde a última transmissão; (Res.) - espaço reservado; (GyroZ) - Variação angular de orientação do veículo desde a última transmissão; (Temper) - valor de 16 bits correspondente à temperatura interna do sensor inercial.

O processo de comunicação é iniciado toda vez que o flag *NewRxData* (Figura 10) é *setado*. Este evento ocorre após a recepção de um *frame* válido, disparado na rotina de interrupção de recepção de dados do módulo de comunicação serial do microcontrolador.

O *frame* recebido fica disponível em um *buffer* de recepção de dados (`unsigned char RxBuff[6]`). A Figura 14 mostra o fluxograma do processo 2.

Inicialmente é feita uma verificação para detectar se o potenciômetro de controle manual está conectado. Caso não esteja, então é feita a extração dos dados do *buffer* de recepção e os valores transferidos para as variáveis globais de controle *SetPos* e *Throttle*; Em seguida, é feita a montagem do *frame* de resposta, conforme a Figura 14, em um *buffer* de transmissão (`unsigned char TxBuff[11]`). Por último é disparado o processo de transmissão, que passa a ser controlado pela rotina de interrupção de transmissão do módulo de comunicação serial. Desse modo o processo não retém o controle do processador até o fim da transmissão o que gastaria aproximadamente 11ms.

Processo2: Comunicação com o Computador Central

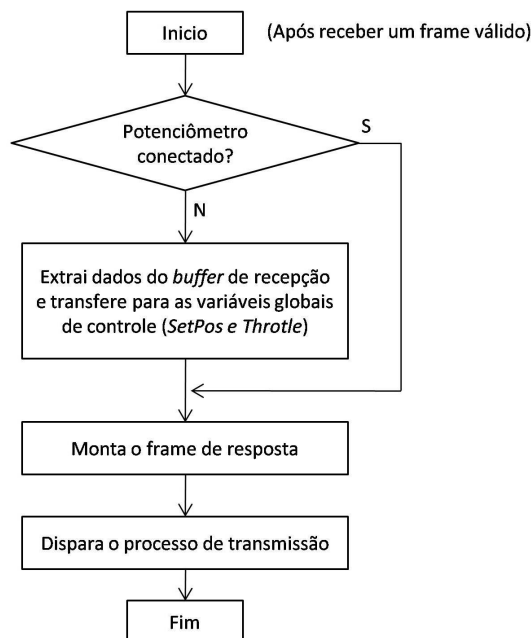


Figura 14 – Processo 2: Comunicação com o computador central.

2.2.2.3 Processo 3: Medição de Velocidade

A medição de velocidade do veículo é feita através da frequência dos pulsos gerados pelo *encoder* original do veículo. A resolução do *encoder* (pulsos por volta) e o raio da roda são tais que a velocidade em km/h é numericamente igual à frequência dos pulsos em Hz. O processo 3 implementa um frequencímetro, auxiliado pelo módulo de comparação e captura CCP0 do *timer1*. Este processo é iniciado toda vez que o *flag CalcSpeed* (Figura 10) é forçado para nível alto, o que ocorre após a recepção de quatro pulsos consecutivos no pino de entrada TA1.0.

O instante de recepção do quarto pulso fica retido no registrador de captura, variável t no fluxograma da Figura 15, e é usado para calcular o período médio dos pulsos. A variável auxiliar t_0 é usada para registrar o instante do pulso anterior e é atualizada a cada execução do processo. O período do pulso Δt então é comparado com o valor de limiar superior ($1/3s$) para detectar se o veículo está em velocidade muito baixa ou parado. No caso da velocidade ser considerada como zero, o processo termina.

Caso o veículo esteja operando em velocidade superior a $3km/h$, o período é comparado com um segundo valor de limiar, fixado em $1/120s$. Este segundo limiar serve para detectar uma possível falha no sistema de medição de velocidade, já que não será permitido ao veículo operar acima desta velocidade. Por último, se o período estiver entre os dois valores de limiar, a velocidade é calculada diretamente fazendo a inversão do período.

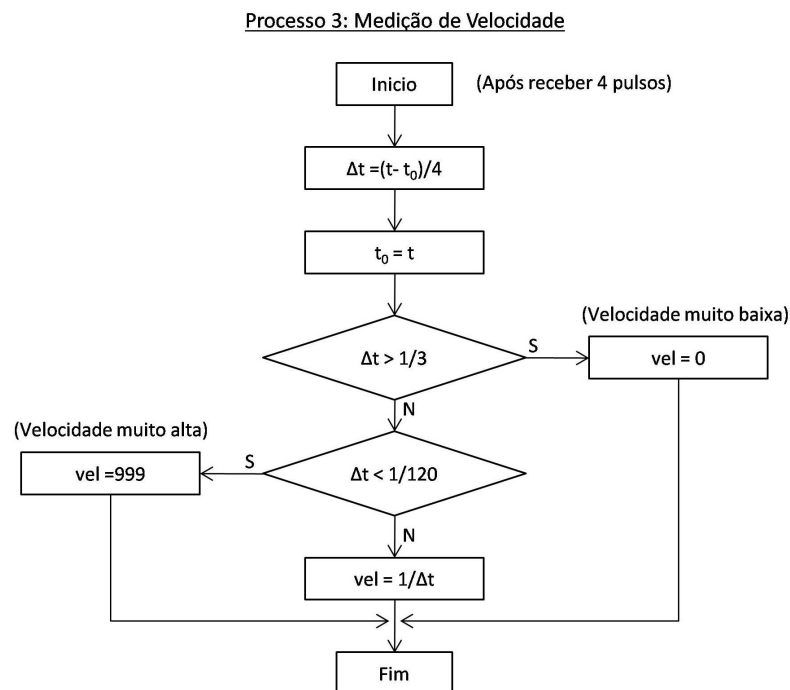


Figura 15 – Processo 3: Medição de velocidade.

2.2.2.4 Processo 4: Controle do Motor de Passo

O processo 4 controla a posição do motor de passo a partir da variável global *SetPos*, que define a posição alvo do motor em unidades de passos. Este controlador foi projetado para trabalhar com dois limites: de aceleração e de velocidade. O limite de aceleração foi estabelecido com base na capacidade de torque gerado pelo motor na velocidade máxima de operação, do torque de fricção devido à torção dos pneus e da inércia de todo o sistema mecânico. Este valor foi encontrado experimentalmente buscando o valor máximo suportado pelo torque do motor sem que houvesse perda de passos. O limite de velocidade foi estabelecido por questões de capacidade de torque do motor e segurança.

À medida que o veículo se desloca em velocidades maiores, a velocidade angular diminui conforme Figura 16. Isto evita que em velocidades altas um comando brusco de mudança de direção seja executado, o que seria uma situação potencialmente perigosa.

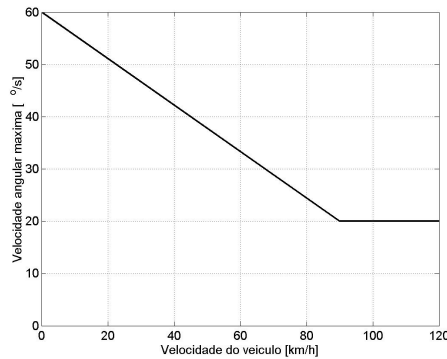


Figura 16 – Limitação da velocidade de giro do volante.

A Figura 17 ilustra o funcionamento do controlador do motor de passo. A variável *SetPos* corresponde à posição alvo desejada para o motor se posicionar e a variável *Pos*, à posição atual. A variável *SetVel* determina a velocidade alvo do controlador e a variável *Vel* a velocidade atual. Antes do instante t_1 , o motor se encontra na posição alvo e com velocidade zero. No instante t_1 a variável *SetPos* é escrita com um novo valor de posição. Como consequência, o controlador estabelece a velocidade alvo para o seu valor máximo. Durante o intervalo t_1 a t_2 o motor está ganhando velocidade em uma taxa constante até alcançar o valor máximo. No intervalo t_2 a t_3 a velocidade permanece constante. Ainda com um erro de posição, no instante t_3 , o controlador estabelece a velocidade alvo *SetVel* para zero de modo que quando a velocidade *Vel* chegar a zero o erro de posição também seja zero, o que ocorre no instante t_4 . Esta condição se mantém até que uma nova posição alvo seja definida.

Em certas condições, pode ocorrer que a diferença entre a posição alvo e a posição atual seja pequena o suficiente de modo que o motor inicia a desaceleração antes de atingir a velocidade máxima. Esta situação está ilustrada na Figura 17 no instante t_6 . O início da fase de desaceleração acontece quando o valor absoluto do erro de posição fica menor que a distância crítica de aproximação, calculada por $D_{aprox} = \frac{V_{el}^2}{2a}$, onde a é a aceleração.

A variável T_{pulse} é usada para controlar o intervalo entre os pulsos gerados para o motor de passo e consequentemente a velocidade de operação do motor. Quando o motor está parado, T_{pulse} está em seu valor máximo (teoricamente deveria ser infinito) correspondente à velocidade inicial da rampa de velocidade. Porém o motor não se move por que os pulsos no pino de saída TA0.0 são bloqueados. Quando inicia a movimentação, os pulsos são desbloqueados e a cada novo pulso, a variável T_{pulse} tem seu valor reduzido de modo a aumentar a velocidade de forma linear. A quantidade reduzida em T_{pulse} é uma função não linear, calculada por um algoritmo recursivo descrito em (PIŠI; KALOVÁ,

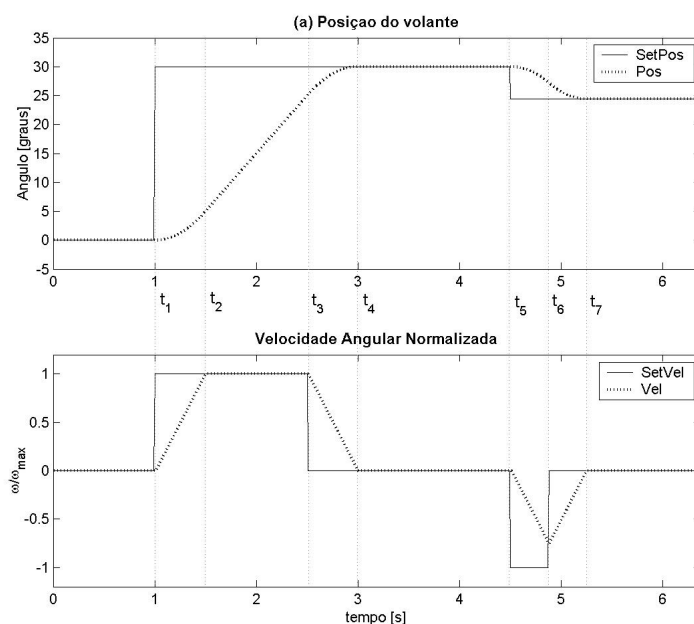


Figura 17 – Funcionamento do controlador do motor de passo.

2011). Independente do estado do motor (parado ou em movimento) o flag *StepTick*, que dispara o processo 4, é forçado para nível alto pela rotina de interrupção do módulo CCP0 do timer0 (modo de comparação) após decorrido o tempo definido por T_{pulse} .

O processo 4 também implementa uma máquina de estados para controlar o estado de operação do controlador do motor de passo, são eles: 1-Desativado, 2-Procurando Referência, 3-Ativado e 4-Intervenção. Ao iniciar o *firmware*, o motor se encontra desativado (motor desenergizado). Ao pressionar o botão de ativação na IHM, o controlador entra no modo 2, e se mantém até que o sensor de referência seja alcançado. Quando isso ocorre, o controlador passa para o estado 3, onde o motor é ativado (motor energizado). No estado 3, duas coisas podem ocorrer: o botão da IHM pode ser pressionado levando o controlador de volta para o estado desativado, ou o volante ser forçado pelo motorista (detectado pelo sensor de corrente), o que leva o controlador para o estado 4. Neste último estado, o motor também é desenergizado mas um *flag* de falha é marcado e transmitido ao computador central.

O fluxograma da Figura 18 detalha o processo 4 operando no estado ativado. Quando um novo valor de posição alvo *SetPos* é escrito, seu valor é comparado com a posição atual e o resultado dessa comparação define a velocidade alvo. Em seguida é feita a verificação da distância de aproximação, que caso seja ultrapassada, força a velocidade alvo para zero. Por último, a velocidade alvo é usada para controlar o processo de aceleração e desaceleração, através do aumento ou diminuição de T_{pulse} .

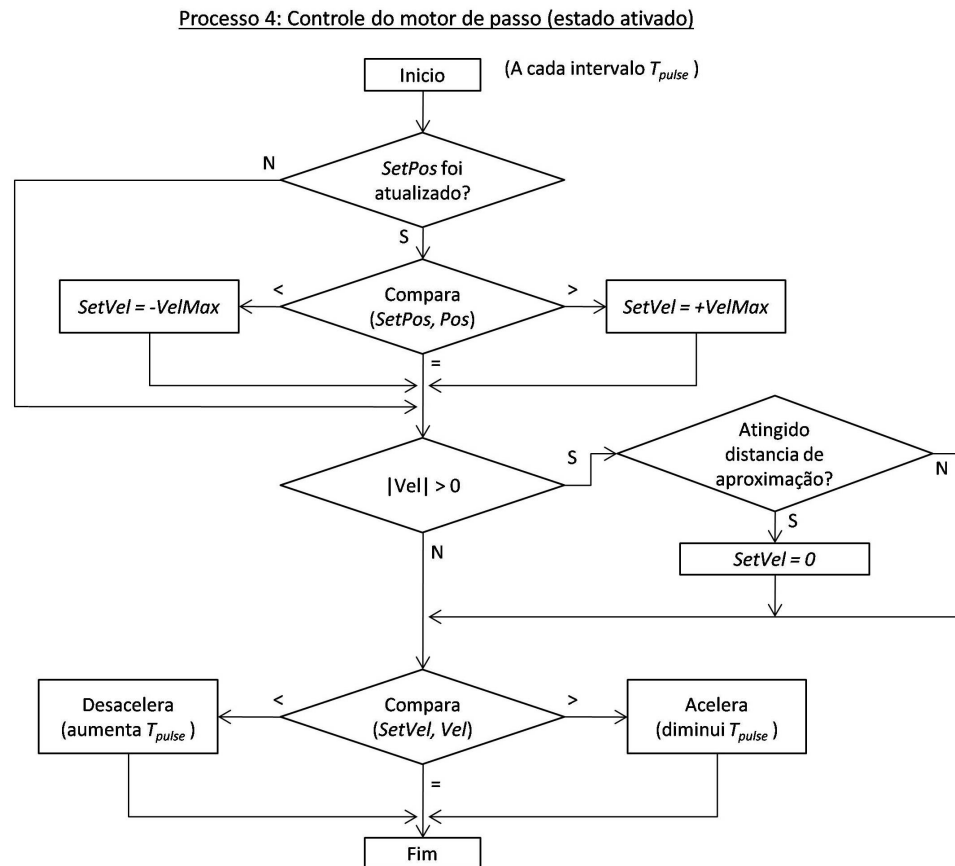


Figura 18 – Processo 4: Controle do motor de passo (estado ativado).

2.2.3 User API

A Interface de Programação de Aplicativos (API) do módulo de interface foi feita na forma de uma classe em linguagem C++, nomeada `Class_CarInterface`, apresentada na Figura 19.

```

class classe_CarInterface {
public:
    // variaveis
    int ComStatus;           // status de comunicacao
    int NewData;             // flag que sinaliza a chegada de um frame
    int Speed;               // velocidade em km/h
    float Temper;           // temperatura em graus
    float EncodIncM;         // deslocamento linear em metros
    float GyroIncRad;        // variacao da direcao em radianos
    float DriveVal;          // valor de acionamento do volante em radianos
    unsigned char Throttle; // valor do acelerador 0=0, 255 = 100%

    // funcoes
    void SetParamet(int _ComPort, int _BaudRate, int _WheelPosOffset);
    void Connect(void);      // abre a porta de comunicacao.
    void CalcThrottle(int SpeedSet); // calcula o acelerador para manter a veloc.
    void SendData(void);    // transmite dados
    int ReceiveData(void);  // recebe dados
};
  
```

Figura 19 – Classe `Car_Interface`.

A função `SetParamet(...)` define os parâmetros: número da porta de comunicação usada (porta COM virtual), a velocidade de comunicação serial, e o *offset* do

volante. O *offset* do volante é o valor de correção da referência de centro do volante, que por questões de erro de montagem, pode não corresponder exatamente à posição em que o veículo se move em linha reta. A função `Connect()` abre a porta de comunicação e estabelece a comunicação com o módulo de interface. No caso de sucesso, a variável `ComStatus` recebe o valor `CONNECTED` e, no caso de falha de comunicação ou não haver resposta válida, recebe o valor `NO_CONNECTED`. A função `CalcThrottle(...)` implementa um controlador PI que atua no valor de aceleração do motor (variável `Throttle`) de modo que a referência de velocidade seja mantida no valor (`SpeedSet`). Para o correto funcionamento do controlador PI, a função deve ser executada a cada 40 ms. A função `SendData()` encapsula os dados a serem transmitidos no *buffer* de transmissão e dispara o processo de transmissão. O intervalo de transmissão não deve ser menor que 20 ms e o valor testado foi de 40 ms. A função `ReceiveData()`, gerencia o *buffer* de recepção da porta de comunicação serial, faz a extração dos campos de dados e sinaliza pelo *flag* `NewData` quando há um *frame* válido disponível.

Para utilização da API, devem ser incluídos no projeto do aplicativo os arquivos `CarInterface.OBJ` e `CarInterface.H`. A Figura 20 ilustra um exemplo de um programa que utiliza a API.

```

void main() {
    classe_CarInterface CarInterface; // declara o objeto
    CarInterface.SetParamet(2, 9600, 0); // conecta via porta com2 a 9
    CarInterface.Connect(); // conecta com o dispositivo (abre porta com)
    /* faz as demais inicializações */

    // laço principal
    while(1) {
        // envia comando para o carro a cada 40ms
        if(++mSecCont==40) {
            mSecCont = 0;

            // controle do volante
            CarInterface.DriveVal = alpha; /* alpha = angulo do volante em radianos */;

            // controle do acelerador
            if(EnableSpeedControl) CarInterface.CalcThrottle(CruiseSpeed);
            else CarInterface.Throttle = 0; // modo neutro, não interfere no acelerado manual

            // transmite os dados
            CarInterface.SendData();
        }

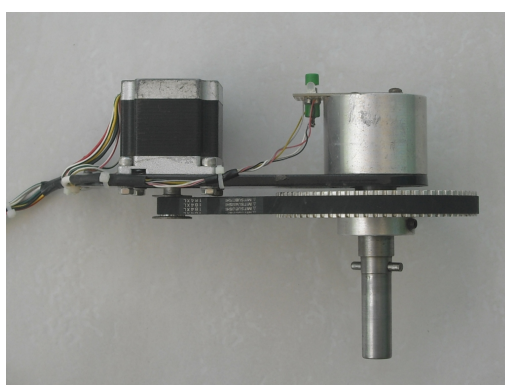
        // faz pooling para detectar a chegada da resposta com os dados
        CarInterface.ReceiveData();
        if(CarInterface.NewData) { // este flag vai para um
            CarInterface.NewData = 0;
            /* novos dados disponiveis */
        }
    } // fim do laço principal
}

```

Figura 20 – Exemplo do uso da API.

2.3 Acionador do volante

O acionador de volante é o dispositivo eletro-mecânico construído para movimentar o volante do veículo. Ele foi projetado para ser montado e removido com facilidade na parte frontal do volante, exigindo o mínimo de modificação mecânica no veículo. Conforme mostrado na Figura 21-a, o acionador de volante é composto por uma base, um motor de passo, uma transmissão por correia sincronizadora e sistema de acoplamento tipo macho e fêmea, e um pino de travamento. A única modificação necessária é a instalação da peça de encaixe fêmea (Figura 21-b), que substitui a porca de fixação original que prende o volante ao eixo de direção.



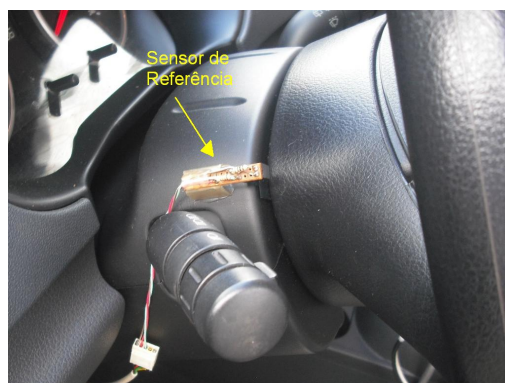
(a) Acionador do volante



(b) Adaptação no volante



(c) Sistema Montado



(d) Sensor de referência

Figura 21 – Acionador do volante: detalhes de montagem.

A Figura 21-c mostra o sistema montado na posição normal de operação, onde pode-se ver a barra de travamento que impede que o conjunto gire livremente junto com o volante. Na Figura 21-d, temos o detalhe do sensor de referência que indica quando o volante está na posição central.

A velocidade de giro máxima foi fixada em $60^\circ/s$. Este valor é alto suficiente para operação em rodovia, onde as curvas são suaves. Procuramos limitar o valor de velocidade de giro do volante como uma forma de aumentar a segurança nos testes de direção autônoma, pois em caso de um erro de comando gerado pelo programa de controle,

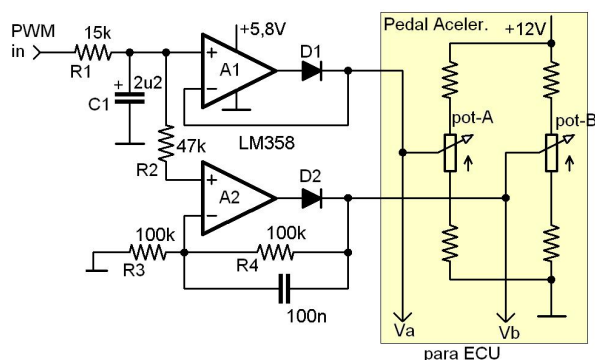
o carro não dá uma guinada brusca. É importante frisar que no caso de uma operação urbana, dentro de bairros, onde o veículo precisa fazer conversões de 90° para dobrar uma esquina, o valor escolhido não seria suficiente.

2.4 Acelerador Eletrônico

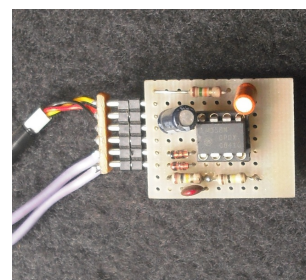
O veículo utilizado para ser o carro protótipo, o GOL 1.6, possui um sistema de acelerador eletrônico tipo *drive by wire*, que substitui o tradicional cabo de aço para conectar o pedal do acelerador à borboleta do TBI (*throttle body inject* - antigo carburador) por um sistema totalmente eletrônico. Este novo sistema usa um potenciômetro para medir a posição do pedal do acelerador, envia esta informação por fios elétricos para a central de controle do motor (*Engine Control Unit* - ECU) que por fim controla a abertura do TBI através de um motor elétrico.

Devido à dificuldade de se obter informações técnicas desse sistema, pois trata-se de informação interna das montadoras, alguns testes e medições foram feitos no acelerador do carro protótipo para descobrir detalhes de seu funcionamento. Estes testes revelaram que o potenciômetro do pedal gera duas tensões de controle. O sinal de principal, (V_a) que varia de 0,3V a 1,7V, e o sinal de controle redundante (V_b) que varia de 0,6V a 3,4V. O sinal redundante permite que no caso de falha do sinal principal, o veículo ainda continue a funcionar. Quando isso ocorre uma indicação de falha é sinalizada no painel do veículo.

O circuito que foi desenvolvido para controle do acelerador, apresentado na Figura 22, recebe os pulsos PWM do módulo de interface e gera as duas tensões de controle (V_a e V_b) para a ECU. O resistor R1 e o capacitor C1 formam um filtro passa baixa que extrai a tensão média do sinal PWM. A tensão filtrada então é entregue a dois amplificadores operacionais, um na configuração seguidor de tensão e outro na configuração não inversora com ganho 2. Os diodos D1 e D2 desacoplam a saída do circuito do potenciômetro quando a tensão de controle PWM for zero. Desse modo, quando o circuito não estiver sendo usado ele não interfere com o comando de aceleração normal proveniente do pedal.



(a) Diagrama esquemático



(b) Montagem

Figura 22 – Circuito de controle do acelerador.

A tensão de alimentação dos amplificadores operacionais é feita por um regulador de tensão LM317, com o valor exato de 5,8V. Este valor de tensão foi escolhido de modo que a máxima tensão produzida pelo circuito sobre os potenciômetros nunca ultrapasse o limite normal de operação ($V_{b,max} = 3,4V$).

2.5 Câmera

A câmera utilizada no veículo foi selecionada com base no seguinte critério: resolução VGA, baixo custo, conexão tipo USB, alta qualidade de imagem, taxa de quadros de até 30 *frames* por segundo. Após uma busca entre os principais fabricantes (Logitech, Microsoft, Genius, Creative), optou-se pela WebCam de alta resolução Logitech C270. Suas principais características estão apresentadas na Tabela 2.

Tabela 2 – Características da câmera utilizada.

<i>Parâmetro</i>	<i>Valor</i>
Tipo de conexão	USB 2.0
Taxa de quadros máxima	30 fps
Angulo de abertura horizontal	43°
Resolução máxima	1280 x 720 pixels
Resolução utilizada	640 x 480 pixels
Distância focal utilizada	814 pixels

Conforme será apresentado no próximo capítulo, o algoritmo de detecção de faixas usa uma transformação perspectiva inversa para converter a informação da imagem para a informação projetada no plano de rodagem do veículo (*bird's eye view*). Esta transformação usa certos parâmetros da câmera como altura do chão, distância focal, distorção geométrica, posição e orientação de montagem em relação à origem do sistema de referência adotado para o carro.

Não foi usado nenhum algoritmo para extração automática desses parâmetros, ou seja, a calibração da câmera. Ao invés disso, eles foram medidos diretamente. Quanto aos parâmetros próprios da câmera, foi desconsiderada a distorção geométrica e assumido o modelo *pinhole*. A distância focal foi medida diretamente com o auxílio de uma padrão de imagem quadrado colocado a uma distância conhecida à frente da câmera. A câmera foi montada sobre o espelho interno do veículo (Figura 23-a), a 1,30m de altura do chão, 2 cm à esquerda da linha central do veículo. A orientação foi fixada, com o auxílio de um horizonte artificial pintado em uma parede à frente do veículo na exata altura de montagem da câmera. Com o auxílio desta marcação zerou-se manualmente o ângulo *roll* e o ângulo *pitch* foi fixado em -10° . A Figura 23-b mostra um quadro capturado na posição de montagem da câmera.

Como a câmera foi fixada ao espelho retrovisor, portanto sujeita à mudança de posição, foram instalados dois marcadores no capô correspondentes a certas posições (pixels) da imagem da câmera. Estes marcadores estão indicados pelas setas na Figura 23-b. Quando estes marcadores aparecem na imagem nas posições corretas, então considera-se que a câmera voltou para a posição de calibração.

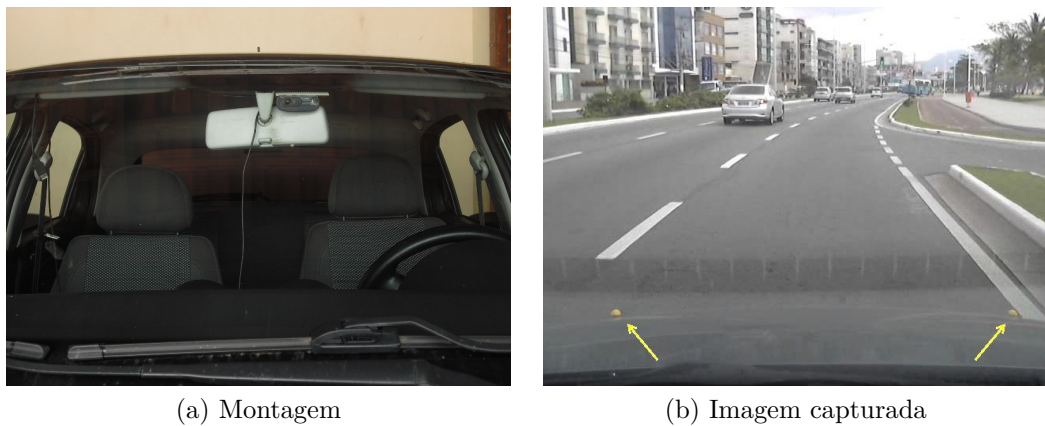


Figura 23 – Montagem da câmera.

2.6 Receptor GPS

O receptor de GPS que equipa o carro protótipo, é um receptor comum, de baixo custo, baseado na arquitetura SiRF III. Suas principais características são:

- Frequência de amostragem de 1Hz;
- Conexão tipo USB, com porta COM virtual em 4800 bps;
- Suporte ao protocolo de dados NMEA 0183 v3.0;
- Suporte para sistemas de melhoramento de precisão: DGPS, WAAS e EGNOS;
- Opera com até 20 satélites simultâneos.

O receptor, mostrado na Figura 24, possui pequenas dimensões (74 x 25 x 14 mm), e fica fixado diretamente no vidro dianteiro do veículo, na parte interna.



Figura 24 – Receptor GPS.

Entre as mensagens definidas pelo protocolo NMEA e reportadas a cada segundo pelo receptor, a mensagem GPRMC informa, entre outras coisas, a hora, a posição atual (latitude e longitude), a direção do deslocamento e da velocidade. Outra mensagem utilizada é a GPGSA, que informa os parâmetros de diluição de precisão vertical e horizontal (vdop, hdop). A diluição de precisão depende da quantidade e posição dos satélites detectados no céu, que por sua vez varia com o tempo e com a posição no globo. A diluição de precisão afeta a incerteza das medidas de posição global e isso pode ser aproveitado no caso do uso de filtragem probabilística. Na abordagem do Capítulo 4, as medições de latitude e longitude são usadas em um sistema de direção autônoma que foi testado no carro protótipo. Na abordagem do Capítulo 5, as mesmas informações são usadas para estimativa inicial da posição do veículo no mapa.

3 Direção Autônoma Baseada em seguimento de Marcadores de Pista

3.1 Introdução

A tarefa de conduzir um veículo em meio a uma estrutura pública de transporte de um endereço qualquer A para outro endereço qualquer B requer do sistema de controle do veículo um conjunto complexo de habilidades específicas, como: seguimento de caminho, ultrapassagem, mudança de faixa, conversão para via transversal, circulação de rotatórias, estacionamento, desvio de obstáculos (outros veículos, pedestres, animais); detecção de sinalização (placas, semáforos, quebra-molas) etc. Dentro desse complexo conjunto, o seguimento de caminho é a mais fundamental. Entende-se por seguimento de caminho a capacidade de seguir pelo centro da via de transporte, ou seja, o caminho garantidamente livre de obstáculos estáticos estabelecido para o veículo se mover com segurança. Por esta razão, este capítulo foi dedicado ao estudo desse tipo básico, porém fundamental, de direção autônoma. Assim como a maioria dos trabalhos relacionados, a abordagem desenvolvida para avaliação prática foi baseada na detecção visual dos marcadores de pista e utilizou o carro protótipo apresentado no Capítulo 2 para realização dos testes experimentais.

O programa de seguimento de caminho desenvolvido e detalhado neste capítulo aborda o problema em duas partes: o *tracking* visual das marcações de pista, que tem o objetivo de detectar e acompanhar a posição das marcações de pista na imagem da câmera, e um controlador da posição do volante, que tem o objetivo de calcular o ângulo de posicionamento do volante para que o carro se mantenha no centro da pista de forma automática. O algoritmo foi projetado para ser computacionalmente leve sem comprometer o desempenho. Algumas ideias foram aproveitadas de outros trabalhos já citados, outras foram aperfeiçoadas e algumas novas ideias foram adicionadas.

A arquitetura está apresentada na Figura 25. Ela se baseia em cinco módulos: os Módulos (1) e (2) formam o sistema visual de detecção e *tracking* das marcações de pista, o Módulo (3) faz o controle lateral controlando a posição do volante, o Módulo (4) é um controlador de velocidade que atua no acelerador para manter a velocidade constante, e o Módulo (5) é o responsável pela interface com os sinais elétricos do veículo.

Nas Seções seguintes são detalhados o funcionamento de cada módulo e o tipo de informação trocada entre eles.

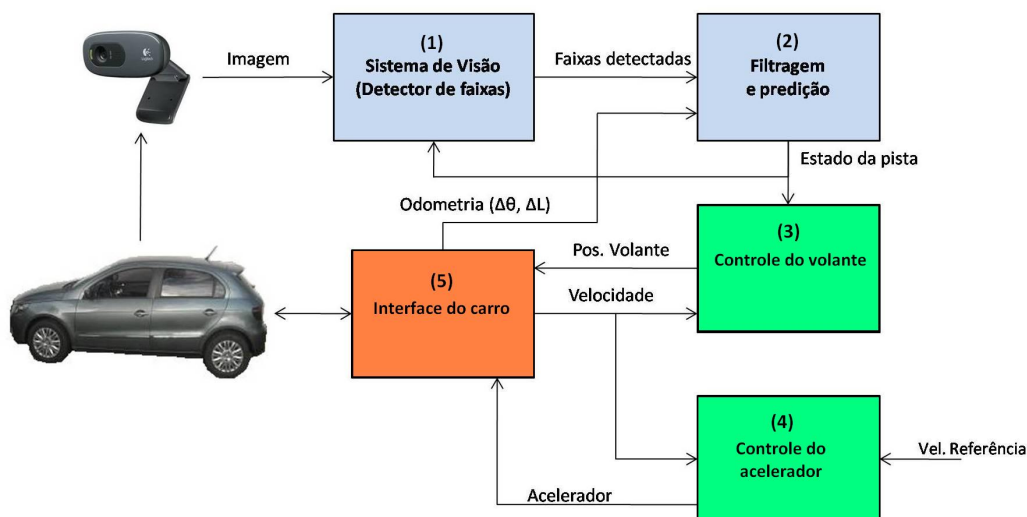


Figura 25 – Arquitetura do programa seguidor de faixas.

3.2 Sistema de Visão

O módulo do sistema de visão computacional recebe os quadros de imagem capturados pela câmera e detecta a posição de duas faixas, esquerda e direita. Cada uma das faixas é representada por um polinômio do segundo grau e recebe um atributo que representa o grau de confiabilidade desta informação. Por exemplo, no caso de uma faixa ter sido gerada por uma faixa longa, sem falhas ou imperfeições, isso reflete a condição mais favorável de operação, o que resulta em um grau de confiabilidade máximo. No caso de uma faixa muito curta ou com muitas falhas, a qualidade da medida será baixa e portanto a confiabilidade também. Caso não haja faixa, a qualidade será zero. Este módulo usa a informação do estado da pista, fornecida pelo módulo de filtragem, para delimitar a região de busca das faixas. O sistema de visão foi projetado para operar na taxa de 25 fps e o processamento ocorre em quatro etapas: conversão RGB-Mono com filtragem de cor; binarização da imagem; associação dos pontos; e construção da melhor representação. Cada uma destas etapas está detalhada nas Seções seguintes.

3.2.1 Conversão RGB-Mono com Filtragem de Cor

A transformação rápida de RGB para monocromático com filtragem de cor proposta neste trabalho implementa a filtragem de cor e conversão para monocromático em um só passo. Diferente de (AUFRERE; CHAPUIS; CHAUSSE, 2000), onde a filtragem foi feita com um filtro óptico colocado sobre a lente da câmera, neste caso a filtragem é feita por *software* junto com a transformação para monocromático em um só cálculo, que por sua simplicidade é mais rápido do que a própria conversão padrão RGB para monocromático.

O uso de algum tipo de filtragem de cor que realce o brilho das faixas melhora a

capacidade de detecção do sistema. A conversão proposta calcula o valor de intensidade de um pixel monocromático como sendo a média aritmética ($V = (R + G)/2$) das duas componentes de cor mais intensas da cor laranja (vermelha - R e verde - G), que é a cor usada nas faixas de sinalização no Brasil. A Figura 26 apresenta o resultado comparativo dos dois tipos de conversão. Com a conversão proposta, a razão entre as intensidades da região laranja (faixa) e do fundo (asfalto) aumentou em média 9%. Isso contribuiu para melhorar a capacidade de detecção das faixas de cor laranja sem alterar a sensibilidade às faixas brancas.

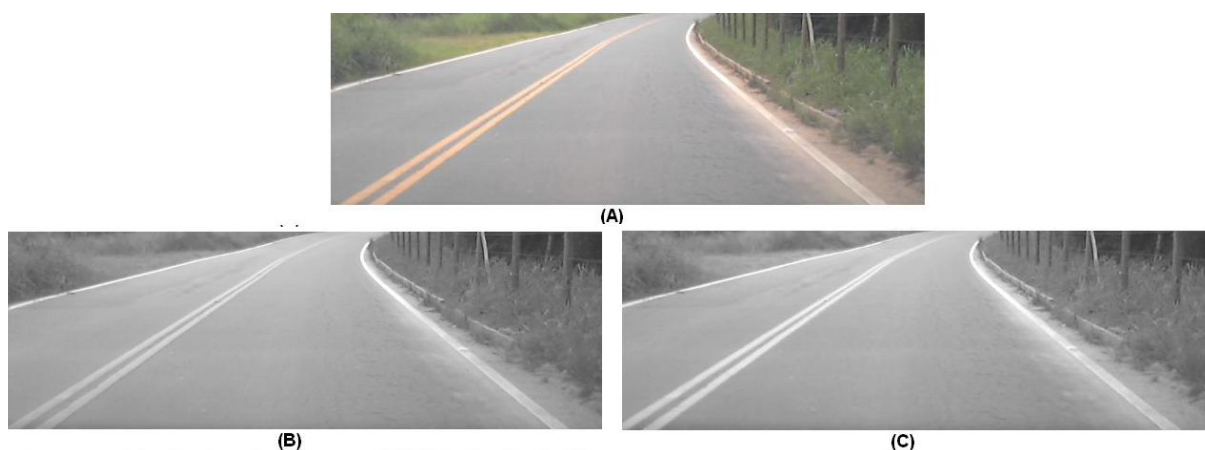


Figura 26 – Transformação para monocromático: (A) imagem original; (B) conversão padrão; (C) conversão proposta, com ênfase na cor laranja.

3.2.2 Binarização da Imagem

Após a conversão para monocromático é feita a detecção dos pixels que são mais prováveis de pertencerem a uma faixa, num procedimento denominado binarização da imagem. Este procedimento se baseia no fato de que uma faixa de sinalização é caracterizada por ser uma região clara cercada lateralmente por uma região mais escura (BROGGI et al., 1999) e (CHOI et al., 2012).

Em (BROGGI et al., 1999) esta condição é detectada de maneira simples e muito rápida em um procedimento que calcula a diferença de brilho entre um pixel central e seus dois vizinhos (esquerda e direita). Um pixel é marcado como pertencente a uma faixa se esta diferença for maior que um valor de limiar. Em (CHOI et al., 2012) este procedimento foi ainda melhorado. Ao invés de calcular a diferença de brilho entre dois pixels somente, calculou-se a diferença de brilho médio entre duas regiões de largura N, tornando o cálculo menos sensível a ruído. A diferença de brilho médio era calculada pela convolução do vetor de brilho de uma linha horizontal com duas máscaras tipo degrau defasadas. Esta ideia também foi usada aqui, porém com duas melhorias: cálculo rápido da convolução e uso de máscara de largura variável.

A Figura 27-a mostra a imagem monocromática com as duas máscaras usadas no algoritmo proposto neste trabalho de doutorado. As máscaras possuem largura variável $N(y)$ de modo a compensar o efeito da projeção perspectiva sobre a largura aparente das faixas. Considerando que a imagem obtida pela câmera tem uma resolução de 640×480 , para as linhas correspondentes aos pontos mais próximos (y_0) foi definido a largura de 22 pixels e para as linhas correspondentes aos pontos mais distantes (y_1) 2 pixels. Para valores intermediários foi feita a interpolação linear abaixo.

$$N(y) = 22 - 20 \frac{y - y_0}{y_1 - y_0} \quad (3.1)$$

As máscaras estão deslocadas entre si em meia largura ($N/2$). Na Figura 27-b está apresentado o perfil de brilho de uma linha, e abaixo o resultado da convolução com as duas máscaras. As setas #1 e #2 mostram os pontos que são marcados como pertencentes à faixa, ou seja, aqueles cujos os valores das duas convoluções estejam acima do valor de limiar.

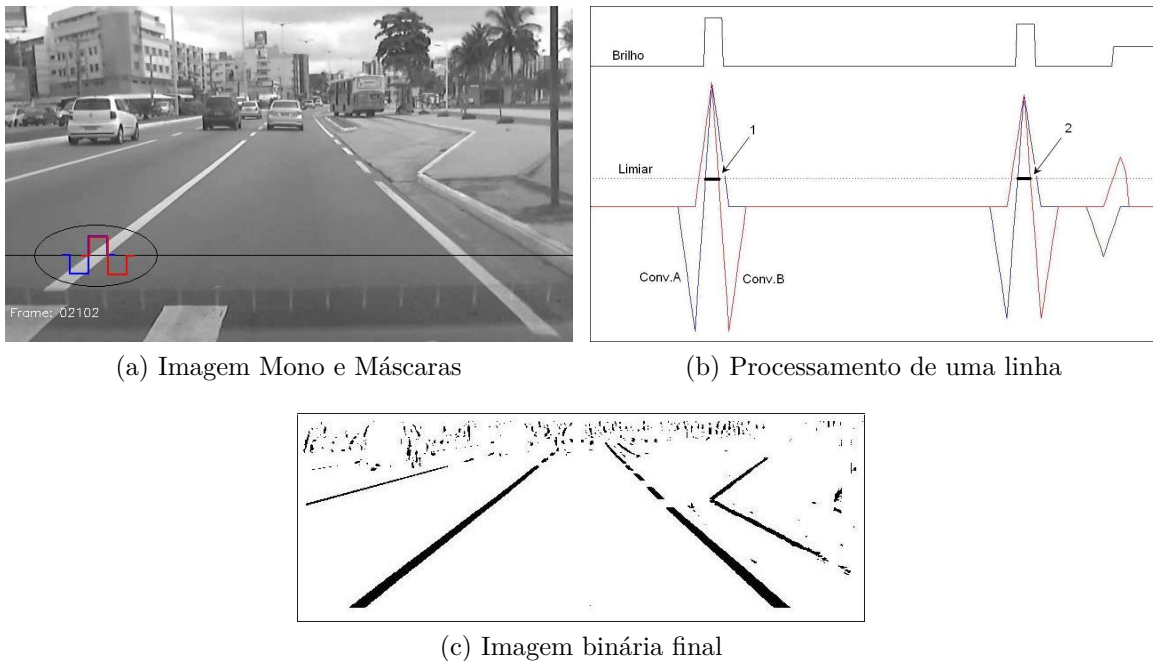


Figura 27 – Convolução e detecção dos pontos da faixa.

3.2.2.1 Algoritmo rápido de convolução

A convolução é um procedimento computacionalmente pesado pois exige uma grande quantidade de multiplicações e somas. Por exemplo, para uma máscara de largura 32 pixels, são necessários 32 multiplicações e 32 somas para cada pixel da linha processada. Em (CHOI et al., 2012) foi proposto o uso de imagens integrais (VIOLA; JONES, 2001), para reduzir a quantidade de cálculos necessários à convolução com as

máscaras tipo degrau. Ainda assim eram necessárias 8 operações de soma para o cálculo da convolução em cada ponto. Se levarmos em conta as 4 operações de soma necessárias para o cálculo da imagem integral de cada pixel, teremos um total de 12 operações.

O algoritmo proposto neste trabalho, calcula a convolução com apenas 4 operações de soma, o que torna a etapa de binarização muito rápida. O algoritmo se baseia no cálculo recursivo da i -ésima convolução, definida por

$$C_i = \sum M_i S, \quad (3.2)$$

onde: C_i é o valor da convolução avaliada para o i -ésimo pixel de uma linha; M_i é a máscara degrau centrada na i -ésima posição; e S é o vetor de brilho de uma linha. A Figura 28-a mostra graficamente o resultado da diferença entre duas máscaras deslocadas de uma posição (dM_i), e a Figura 28-b mostra como a máscara B (em vermelho, Figura 27-a) pode ser produzida diretamente a partir da máscara A (em azul, Figura 27-a).

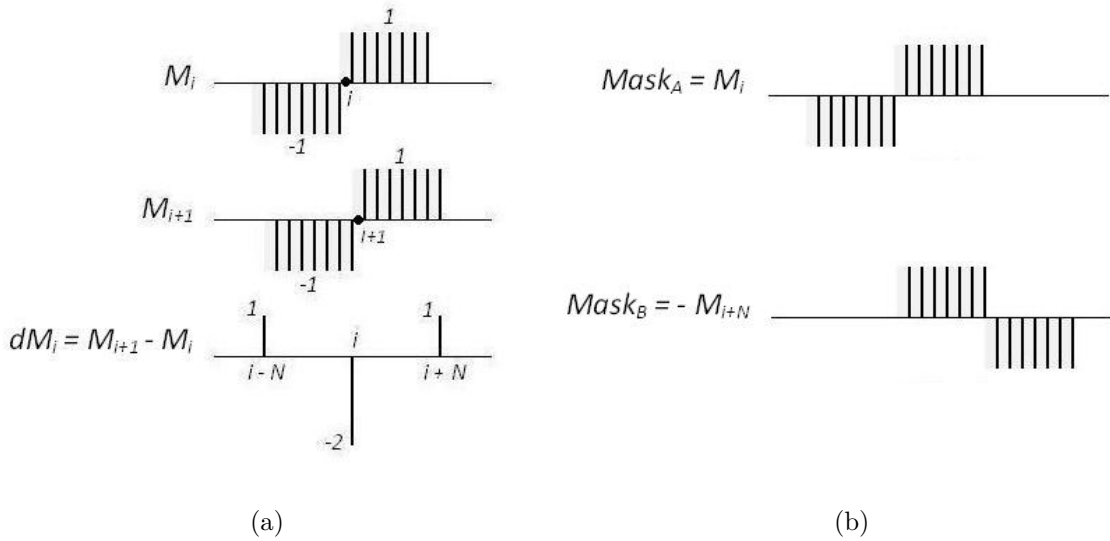


Figura 28 – Manipulação das máscaras para cálculo rápido de convolução.

O resultado dM_i é usado para calcular a convolução de forma recursiva conforme as equações

$$C_{i+1} - C_i = \sum (M_{i+1} - M_i) S = S_{i-N} - 2S_i + S_{i+N}, \quad (3.3)$$

$$C_{i+1} = C_i + (S_{i-N} - 2S_i + S_{i+N}), \quad (3.4)$$

onde S_i é o brilho do i -ésimo pixel de uma linha horizontal. O primeiro valor (C_0) é calculado normalmente pela expressão 3.2. O algoritmo teve um desempenho tão rápido (2ms) que não foi necessário o uso de ROIs, ficando esta possibilidade para melhoramentos futuros.

3.2.3 Associação dos Pontos

Esta etapa de processamento converte os aglomerados de pixels resultantes da binarização em polinômios do segundo grau que representam possíveis faixas ou fragmentos de faixas, no domínio do mundo (*bird's eye view*). Estes polinômios são agrupados dentro de duas regiões de validação (RdV), esquerda e direita, resultando em um polinômio final para cada lado. Cada um dos polinômios finais recebe uma nota de confiabilidade que varia de acordo com a qualidade e quantidade da informação visual usada na sua construção. O algoritmo proposto apresenta duas inovações, são elas: a etapa de conexão entre fragmentos adjacentes é feita com o auxílio de polinômios de segundo grau; e quando houver mudança brusca de direção (dx/dy) em uma linha gerada a partir de um aglomerado de pixels, a linha é dividida no ponto de mudança de direção pois entende-se que não se trata de uma condição normal.

3.2.3.1 Construção das cadeias conectadas por 8

Após o processamento dos pixels, o resultado obtido é uma imagem binária conforme mostrado na Figura 29-a, onde os pixels classificados como pertencentes à faixa aparecem em preto (pixels ativos). Estes pixels são agrupados com seus vizinhos para formarem cadeias conectadas por 8. O processamento é feito dentro de duas regiões de validação (RdV), uma para a esquerda e outra para direita, linha a linha, iniciando de baixo para cima. Cada cadeia conectada tem a sua linha média calculada (esqueleto) e a informação armazenada em um vetor de pontos $[y, x(y)]$, chamado de fragmento. O resultado são dois conjuntos de fragmentos, um associado à faixa esquerda e outro à faixa direita, conforme Figura 29-b.

As RdVs estão centralizadas de acordo com o estado atual do modelo que descreve a estrada e acompanha sua curvatura. O valor máximo de curvatura da RdV é estabelecido pelo limite visual da câmera (campo de visão) onde a informação visual de uma faixa começa a ser perdida. Considerando que ao aproximar-se de uma curva acentuada o veículo esteja em velocidade baixa, é aceitável que a informação visual considerada esteja à distância de até 15m (metade do alcance do sistema de visão). Considerando-se também uma largura de pista de 3,0m e os parâmetros da câmera utilizados no carro protótipo, isso resultaria em uma curvatura máxima 0,065 rad/m ($R = 15,8m$). A largura inicial da RdV (região próxima do veículo) é de 1/3 da largura total da imagem, reduzindo linearmente até a largura da região final (região distante do veículo) que é de 1/9 da largura total da imagem.

A Figura 27-a mostra uma condição onde parte do meio fio está dentro da RdV direita e, como seu brilho é mais claro que os arredores, ele é classificado como uma faixa. Na Seção 3.2.4, será apresentado o algoritmo de construção da melhor representação da faixa, aproveitando a informação de uma linha adicional paralela, como por exemplo o

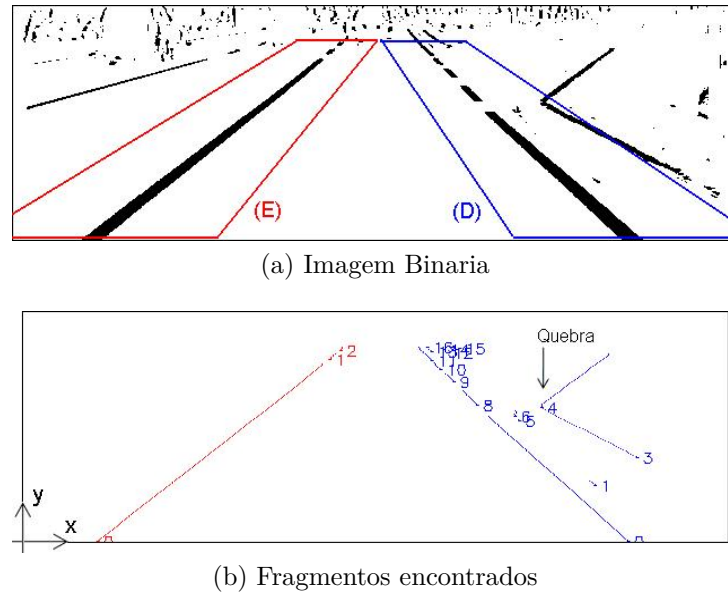


Figura 29 – Construção dos fragmentos de faixa.

meio fio, para melhorar a reconstrução da faixa. O problema é que o trecho de meio fio em questão corresponde a uma entrada de ponto de ônibus, caracterizada por uma "quebra" de direção acentuada. Se fosse usado um agrupamento (*clustering*) de pixels por conexão por 8, como em (BROGGI et al., 1999), o resultado seria um fragmento com características tão diferentes da esperada (segundo o modelo), que ele seria inteiramente descartado no processo de reconstrução da faixa.

Para evitar esse problema, foi proposto um filtro para detectar esta condição, que se baseia na análise do valor absoluto da derivada segunda ($\frac{\partial^2 x}{\partial y^2}$) para todos os pontos y_i de cada fragmento gerado, segundo o teste definido por: $|x_{i+6} - 2x_i + x_{i-6}| > 9$. O valor de limiar 9 foi encontrado com base nos testes com vídeo gravado, descrito na Seção 3.5.1.

Quando o resultado da comparação for verdadeiro, como indicado na Figura 29-b, um ponto de "quebra" é inserido, criando dois novos fragmentos independentes. Em (CHOI et al., 2012) este problema foi resolvido usando-se RANSAC para tirar os pontos *outliers* do cálculo. Esse método produz bons resultados, testando-se diversas hipérbolas ajustadas a partir um subconjunto de dados retirados aleatoriamente do conjunto principal (CHOI; KIM; YU, 1997). Em cada teste é contado a quantidade de *inliers*, por um critério de erro, e o ajuste que apresentar o maior número, é usado como solução. A necessidade de realizar vários ciclos de ajuste, e verificação do erro, torna esse algoritmo mais pesado do que o proposto aqui.

3.2.3.2 Projeção Perspectiva e transformação em polinômios

Cada fragmento tem seus pontos mapeados da referência da imagem da câmera para a referência do mundo $(x, y) \mapsto (X, Z)$, usando projeção perspectiva inversa (IPM),

conforme as Equações 3.6 e 3.5. A transformação IPM projeta os pontos da imagem no plano da estrada à frente do veículo, assumindo um modelo de estrada plana. Diferente da abordagem proposta em (BROGGI et al., 1999) onde a projeção IPM é feita sobre todos os pontos da imagem monocromática, neste caso a transformação é feita somente sobre um conjunto reduzido de pixels e, portanto, gasta-se menos tempo. A projeção é feita pelas expressões

$$Z = -H_{cam} \frac{\bar{y} \tan \alpha + 1}{\bar{y} - \tan \alpha}, \quad (3.5)$$

$$X = H_{cam} \bar{x} \frac{\sqrt{\tan^2 \alpha + 1}}{\bar{y} - \tan \alpha}, \quad (3.6)$$

onde: H_{cam} é a altura de montagem da câmera em relação ao plano da estrada; $\bar{y} = (y - y_0)/D_{foc}$ é o valor de y normalizado em relação à distância focal da câmera e centralizado no eixo óptico; $\bar{x} = (x - x_0)/D_{foc}$ é o valor de x normalizado em relação à distância focal da câmera e centralizado no eixo óptico; e α é o ângulo de inclinação do eixo focal da câmera para baixo em relação à linha do horizonte.

A transformação de cada fragmento gera um novo vetor de pontos, denominado fragmento mapeado, mostrado na Figura 30. Os fragmentos mapeados são então transformados em polinômios para acelerar o processo de conexão entre fragmentos, detalhado na seção seguinte. O algoritmo proposto nesse trabalho funciona da seguinte forma:

- Se houver apenas um ponto, ele é descartado;
- Se houver dois pontos, calcula-se a reta.
- Se houver três ou mais pontos, calcula-se um polinômio de segundo grau que contenha os dois pontos da extremidade e um ponto próximo do centro do intervalo.

A Figura 30 mostra o caso da transformação em polinômio de um fragmento com três ou mais pontos. Devido à característica não linear da transformação $y \mapsto Z$, os incrementos de Z são não lineares. Para acelerar o algoritmo, o ponto central escolhido é o ponto imediatamente acima do centro. Uma vez que se tenha os três pontos definidos (P_1 , P_2 e P_3), o cálculo dos coeficientes do polinômio é rápido e direto. Os coeficientes de cada polinômio e seu intervalo de validade (Z_{ini} e Z_{end}) também são armazenados na estrutura de dados criada para esse propósito.

A Figura 31 mostra o resultado final após o mapeamento e construção dos polinômios a partir dos fragmentos mostrados na Figura 29-b. Em vermelho estão os polinômios associados aos fragmentos capturados dentro da RdV esquerda e em azul os da RdV direita. A "quebra" do fragmento associado ao meio fio, discutido na seção anterior, resultou na construção dos polinômios 3 e 4 em azul.

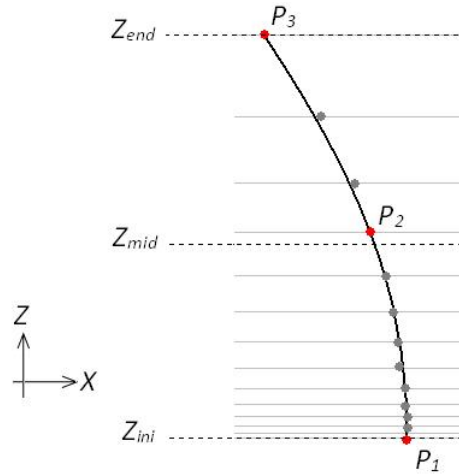


Figura 30 – Mapeamento de um fragmento e transformação para polinômio.

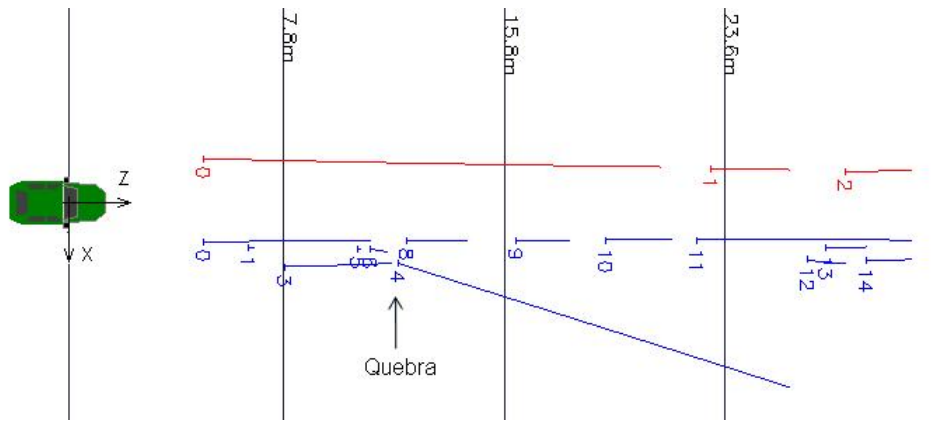


Figura 31 – Visão da perspectiva inversa dos fragmentos.

3.2.4 Construção da Melhor Representação

Em determinadas condições de operação, como por exemplo, com faixas intermitentes ou faixas com pedaços apagados, torna-se necessário conectar os pedaços separados que pertençam à mesma faixa para criar uma representação completa e contínua. Esta informação corresponde à medição instantânea da posição das faixas, que é fornecida para o módulo de filtragem para a atualização do modelo.

Em (BROGGI et al., 1999), (AUFRERE; CHAPUIS; CHAUSSE, 2000) e (CHOI et al., 2012) o estado do modelo da estrada é atualizado por duas faixas, sendo uma de cada lado (E - Esquerda e D - Direita). O algoritmo proposto neste trabalho pode utilizar até quatro faixas, sendo duas para cada lado (E e D). Por exemplo, no caso de haver faixa dupla no centro da pista e uma faixa simples à direita, a informação extraída dessas três faixas será usada para atualização do modelo. O uso da informação de mais faixas melhora a confiabilidade da medição instantânea. Além disso, o algoritmo proposto estima o grau de incerteza da informação que está sendo gerada e com isso permite, por exemplo, o uso de filtragem probabilística.

O processo de construção se inicia conectando-se os polinômios adjacentes que estejam alinhados, depois são selecionados os dois caminhos que têm a maior probabilidade de representar bem as faixas (maior qualidade) e por último a informação dos dois melhores são combinadas para construir o polinômio representativo.

3.2.4.1 Conexão entre polinômios

Dois polinômios são conectados quando há bons indícios de que fazem parte de uma mesma faixa. Para avaliar esta condição, foi proposto um conjunto de testes rápidos, que se baseia no alinhamento entre os polinômios. Dois polinômios são conectados quando todas as seguintes condições forem satisfeitas:

- Não haja sobreposição entre eles;
- O menor fragmento tenha um tamanho maior que um mínimo determinado;
- A distância entre os fragmentos (Δz) seja menor que soma de seus comprimentos;
- A diferença angular entre suas extremidades seja menor que 10° ;
- O teste de alinhamento esquematizado na Figura 32 seja atendido.

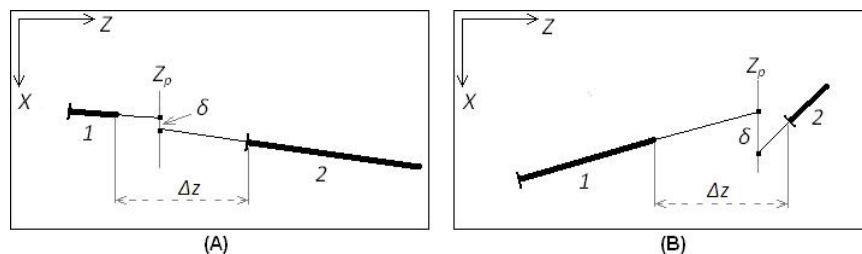
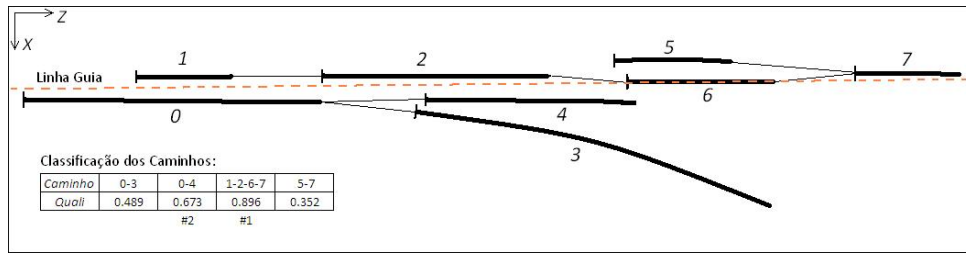


Figura 32 – Teste de alinhamento entre dois fragmentos: (a) Conexão estabelecida ($\delta \leq \delta_{min}$); (b) Conexão não estabelecida ($\delta > \delta_{min}$)

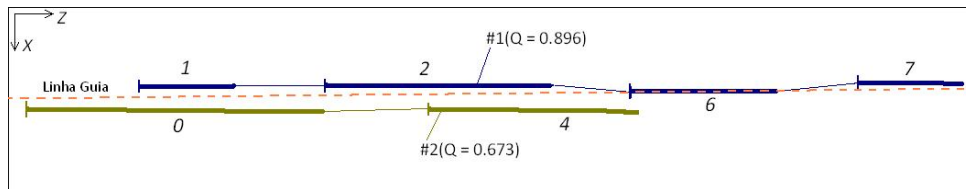
Cada par de polinômios é avaliado, através das expressões $X(Z)$, sobre a reta $Z = Z_p$ (Figura 32). O valor de Z_p é calculado de modo a ser proporcionalmente mais próximo do fragmento mais curto. Na Figura 32-a, por exemplo, o polinômio 1 é menor que o polinômio 2 portanto nota-se que a reta ($Z = Z_p$) está mais próxima do final do polinômio 1 do que do início do polinômio 2. Uma vez calculado o valor de X na interseção de cada polinômio com a reta ($Z = Z_p$), pode-se calcular a diferença ($\delta = X_1(Z_p) - X_2(Z_p)$). Se o módulo deste valor for menor que um determinado limiar, a conexão é estabelecida. A Figura 33-a mostra um exemplo de polinômios que já foram conectados. O valor de limiar, em metros, foi definido empiricamente como $0,15 + 0,05\Delta Z$.

3.2.4.2 Escolha dos melhores caminhos

Esta rede de polinômios conectados cria uma série de caminhos possíveis de serem percorridos, iniciando à esquerda e terminando à direita, em relação à figura. Cada caminho possível recebe um atributo de qualidade (0,0 = min; 1,0 = max), correspondente ao grau de confiabilidade da informação. Os caminhos possíveis com qualidade acima de 0,25 são ordenados pelo valor de qualidade, e os dois primeiros usados no cálculo da melhor representação da faixa (Figura 33-b).



(a) Caminhos possíveis



(b) Caminhos selecionados

Figura 33 – Exemplo de seleção dos dois melhores caminhos.

A qualidade de cada caminho é calculada pelo somatório das qualidades dos polinômios que o compõe. Por sua vez, cada polinômio tem sua qualidade calculada de forma proporcional ao seu comprimento, similaridade com a linha guia (linhas laterais do modelo da pista - Secção 3.3.1) e distância do seu início em relação ao veículo. A expressão heurística proposta em 3.7, fundamenta-se no fato de que a informação entre quadros consecutivos não pode variar muito e que as informações provenientes de pontos mais distantes possuem maior incerteza. As conexões entre os polinômios são consideradas como segmentos de retas. Por serem uma complementação artificial da informação visual, estes polinômios recebem uma penalidade extra de 20% no cálculo da qualidade. O cálculo da qualidade de um caminho é dado por

$$Q = \frac{1}{Z_{max}} \sum_{i=1}^n \frac{L_i}{(1 + |\frac{\Delta x_i}{\sigma_x}|)(1 + |\frac{\Delta \alpha_i}{\sigma_\alpha}|)(1 + |\frac{Z_{i,ini}}{0,5Z_{max}}|)}, \quad (3.7)$$

onde: n é o número de polinômios; L_i é o comprimento do i -ésimo polinômio; Δx_i representa o desvio médio horizontal; $\Delta \alpha_i$ é a diferença angular; σ_x é o parâmetro que define a sensibilidade ao desvio horizontal; σ_α é o parâmetro que define a sensibilidade ao desvio angular; $Z_{i,ini}$ é a distância do início do polinômio em relação ao carro; e Z_{max} é o alcance do sistema de visão (31,5m).

Os parâmetros σ_x e σ_α definem a intensidade com que um fragmento é penalizado ao se diferenciar da linha guia. Seus valores foram obtidos experimentalmente e os valores 0,50m e 20° produziram bons resultados.

Os caminhos (sequência de polinômios conectados) que foram selecionados por apresentarem os maiores valores de qualidade são primeiro transformados em polinômios auxiliares e depois combinados para formar os dois polinômios representativos finais, uma para esquerda e outro para direita. Cada polinômio representativo final ($\mathbf{u}_{rf,e}$ e $\mathbf{u}_{rf,d}$) é definido pelo seu vetor de coeficientes $\mathbf{u}_{rf} = (a_2, a_1, a_0)$, calculado a partir da expressão

$$\mathbf{u}_{rf} = \frac{Q_{aux1}\mathbf{u}_{aux1} + Q_{aux2}\mathbf{u}_{aux2}}{Q_{aux1} + Q_{aux2}}. \quad (3.8)$$

A Equação 3.8 corresponde ao cálculo da média ponderada pela qualidade. Desse modo, a representação final vai se aproximar mais das características do caminho de maior qualidade.

3.3 Filtragem

O módulo de filtragem recebe a informação visual instantânea, na forma de dois polinômios que representam a melhor reconstrução das faixas esquerda e direita (polinômio representativo final), e atualiza o estado do modelo de pista. Esta etapa de filtragem é fundamental para aumentar a imunidade a ruído do sistema. A filtragem proposta nesse trabalho é do tipo passa baixa de primeira ordem e considera no cálculo a qualidade (confiabilidade) das medidas. Medidas com baixa qualidade alteram pouco o estado do filtro (estado do modelo de pista) enquanto medidas com alta confiabilidade alteram mais. Opcionalmente pode-se usar ainda um processo de predição, baseado nas medidas de odometria (deslocamentos angular e linear) fornecidas pelo módulo de interface do carro. Este processo está descrito na Seção 3.3.5. O tempo de amostragem do algoritmo de filtragem é de 40ms.

3.3.1 Modelo da Pista

O modelo de pista corresponde à representação abstrata de como o veículo acredita que está em relação à estrada, na referência do mundo. Conforme apresentado na Figura 34, assim como em (BROGGI et al., 1999), (Lu, 2002), (CHOI et al., 2012), o modelo considera que as linhas laterais são paralelas e a estrada é plana. O modelo proposto é formado por uma linha central descrita por um polinômio do 2º grau ($x_c(z) = a_2z^2 + a_1z + a_0$), pela largura da pista e pela confiabilidade.

O estado do modelo é descrito por 5 variáveis: um vetor de coeficientes do polinômio central ($\mathbf{w} = (a_2, a_1, a_0)$), pela largura L_w , e pela qualidade Q_w .

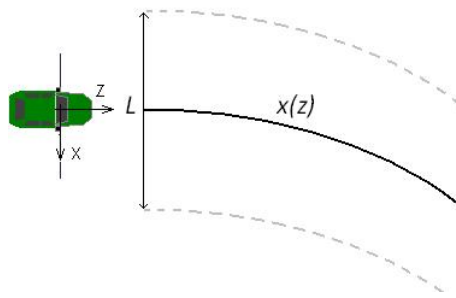


Figura 34 – Modelo da pista.

3.3.2 Atualização do Polinômio

Cada polinômio representativo é deslocado para o centro, transladando-se lateralmente em meia largura de pista ($\mathbf{u}_{rf,e}$ e $\mathbf{u}_{rf,d}$) e em seguida é feita a média ponderada pela qualidade, conforme a Equação 3.9, para gerar a medida instantânea da linha central.

$$\mathbf{u} = \frac{Q_{rf,e}\mathbf{u}_{rf,e}^* + Q_{rf,d}\mathbf{u}_{rf,d}^*}{Q_{rf,e} + Q_{rf,d}} \quad (3.9)$$

A qualidade desta medida Q_u é considerada como sendo a maior entre as duas pois de maneira geral uma medida de alta qualidade permite fazer uma melhor estimativa de curvatura do que duas medidas de qualidade mediana.

$$Q_u = \max(Q_{rf,e}, Q_{rf,d}) \quad (3.10)$$

A atualização da linha central do modelo proposto neste trabalho é feita a partir da medida instantânea da linha central (\mathbf{u}) por um filtro de primeira ordem com o ganho (k) variável, calculado pela proporção entre as qualidades (incertezas), de forma similar ao filtro de Kalman.

$$k_i = \frac{Q_{u,i}}{Q_{u,i} + 3Q_{w,i}} \quad (3.11)$$

A Equação 3.12 é usada para calcular o novo valor dos parâmetros da linha central a partir dos valores anteriores e da entrada (\mathbf{u}_i).

$$\mathbf{w}_{i+1} = \mathbf{w}_i + k_i(\mathbf{u}_i - \mathbf{w}_i) \quad (3.12)$$

É importante lembrar que a linha central estabelece o centro da região de validação (RdV) conforme discutido na Seção 3.2.3.1.

3.3.3 Atualização da Largura

A atualização da largura do modelo (L_w) é feita a partir da medida de largura instantânea (L_u), calculada pela distância entre os dois polinômios que melhor representam as faixas esquerda e direita, avaliado próximo ao veículo ($Z = 5,0\text{m}$). Neste caso, a atualização é feita segundo a equação

$$L_{w,i+1} = L_{w,i} + \frac{1}{16}(L_{u,i} - L_{w,i}). \quad (3.13)$$

No caso de não haver as duas faixas disponíveis, não é possível medir a largura e portanto ela é considerada como tendo o valor padrão de 3,5m. A constante de tempo da filtragem se torna maior neste caso, pois tipicamente trata-se de uma condição temporária de curta duração. Neste caso, a atualização é dada por

$$L_{w,i+1} = L_{w,i} + \frac{1}{32}(3,5 - L_{w,i}). \quad (3.14)$$

Os valores escolhidos são empíricos.

3.3.4 Atualização da Qualidade

A atualização da qualidade (confiabilidade) do estado do modelo é feita a partir da medida de qualidade das medições visuais instantâneas (Equação 3.10). Quando a qualidade da medida (Q_u) for maior que a do modelo (Q_w), foi definido que a qualidade do modelo deve aumentar rapidamente de modo a acompanhar esta variação. Este comportamento é desejável, pois assim que uma nova medida de alta qualidade é gerada o estado de incerteza de localização da pista deve cair imediatamente. O comportamento do filtro neste caso é definido por

$$Q_{w,i+1} = Q_{w,i} + \frac{1}{2}(Q_{u,i} - Q_{w,i}). \quad (3.15)$$

Nos instantes em que a qualidade das medidas instantâneas (Q_u) está abaixo a do modelo, definiu-se que o valor da qualidade do modelo deve cair a uma taxa constante com o avanço do veículo. Este comportamento reflete o aumento da incerteza do modelo na medida em que o veículo avança e não recebe novas informações visuais. O comportamento do filtro neste caso é definido por

$$Q_{w,i+1} = Q_{w,i} - 0,006. \quad (3.16)$$

É importante destacar que a qualidade do modelo é o indicador fundamental que permite avaliar o grau de incerteza na localização da estrada e portanto a condição de direção autônoma. Este tipo de informação pode ser usado, por exemplo, para determinar a

velocidade máxima permitida para o carro. No seguidor de pista desenvolvido neste trabalho de doutorado, a informação de qualidade do modelo foi usada para disparar um alarme sonoro sempre que o valor ficar abaixo de 40% para que o motorista de teste aumente sua atenção para eventualmente reassumir o controle do veículo a qualquer momento.

3.3.5 Predição da Posição da Linha Central

O uso da predição proposta neste trabalho visa reduzir o atraso introduzido pelo processo de filtragem, usando para isso a informação de deslocamento do veículo obtida através dos sensores de odometria. Esta ideia, publicada em (VIVACQUA; MARTINS; VASSALLO, 2013), ainda não havia sido aplicada em nenhum dos seguidores de pista estudados ((NASHMAN; SCHNEIDERMAN, 1993), (SCHNEIDERMAN; NASHMAN, 1994), (POMERLEAU; JOCHEM, 1996), (BROGGI et al., 1999), (AUFRERE; CHA-PUIS; CHAUSSE, 2000), (LU et al., 2002), (CHOI et al., 2012)).

A predição da posição da linha central calcula a posição esperada para a linha central no próximo quadro de imagem, a partir do conhecimento do modelo cinemático do carro, posição do volante e deslocamento linear. A Figura 35 ilustra esta ideia.

Na Figura 35-a temos o veículo se deslocando em curva à esquerda, onde no instante inicial (t_0) o veículo está sobre a origem do sistema de referência, neste caso o mundo (X_w, Z_w), e alinhado com eixo Z . No instante de tempo (t_1) o veículo aparece mais adiantado e sua posição relativa às placas diminuiu. Na Figura 35-b, observamos o mesmo deslocamento em outro sistema de referência (X_p, Z_p), onde a origem está presa ao veículo, denominado referência própria. Nesta referência, percebemos que o efeito do deslocamento do veículo foi a aproximação das placas, como se elas tivessem se movido em trajetos circulares, no sentido oposto.

A informação visual detectada pela câmera pode ser diretamente mapeada para o sistema de referência próprio, usando a transformação perspectiva inversa, conforme já foi discutido na Seção 3.2.3. Para calcular a posição de qualquer objeto, basta então conhecer a posição do centro de rotação (\mathbf{P}_r) e o deslocamento angular ($\Delta\theta$).

3.3.5.1 Centro de Rotação e Deslocamento Angular

A informação necessária para efetuar a rotação de um objeto qualquer (\mathbf{P}_r e $\Delta\theta$), posição do centro de rotação e o deslocamento angular, podem ser calculados a partir do modelo cinemático do veículo, mostrado na Figura 36, e das informações dos sensores de odometria, onde L_t é a distância entre os eixos; L_r é a distância entre a posição da câmera e o eixo traseiro; ξ é o ângulo de posicionamento das rodas; R o raio de curvatura; e \mathbf{P}_r o centro de rotação.

Na primeira versão do veículo protótipo, ainda não havia sido instalado um sensor

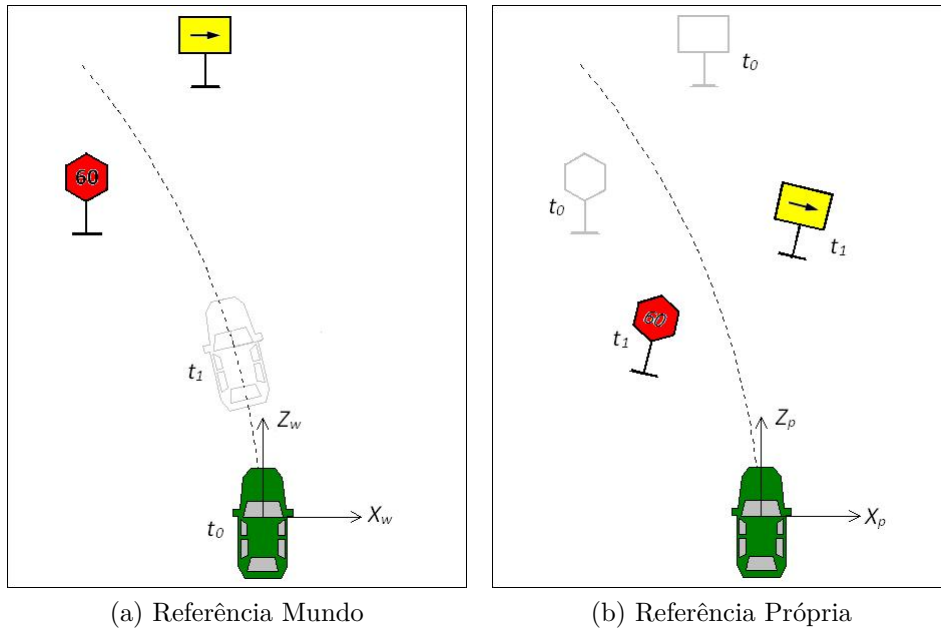


Figura 35 – Efeito do movimento sobre a posição dos objetos.

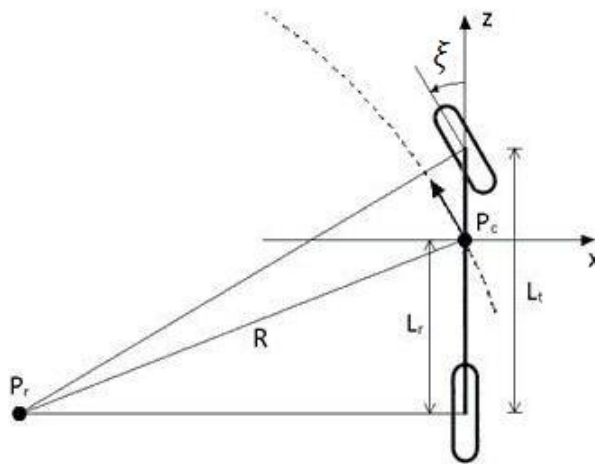


Figura 36 – Modelo Cinemático do carro.

inercial e o cálculo de predição era feito usando a informação de posição do volante e deslocamento linear. A posição do volante é conhecida através da contagem de passos do motor de acionamento do volante e permite calcular diretamente o ângulo ξ . O deslocamento linear é calculado pela contagem de pulsos do *encoder* original do veículo, usado pelo velocímetro. Neste caso os valores de $(\mathbf{P}_r$ e $\Delta\theta$) são calculados por

$$\mathbf{P}_r = (L_t \cot \xi; -L_r), \tag{3.17}$$

$$\Delta\theta = \frac{\Delta L}{\sqrt{(L_t \cot \xi)^2 + L_r^2}}. \tag{3.18}$$

Com a instalação do sensor inercial no veículo protótipo, a medida de deslocamento angular ($\Delta\theta$) já pode ser acessada diretamente e o centro de rotação calculado por

$$\mathbf{P}_r = \left(\sqrt{\left(\frac{\Delta L}{\Delta\theta}\right)^2 - L_r^2}; -L_r \right). \quad (3.19)$$

3.3.5.2 Rotação da Parábola

Conforme apresentado na Seção 3.3.1, a linha central do modelo de pista é uma parábola com eixo paralelo ao eixo X, descrita pela expressão

$$x(z) = a_2 z^2 + a_1 z + a_0. \quad (3.20)$$

Ao aplicarmos uma operação de rotação, obtemos uma parábola rotacionada, cujo eixo não está mais paralelo ao eixo X. Este tipo de curva pode ser descrito pela expressão geral

$$x_r^2 + c_1 z_r^2 + c_2 x_r + c_3 z_r + c_4 x_r z_r + c_5 = 0. \quad (3.21)$$

Esta forma é incompatível com a estrutura original de dados usada na classe que define o modelo de pista e não permite o cálculo explícito de x dado z , o que é computacionalmente indesejado. Para contornar esse problema, foi proposto um método para encontrar uma boa aproximação da parábola rotacionada mantendo a forma padrão dada por 3.20. Considere a equação da parábola rotacionada aproximada por

$$x_r(z) = a_{2r} z^2 + a_{1r} z + a_{0r}. \quad (3.22)$$

O problema a ser resolvido é encontrar os coeficientes a_{2r} , a_{1r} e a_{0r} . A ideia geral para resolver esse problema é: encontrar o ponto central \mathbf{M} (Figura 37) da parábola original $x(z)$ no intervalo $[z_0, z_1]$; aplicar a rotação angular $\Delta\theta$ em torno do centro de rotação \mathbf{P}_r fazendo $\mathbf{M} = (x_M, z_M)$ virar $\mathbf{M}_r = (x_{Mr}, z_{Mr})$; considerar que \mathbf{M}_r pertence à nova parábola $x_r(z)$; e considerar que a direção da reta tangente neste ponto foi subtraída pelo ângulo de rotação $\Delta\theta$.

A inclinação da reta tangente no ponto \mathbf{M} é dada por $x' = 2a_2 z_M + a_1$. Usando a propriedade da diferença de tangente, temos: $x'_r = \frac{x' - \tan\Delta\theta}{1 + x' \tan\Delta\theta}$. Como a curvatura da parábola rotacionada não é alterada, temos $a_{2r} = a_2$. O termo a_{1r} pode ser calculado por $a_{1r} = x'_r - 2a_{2r} z_{Mr}$, e o termo a_{0r} é calculado por $a_{0r} = x_r - a_{2r} z_{Mr}^2 - a_{1r} z_{Mr}$.

A Figura 37 mostra a operação de rotação. A curva contínua representa a parábola original $x(z)$ com a ordenada do seu ponto médio calculada por $z_M = \frac{z_0 + z_1}{2}$. Após a rotação, temos a parábola rotacionada descrita pela curva pontilhada.

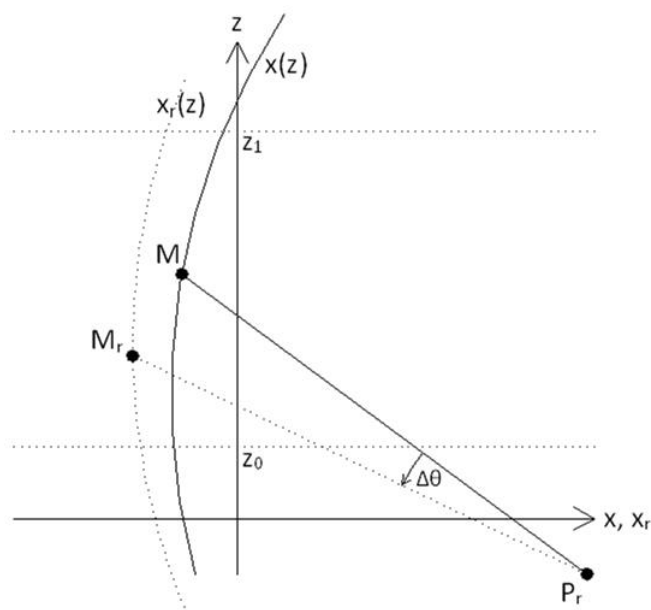


Figura 37 – Rotação de uma parábola.

3.3.5.3 Avaliação Experimental

Para medir o desempenho do sistema de predição proposto, foram realizados alguns experimentos com o veículo protótipo, configurado para capturar e gravar imagens sincronizadas com os dados de odometria. O veículo foi dirigido manualmente em três condições de curvatura como mostrado na Figura 38: (a) curva à esquerda; (b) direção reta; e (c) curva à direita. Em cada condição de curvatura, o veículo foi conduzido em três diferentes níveis de oscilação na trajetória (Figura 38): (L0) sem oscilação, onde o veículo viaja paralelamente às faixas; (L1) com baixo nível de oscilação, onde o veículo cruza ciclicamente a linha central; e (L2) com forte nível de oscilação, onde o veículo também cruza a linha central mas com um desvio mais acentuado.

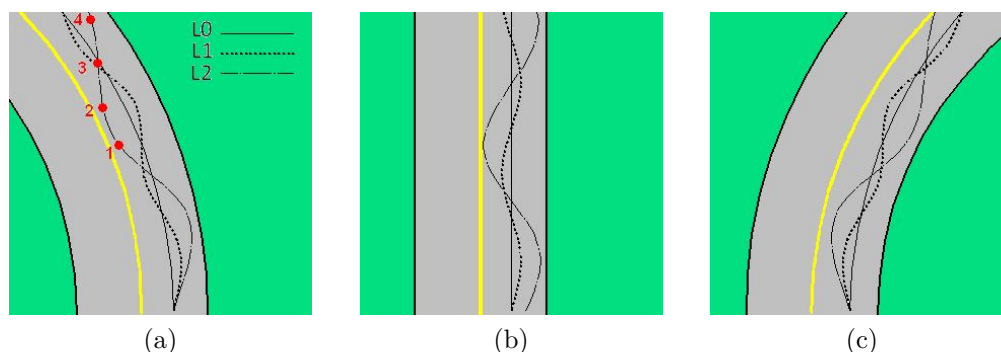


Figura 38 – Trajetos dos testes experimentais.

As oscilações foram introduzidas na trajetória para forçar o veículo a se mover de modo não paralelo às faixas para que a predição pudesse mostrar sua capacidade em manter o estado do modelo próximo às medições instantâneas. Os testes foram conduzidos

na velocidade aproximada de 40km/h em percursos de 100m de comprimento, e os dados (imagem e odometria) coletados a cada 40ms.

A Figura 39 mostra o comportamento dos coeficientes a_0 (deslocamento lateral) e a_1 (direção) do polinômio que descreve a faixa central com e sem o uso da predição. O coeficiente de curvatura a_2 não foi mostrado porque ele é praticamente constante no trecho considerado. A medição corresponde aos valores instantâneos gerados pelo sistema de visão. Pode-se notar o efeito da filtragem na remoção do ruído de alta frequência, que resulta em uma curva mais suave, porém atrasada (caso sem predição). Com o uso da predição é possível reduzir o atraso indesejado. Esta redução ocorre porque a predição compensa o efeito do movimento do carro sobre as imagens capturadas.

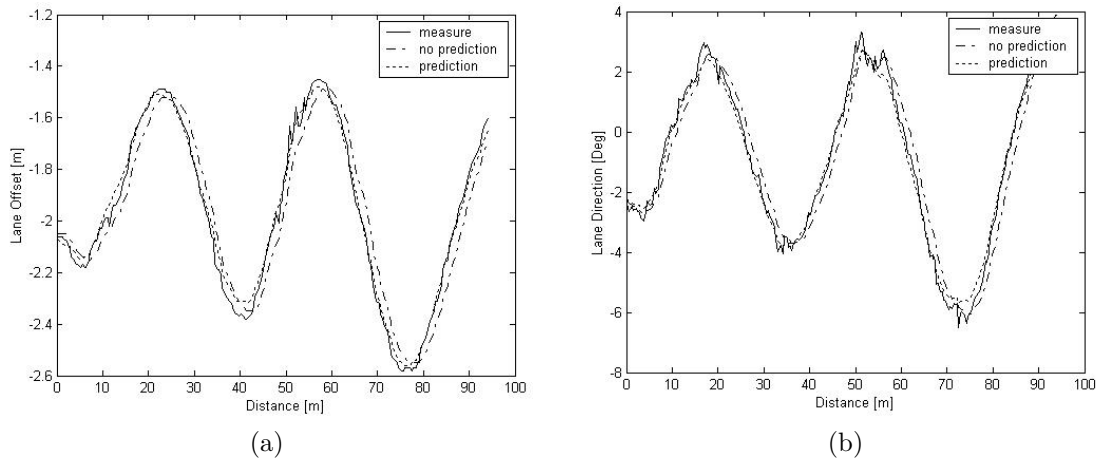


Figura 39 – Efeito da predição sobre os coeficientes do polinômio da linha central.

Para quantificar a diferença entre o estado do modelo e as medições, foi usado o índice EAH (erro absoluto horizontal), calculado como a diferença horizontal Δx entre os pontos da curva definida pelo modelo e pela medida instantânea. Para este cálculo são usados n pontos, tomados em incrementos $\Delta z = 0,5m$, conforme a expressão

$$EAH = \frac{1}{n} \sum_{i=1}^n |x_{medid,i} - x_{model,i}|. \quad (3.23)$$

Esta expressão mede o grau de casamento entre as duas curvas, em cada instante. Quanto menor for o valor, melhor. Quando o veículo está dirigindo paralelo à faixa, seja em curva ou em retas, o valor do EAH fica baixo, pois as diferenças ocorrem somente devido a imperfeições na forma da faixa e ruído. Mas quando o veículo está cruzando uma faixa (no caso de troca de pista), a posição horizontal da faixa muda constantemente em relação ao veículo e, neste caso, o valor instantâneo do EAH fica alto.

Para medir o valor médio do EAH em um trecho de estrada, e a influência do uso da predição sobre ele, seu valor foi integrado ao longo de um trecho de viagem. A Figura 40 mostra o resultado com e sem o uso da predição. Este gráfico foi gerado com os dados

do teste em curva à esquerda com oscilação nível L2 (Figura 38-a), e os valores estão normalizados em relação ao valor final sem predição. Para os dados de teste, o uso da predição reduziu o valor médio do EAH para 43,6% do valor sem predição.

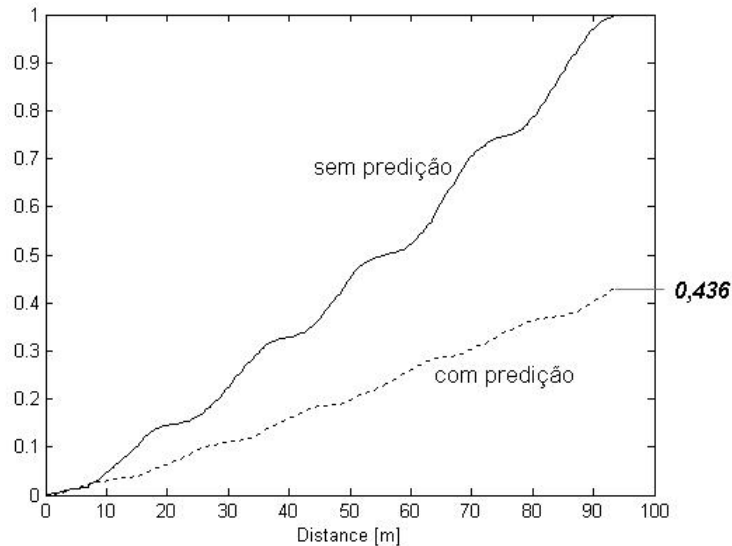


Figura 40 – Valor acumulado de EAH em um percurso de 100m.

Este cálculo também foi feito para todas as outras 8 condições de testes e os valores estão resumidos na Tabela 3.

Tabela 3 – Efeito do uso da predição na redução do erro EAH.

	<i>Esquerda</i>	<i>Reta</i>	<i>Direita</i>
L0	0,996	1,099	1,069
L1	0,549	0,503	0,589
L2	0,436	0,418	0,529

Pode-se notar que quanto mais oscilatória for a trajetória de direção, mas eficaz é o uso da predição em manter o modelo próximo às medições instantâneas sem abrir mão da robustez dada pelo processo de filtragem. Isso ocorre porque nestas condições as faixas da estrada aparentam estar se deslocando horizontalmente em relação à câmera e o uso da predição compensa internamente este efeito.

Acredita-se que o uso deste método pode melhorar o desempenho do veículo operando em modo autônomo durante as manobras de mudança de faixa.

3.4 Controle do Volante

O módulo de controle lateral usa a informação do estado da pista para dirigir o veículo. O controle é feito movimentando-se o volante de modo que o carro descreva uma trajetória circular que atinge um ponto alvo definido no meio da estrada a uma certa distância (Z_{alvo}), conforme mostrado na Figura 41.

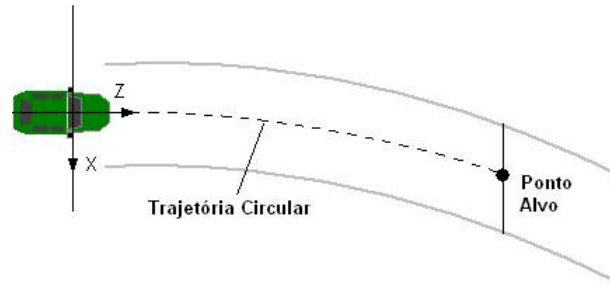


Figura 41 – Caminho circular para atingir o alvo.

A distância é definida como sendo proporcional à velocidade atual do veículo ($Z_{alvo} = 1,5v$) de modo que se o ponto alvo ficasse parado o veículo o alcançaria em 1,5s. Com o valor de Z_{alvo} calculado, o controle usa a informação do estado da pista para calcular a posição do centro (X_{alvo}), e através das equações 3.24 e 3.25 calcula o raio de curvatura em metros e o ângulo de posicionamento do volante (φ) em radianos.

$$R = \frac{1}{2} \left(X_{alvo} + \frac{Z_{alvo}^2}{X_{alvo}} \right) \quad (3.24)$$

$$\varphi = \frac{70,0}{R} \quad (3.25)$$

3.5 Testes e Resultados

A verificação experimental do funcionamento de um veículo autônomo, neste caso um automóvel em tamanho real, não é uma tarefa simples. Primeiro, por questão de segurança, como as velocidades envolvidas são altas e a massa do veículo é grande, a ocorrência de uma falha pode causar danos materiais gravíssimos, inclusive à vida. Segundo, por questão de medição de desempenho, não é fácil definir uma métrica para medir o quão bem uma pessoa ou um sistema de direção autônoma dirige.

A metodologia de testes foi proposta de modo a priorizar a segurança, aumentando-se gradativamente o grau de dificuldade. Os primeiros testes foram feitos em laboratório usando vídeo gravado, depois foram feitos os testes em modo semi-autônomo e por último em modo autônomo.

Os testes em modo semi-autônomo e autônomo foram conduzidos em estradas de baixo tráfego no interior do estado, nos horários de menor movimento e em baixa velocidade (30km/h). À medida que os *bugs* foram sendo corrigidos e o sistema se mostrando mais confiável, a velocidade foi progressivamente aumentada e estradas mais movimentadas usadas.

3.5.1 Testes com Vídeo Gravado

Conforme foi discutido na Seção 3.4, o controle da direção do veículo é feito a partir do estado do modelo, que por sua vez é atualizado pelo sistema de visão computacional. Para que o veículo tivesse alguma chance de obter sucesso nos testes de direção autônoma, era necessário certificar primeiro que o modelo de estrada estava de fato sendo atualizado corretamente com as informações visuais.

Toda esta verificação, assim como o desempenho computacional, foi feita com base em vídeos gravados pela câmera do carro protótipo. Primeiro, o veículo foi dirigido por uma pessoa pelo trecho de estrada onde pretendia-se fazer o teste de direção autônoma. Em seguida, o programa seguidor de faixas era executado, usando o vídeo salvo em arquivo, para que se pudesse avaliar o comportamento do módulo de visão computacional em conjunto com o algoritmo de filtragem. Foram observados, a capacidade do modelo em acompanhar as faixas presentes na informação de vídeo assim como a imunidade para seguir falsas faixas (como em bifurcações). Estas avaliações permitiram fazer os ajustes finos dos parâmetros do filtro. Também foram importantes para identificar situações de falha antes que se fosse para a etapa seguinte de testes com direção semi-autônoma.

Um desses testes foi feito com vídeo gravado na avenida Dante Micheline, em Camburi, e pode ser visualizado em <http://youtu.be/mwSNEWhbJhg>. O vídeo mostra detalhes do algoritmo operando através de três janelas, como na Figura 42. A janela de cima mostra a imagem da câmera e os fragmentos de faixas sendo detectados. A de baixo mostra o gráfico que registra o comportamento da qualidade da informação visual detectada, sendo em preto a qualidade do modelo, em verde o valor instantâneo da medição esquerda e em laranja o da direita. A janela da esquerda mostra a projeção perspectiva inversa, com destaque em vermelho da melhor representação.

A Figura 43 mostra alguns exemplos do sistema operando em diversas condições onde pelo menos em um lado a posição da linha foi construída a partir da informação de duas faixas. Em verde temos todos os fragmentos que foram detectados, em vermelho fino, o segundo melhor caminho gerado, em vermelho grosso o melhor caminho, e em branco fino o estado do modelo. Na Figura 43-a a representação final da faixa direita foi feita misturando-se a informação da faixa direita com a do meio fio. Na Figura 43-b foi detectada a condição de "quebra" de direção no meio fio direito, de modo que apenas a parte mais próxima foi usada para o cálculo da representação final da faixa direita. A faixa esquerda foi gerada a partir da concatenação dos pedaços de faixas intermitentes.

Na Figura 43-d, mostra o sistema operando em uma rodovia onde a construção da faixa esquerda foi feita aproveitando-se as duas faixas centrais amarelas detectadas. A Figura 43-e ilustra a robustez do sistema, mostrando que ele é capaz de processar a imagem em uma região de sombra. Finalmente, a Figura 43-f mostra o sistema usando



Figura 42 – Um quadro do vídeo demonstrativo do algoritmo de detecção visual operando: acima - imagem da câmera; abaixo Gráfico da qualidade do modelo; a direita projeção perspectiva inversa. Acessível em <<http://youtu.be/mwSNEWhbJhg>>

parte da lateral da ponte para construir a representação final.

O desempenho computacional do algoritmo foi medido em termos do tempo de execução de um quadro de vídeo. Foi utilizado um notebook com processador Intel Celeron 32 bits de 1,86 GHz, com sistema operacional Windows XP. Os resultados estão na Tabela 4.

Tabela 4 – Desempenho Computacional para um Quadro de Vídeo

#	Processo	Tempo (ms)
1	Binarização da imagem	3,5
2	Associação dos pontos (max.)	0,7
3	Construção da melhor representação (max.)	0,5
4	Filtragem	0,3
Total		5,0

3.5.2 Testes de Direção Autônoma

Por se tratar de um campo novo de pesquisa, ainda não há nenhuma norma que estabelece os procedimentos para realização de testes de direção autônoma no Brasil. Os procedimentos adotados neste trabalho foram baseados nos trabalhos científicos citados e nas normas de segurança de competições de veículos robóticos, como DARPA. Os procedimentos fundamentais adotados são: (i) no caso de teste em vias públicas, sempre ter



Figura 43 – Exemplo de reconstrução dupla de faixas.

um motorista de segurança sentado atrás do volante; (ii) o veículo deve dispor no mínimo de um botão ou sistema de emergência, facilmente acessível, que serve para desativar imediatamente todo o sistema de acionamento autônomo, deixando o carro livre para ser dirigido em modo manual.

A intervenção humana no volante é a questão mais importante que afeta a segurança em um teste. No caso de operação em vias públicas de mão dupla, a intervenção do volante é absolutamente crítica pois um comando de posicionamento errado, gerado pelo programa de direção autônoma, pode levar a uma colisão frontal. Em diversos projetos, o acionamento do volante é feito diretamente no motor elétrico de assistência do volante. Neste caso, devido ao grande torque do motor, o motorista não tem a capacidade de sobrecomandá-lo, portanto, a desenergização deve ser feita o mais rápido possível através de um botão.

O sistema de acionamento do volante do carro protótipo foi projetado cuidado-

samente para tornar a operação o mais segura possível. Ele gera pouco mais do que o torque mínimo necessário para movimentar o volante, desse modo, se o motorista precisar de reagir a um comando errado do computador, basta ele segurar o volante e girá-lo. O sistema de desarme (Capítulo 2) detecta que a posição do motor de passo foi alterada e desativa o acionador imediatamente. Esta reação do motorista, além de desernegizar o volante, salva os últimos 5 segundos de vídeo antes da falha, que podem ser usados para depuração do algoritmo.

Os primeiros testes foram de curta distância, realizados em uma rodovia de baixo movimento, em baixa velocidade, nos horários de mínimo movimento (domingo de manhã), em uma rodovia estadual com boa marcação de pista. A rodovia escolhida foi a ES264, próximo a localidade de Lagoinha (S41,064630; W20,237158) no município de Afonso Claudio, ES. Estes primeiros testes permitiram avaliar o efeito do acionador do volante na resposta do sistema de controle em malha fechada, e também, permitiu o ajuste de alguns parâmetros empíricos. Foi observado o comportamento do veículo em trechos retos e em curva, nas velocidades de 40km/h e 60km/h. Esta primeira fase de testes está detalhada em (VIVACQUA; MARTINS; VASSALLO, 2012).

A continuação dos testes foi feita em uma condição de dificuldade maior, em uma rodovia com trechos em subida e descida, falhas nas marcações de pista e numa velocidade maior (80km/h). O local escolhido foi a rodovia federal BR262, entre os municípios de Venda Nova do Imigrante e Marechal Floriano-ES. O teste apresentado (Figura 44) foi feito em um trecho de 5,7 km iniciando na coordenada (S20,4136112; W40,8801696). O trajeto contém 7 curvas à esquerda (total 357°), 6 curvas à direita (total 253°) e foi realizado com a velocidade máxima de 80km/h. Durante todo o trajeto, foi necessário apenas uma intervenção humana para desviar de um veículo que ultrapassava na contra-mão. Seguindo o protocolo de segurança, o motorista manteve a mão sobre o volante (mas sem exercer força alguma) ao cruzar com outros veículos.



Figura 44 – Percurso do teste com direção autônoma.

A Figura 45 mostra algumas imagens capturadas do interior do carro protótipo

durante a realização desse experimento. O vídeo completo pode ser acessado em (<http://www.youtube.com/watch?v=otWD6s8T3Js>). Este teste foi interrompido no km 5,7 porque o carro protótipo perdeu o *track* visual da faixa, provocado pela combinação de uma forte trepidação com a mudança de inclinação da pista em uma curva.

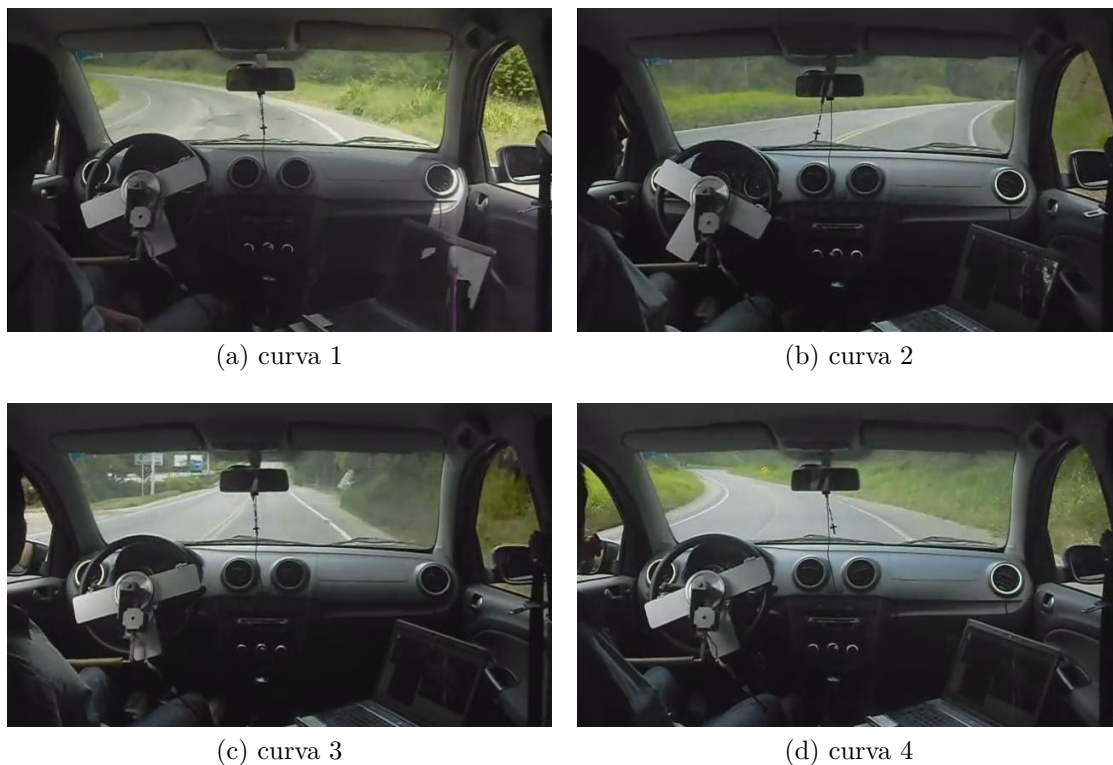


Figura 45 – Teste com direção autônoma, acessível em <http://www.youtube.com/watch?v=otWD6s8T3Js>.

3.6 Conclusões

Diversos testes com distância superior a 5km foram realizados sendo o mais longo, com 12,7km, interrompido pelo término da pintura da faixa em um trecho em obras. Estes testes demonstraram que, se a estrada oferecer as condições básicas de direção, como pelo menos uma marcação de pista sem falhas superiores a 15m, curvas não muito acentuadas, o veículo é capaz de segui-la indefinidamente. Note que no caso de uma *highway*, onde a maioria dos trabalhos relacionados testaram seus algoritmos, estas condições são atendidas com folga.

Os testes com vídeo gravado, em condição urbana, mostraram que, o uso de até quatro faixas para construção da melhor representação, mostrou uma grande capacidade do sistema se adaptar a condições mais variadas de faixas, o que representa um ganho de robustez ao algoritmo. Os testes com direção em zig-zag, demonstraram que o uso da predição da posição da linha central contribuiu para reduzir o tempo de resposta

do filtro sem alterar sua imunidade a ruído. A arquitetura leve e os algoritmos rápidos desenvolvidos foram testados em tempo real em um sistema de computação modesto, o que demonstra que estes algoritmos também podem ser implementados em sistemas de computação embarcados.

Assim, considerou-se que o algoritmo proposto atendeu ao seu objetivo inicial, mostrando-se leve, robusto e preciso. Considerou-se também que o algoritmo implementado possui qualidades suficientes para ser utilizado em um ADAS, na geração de alerta de saída de pista. O algoritmo desenvolvido foi aproveitado como módulo de detecção de marcações de pista no algoritmo de direção autônoma por mapeamento e localização apresentado no Capítulo 5.

4 Direção Autônoma Baseada em Fusão GNSS e *Dead Reckoning*

4.1 Introdução

O sistema de localização global por satélite (GNSS) permite a medição direta da posição de um veículo na superfície da terra. O erro de localização de um receptor GNSS comum pode variar de aproximadamente 3m, em condições excelentes, a dezenas de metros em condições menos favoráveis, como em meio a edifícios altos. Isso torna este tipo de receptor incapaz para realizar direção autônoma em rodovias, onde o erro de localização deve ser de poucos centímetros. Receptores GNSS especiais tais como GNSS com correção RTK permitem que, em certas circunstâncias, esse nível de precisão seja alcançado. A fusão de um receptor GNSS com sensores inerciais também é uma maneira de diminuir o erro de localização sem depender de um sinal especial de correção, fornecido externamente por estações base. Porém, este sistema não consegue eliminar o erro de *bias* devido á variação lenta das condições atmosféricas.

Neste capítulo será apresentado o estudo e a implementação de um sistema de localização global baseada na fusão das medidas de GNSS com *dead reckoning*. O sistema *dead reckoning* desenvolvido aqui foi aproveitado no sistema de direção autônoma por mapas de marcações de pista (apresentado no próximo capítulo). Ao final deste capítulo o sistema desenvolvido foi colocado em prova em um teste de direção autônoma para repetição de um trajeto previamente determinado. A visão geral do sistema de direção autônoma por *dead reckoning* e GNSS está apresentada na Figura 46.

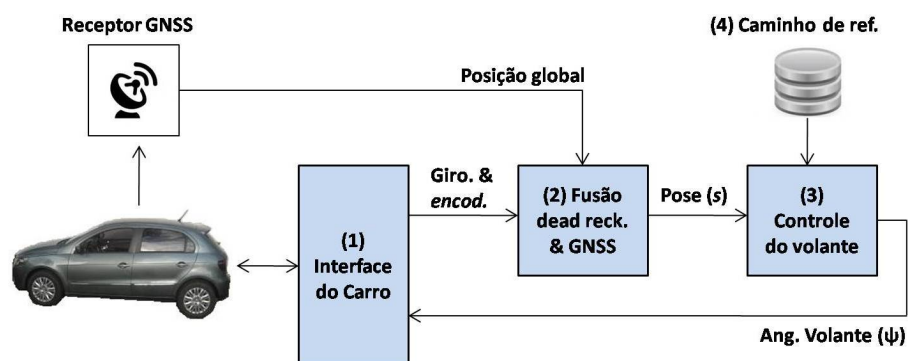


Figura 46 – Arquitetura do sistema de direção autônoma por *dead reckoning* e GNSS.

O Módulo 1 é responsável pela interface com os sinais elétricos do carro, assim como o controle de posicionamento do volante (detalhes na Secção 2.2). O Módulo 2 faz a fusão dos sinais de *dead reckoning* (giroscópio e *encoder*) com as medições do receptor

GNSS para estimar a pose global do veículo ($\mathbf{s} = (x, y, \theta)$). O Módulo 3 é o controlador de seguimento de caminho, responsável por calcular o ângulo de posicionamento do volante a partir da pose atual e do caminho de referência. O Módulo 4 é o caminho de referência a ser seguido autonomamente, gravado em arquivo.

4.2 Sistema *Dead Reckoning*

Dead Reckoning (navegação estimada) é um método bastante usado para se estimar a posição de um veículo através de sensores proprioceptivos (sensores internos) tais como sensores inerciais e de deslocamento incremental (odômetro). Para pequenas distâncias *dead reckoning* possui grande precisão e velocidade de resposta, porém com o passar do tempo a precisão cai devido aos erros acumulativos em seu cálculo.

O sistema *Dead Reckoning* implementado no veículo protótipo estima a localização 2D do veículo (x, y) e seu ângulo de orientação (θ). Estas três informações juntas determinam a pose (ou estado) do veículo ($\mathbf{s} = (x, y, \theta)$). As informações usadas para o cálculo do *Dead Reckoning* são fornecidas pelos sensores proprioceptivos do carro protótipo, transmitidas através do módulo de interface (Módulo 1 da Figura 46).

A Figura 47 ilustra a mudança de estado do veículo do estado \mathbf{s}_i para o estado \mathbf{s}_{i+1} . Este processo é calculado na frequência de 25Hz, o que significa que a 90 km/h, considerada uma velocidade de operação alta, os incrementos lineares serão de apenas 1,0m. Note que pequenos incrementos lineares estão associados a pequenos incrementos angulares, que leva as expressões simplificadas de transição de estado (4.1,4.2,4.3).

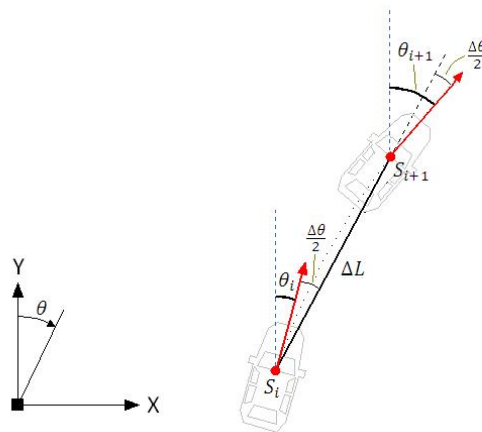


Figura 47 – Transição de Estado.

A equação de transição de estado $\mathbf{s}_{i+1} = \mathbf{f}(\mathbf{s}_i, \Delta\theta, \Delta L)$ é definida pelo conjunto de expressões 4.1 a 4.3, onde $\Delta\theta$ é o incremento angular fornecido pelo giroscópio e ΔL o

incremento linear fornecido pelo *encoder*.

$$x_{i+1} = x_i + \Delta L \sin(\theta_i + \Delta\theta/2) \quad (4.1)$$

$$y_{i+1} = y_i + \Delta L \cos(\theta_i + \Delta\theta/2) \quad (4.2)$$

$$\theta_{i+1} = \theta_i + \Delta\theta \quad (4.3)$$

O deslocamento linear ΔL é medido diretamente pela contagem de pulsos (N_p) do *encoder* montado no eixo dianteiro do veículo. O fator multiplicador foi definido a partir do diâmetro do pneu e da quantidade de pulsos por volta do *encoder*. O deslocamento linear em metros é calculado por: $\Delta L = 0,2674N_p$.

O incremento angular é calculado a partir do sensor giroscópio, conforme detalhado na próxima secção.

4.2.1 Medição do Incremento Angular

A medição do deslocamento angular correspondente à mudança de direção do veículo é feita pela integração do sinal de velocidade angular (ω_z) gerado pelo eixo-z do giroscópio, subtraído do valor de *bias* (ω_{bias}), conforme a Equação 4.4. O valor de *bias* corresponde ao valor gerado na saída do dispositivo quando sua velocidade angular é nula. Este valor deveria ser zero, mas no caso dos dispositivos que usam tecnologia MEMs (*micro-eleto-mechanical-sensors*), pequenas tensões mecânicas sobre o *chip* causam assimetrias nas estruturas vibratórias internas do giroscópio que levam a um valor não nulo.

$$\Delta\theta_i = k_z \int_{t_{i-1}}^{t_i} (\omega_z(u) - \omega_{bias}) du \quad (4.4)$$

O valor do *bias* sofre forte influência da temperatura de operação e esta relação depende da montagem do dispositivo na placa de circuito impresso. Por esta razão não pode-se comprar dispositivos compensados de fábrica. Para tirar o máximo de desempenho de um dispositivo em particular, é necessário fazer o levantamento experimental da relação *Bias* x Temperatura após a montagem final do dispositivo.

O sensor inercial MPU6000 usado no carro protótipo possui um sensor de temperatura interno, que pode ser usado para compensar o efeito da temperatura sobre o *bias*. A Figura 48 mostra o levantamento experimental da relação *Bias* x Temperatura para o sensor que foi usado no carro protótipo. O valor do *bias* foi calculado em intervalos de 10s com o sinal do giroscópio amostrado em 125Hz. Pode-se notar que para uma variação de 10°C o *bias* variou 8,8°/min.

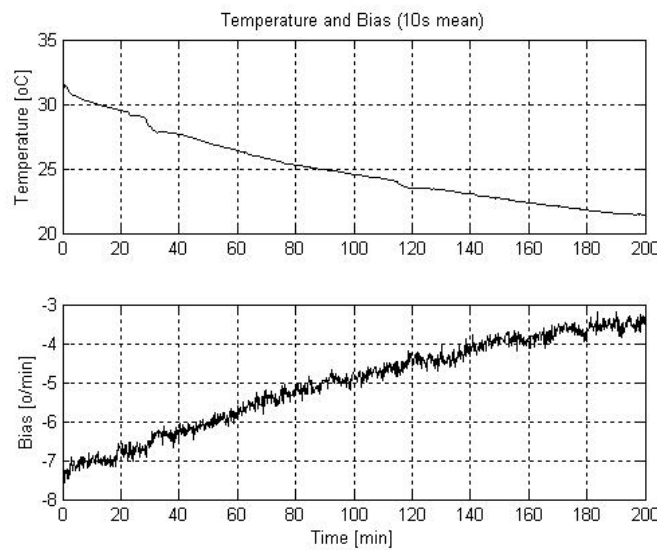


Figura 48 – Influência da temperatura no *bias* do giroscópio.

Há duas maneiras comuns de se fazer o cancelamento do *bias*: (1) levantando-se experimentalmente a curva *Bias* x Temperatura e usando depois esta informação para compensação; (2) medir o *bias* para uma determinada temperatura de operação e depois manter o dispositivo operando nesta mesma temperatura por meio de controle externo de temperatura. Esta segunda opção foi adotada para o veículo protótipo, pois tem a vantagem de eliminar também a influência da temperatura sobre outros parâmetros do sensor, como por exemplo a sensibilidade.

O correto cancelamento do *bias* é bastante crítico no processo da navegação por *dead reckoning* pois seu efeito é acumulativo. Considere a simulação mostrada na Figura 49 (fora de escala), onde o carro percorreu um trajeto reto com 500m de comprimento a 36km/h. Um erro no cancelamento do *bias* do giroscópio de apenas $1^\circ/\text{min}$, provoca no final do percurso um erro lateral de 3,6m.

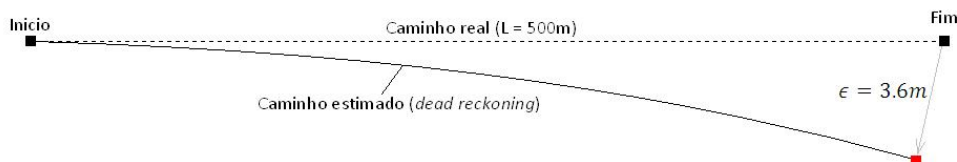


Figura 49 – Erro causado pelo *bias* em um trajeto retilíneo.

4.3 Fusão com GNSS

Como já foi dito, a estimativa de localização feita por *dead reckoning* está sujeita a erro acumulativo que cresce indefinidamente com o passar do tempo e/ou distância percorrida. O uso do receptor GNSS permite limitar o crescimento desse erro uma vez

que esse tipo de sensor possui erro limitado. A união (fusão) desses dois tipos de sensores aproveita as melhores características de cada sistema e, conseqüentemente, produz uma estimativa de localização mais precisa, contínua e robusta.

O algoritmo geral da fusão GNSS/*dead reckoning* é formado por operações de predição (Equação 4.5) e de atualização (Equação 4.6). A predição nada mais é do que o cálculo de *dead reckoning* usando as medições dos sensores proprioceptivos, em nosso caso giroscópio e *encoder*, e calculada por

$$\mathbf{s}_{i+1} = \mathbf{f}(\mathbf{s}_i, \Delta\theta, \Delta L). \quad (4.5)$$

A atualização (ou *update*) é a correção da posição atual com medidas geradas por um ou mais sensor exteroceptivo, em nosso caso o GNSS, e calculada por

$$\mathbf{s}_{i+1} = \mathbf{s}_i + K(\mathbf{s}_{gnss} - \mathbf{s}_i), \quad (4.6)$$

onde \mathbf{s}_i é o estado de localização do veículo, s_{gnss} é a medida de posição do GNSS, K é a matriz de ganho de atualização.

Em certos casos, a fusão é feita considerando a distribuição de erro dos sensores, o que permite não somente calcular a posição atual do veículo, mas também o grau de confiabilidade da estimativa a cada instante. Quando os sensores utilizados estão sujeitos a erro com distribuição gaussiana, uma estrutura de fusão bastante utilizada é o filtro de Kalman.

Em nosso caso, como não conhecemos a distribuição do nosso receptor GNSS, optamos por uma estrutura de filtragem simples e não probabilística, com matriz de ganho diagonal e com fatores iguais e constantes. A escolha do valor exato do ganho foi feita de modo a minimizar o erro do cálculo de uma posição conhecida em um teste de múltiplas voltas em um circuito fechado (ver Figura 52).

Os dados experimentais brutos de giroscópio, odometria e medições de GNSS foram coletados com o veículo protótipo e depois usados *offline* para rodar o algoritmo de filtragem, com diversos valores de ganho. A Figura 50 mostra graficamente o efeito do espalhamento do ponto conhecido. O espalhamento foi medido por meio da variância de x e y , e o valor de ganho escolhido foi o que gerou o menor a variância. Neste caso o valor $K = \text{diag}([0, 016; 0, 016; 0, 016])$ produziu uma variância de $0,62m$.

4.4 Testes e Resultados

Para avaliar o desempenho do sistema de localização e direção autônoma apresentado neste capítulo, foram realizados dois experimentos. O primeiro para avaliar o

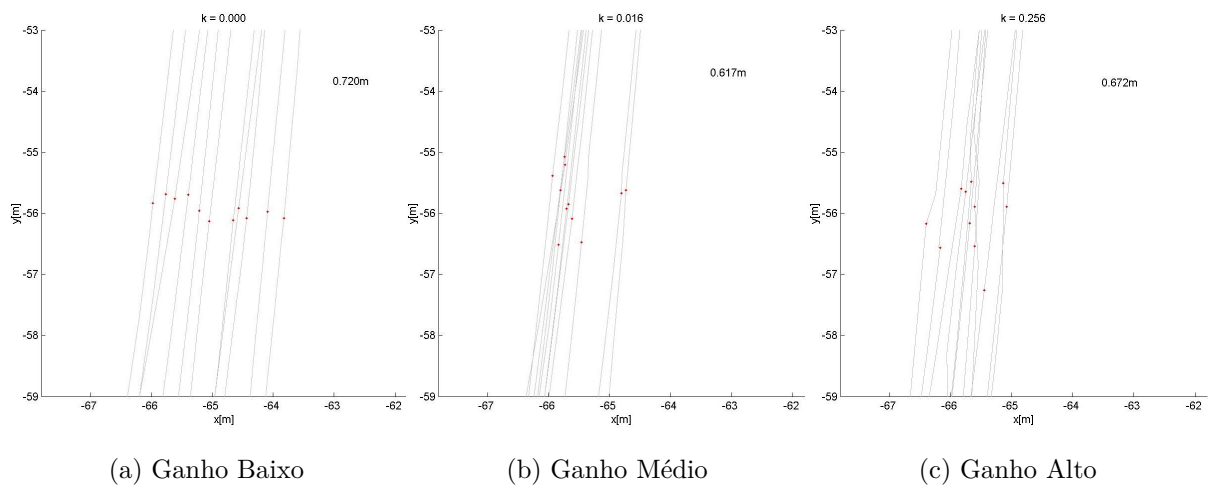


Figura 50 – Efeito de variação do ganho de *update* no espalhamento dos pontos de referência.

desempenho do sistema *dead reckoning* separadamente e o segundo para avaliá-lo em conjunto com GNSS em um teste de direção autônoma em um trajeto cíclico.

4.4.1 Desempenho do Sistema *Dead Reckoning*

O desempenho do sistema *Dead Reckoning* foi avaliado por meio de dois testes em caminho fechado, um para medição do erro total, e outro para medição de erro lateral em múltiplas voltas.

O erro total foi medido no percurso mostrado na Figura 51, com comprimento total de 556 m, feito na velocidade média de 30 km/h. Neste teste o motorista iniciou e terminou a viagem na mesma posição, indicada por uma marca pintada no chão. As medições de GNSS foram colocadas na Figura 51 para efeito de comparação. O erro total, calculado por $\epsilon_f = \sqrt{x_f^2 + y_f^2}$, foi de 0,56m ($\approx 0,1\%$). Este percurso foi repetido uma segunda vez, e o novo valor de erro encontrado foi 0,67m.

O erro lateral acumulado em múltiplas voltas foi medido em um percurso de 350,2m (52-a). Uma marca foi pintada no chão para indicar ao motorista o exato local onde o veículo devia parar a cada volta. Toda vez que o veículo cruzava esta linha, sua posição estimada por *dead reckoning* era registrada. A Figura 52-b mostra o resultado obtido, onde percebe-se claramente que efeito acumulativo de erro no processo de *dead reckoning* produz um erro lateral crescente.

O espalhamento desses pontos foi usado como critério para escolha de um dos parâmetros do filtro de fusão do GNSS com *dead reckoning*, explicado na próxima secção.

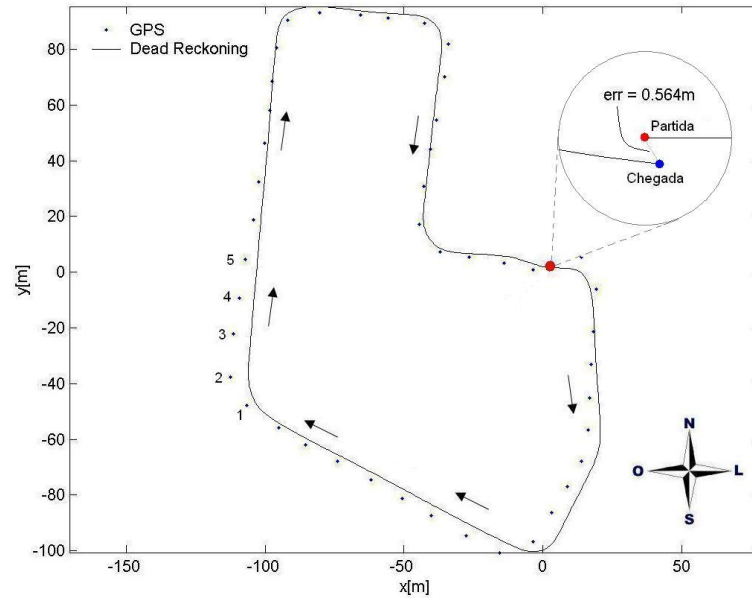


Figura 51 – Caminho usado no teste de circuito fechado.

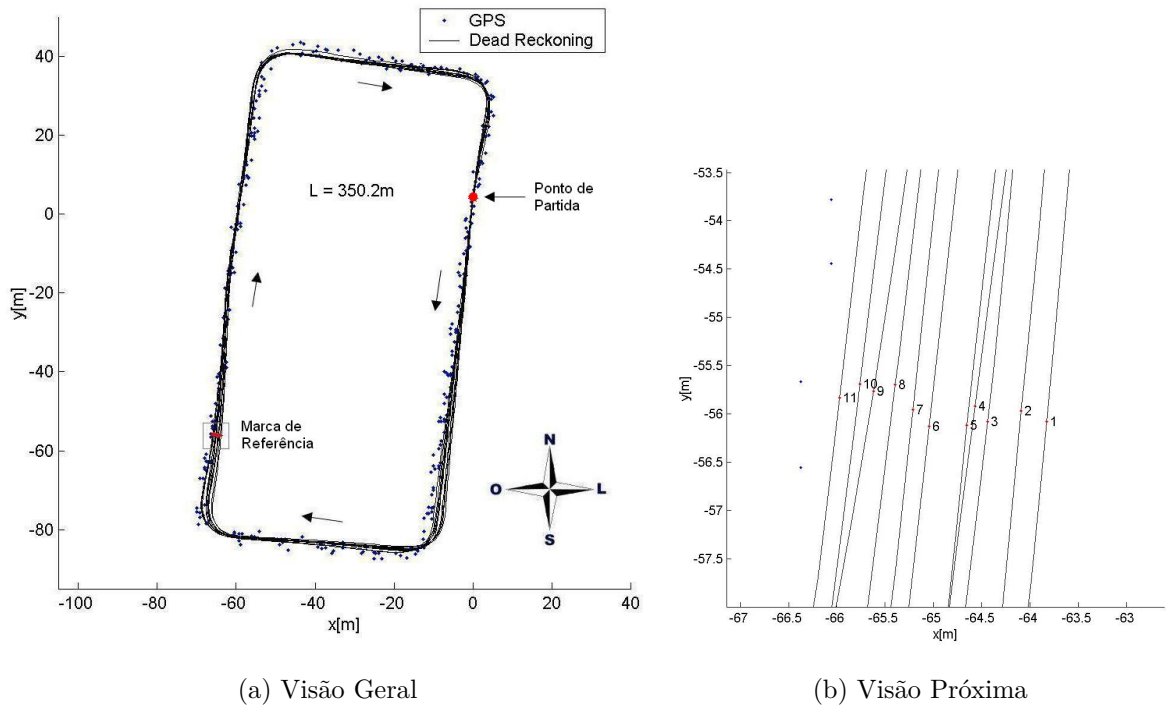


Figura 52 – Desvio acumulado.

4.4.2 Direção Autônoma Baseada em Fusão GNSS e *Dead Reckoning*

A fusão da informação do receptor GNSS com o sistema *dead reckoning* permite limitar o crescimento do erro de localização, uma vez que o sistema GNSS possui erro limitado. Com isso obtém-se uma estimativa de localização que incorpora as vantagens do *dead reckoning* (alta taxa de amostragem, continuidade, alta precisão para pequenos deslocamentos) com a principal vantagem do sistema GNSS (erro absoluto, limitado). Os

experimentos descritos nesta seção têm como principal objetivo avaliar o desempenho de um sistema de direção autônoma, em uma implementação baseada em sensores de baixo custo, e filtragem simples para fusão de GNSS e *dead reckoning*, descrita na seção 4.3. Adicionalmente este experimento permite avaliar o efeito do *drift* rápido (em 10 minutos) do *bias* do sinal de GNSS, e seu efeito na repetição autônoma de um percurso cíclico. O procedimento de teste se divide em duas partes: (a) construção do caminho de referência (caminho a ser copiado pelo carro), quando um motorista percorre o trajeto escolhido e o computador salva os pontos que descrevem a trajetória; (b) execução automática deste caminho, quando o computador controla o volante para que o carro repita o caminho de referência.

Após a construção do caminho de referência, a direção autônoma inicia com o posicionamento do veículo exatamente na origem do caminho. Para isso utilizam-se marcações auxiliares mostradas na Figura 53.

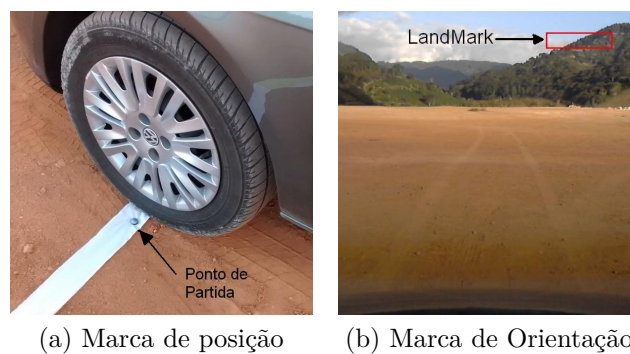


Figura 53 – Marcas para auxiliar o posicionamento inicial do veículo.

O controlador do volante (Figura 46 - Módulo 3) tem seu princípio de funcionamento ilustrado da Figura 54. O ângulo de posicionamento do volante, calculado pela expressão $\varphi = 12 \sin \alpha$, considera os parâmetros cinemáticos do carro protótipo e o ângulo de desvio da direção α ao ponto alvo. O ponto alvo se mantém a uma distância temporal constante de 1,5s.

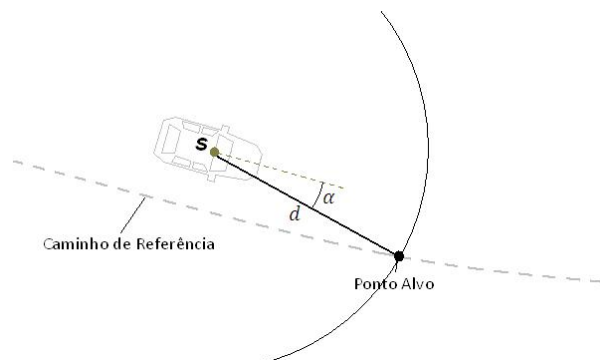
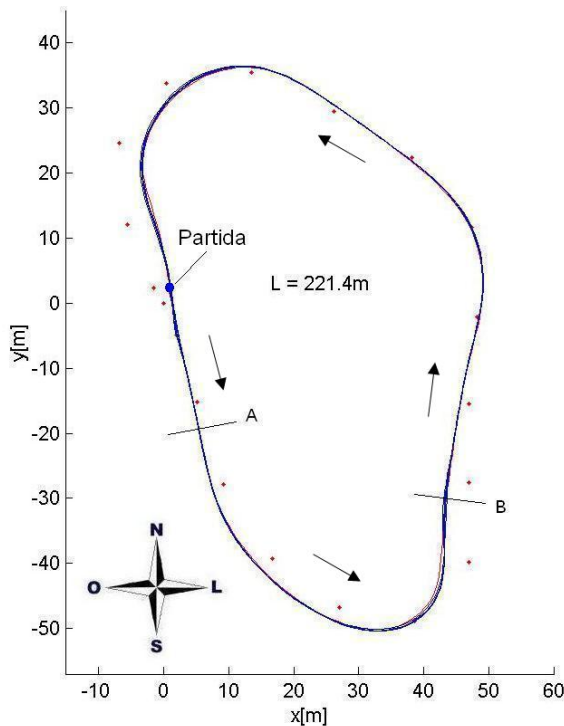


Figura 54 – Princípio de funcionamento do controlador do volante.

Ao todo foram realizadas 9 repetições autônoma consecutivas do caminho de referência (Figura 55). A distância total percorrida foi de 2km e a duração do teste foi de 10 minutos. A Figura 55-a mostra o registro do caminho estimado pelo sistema de localização do veículo e a Figura 55-b, o caminho real registrado por uma imagem aérea.



(a) Registro Estimado



(b) Registro Real

Figura 55 – Caminho dirigido em modo autônomo.

A medição do erro lateral foi feita em dois pontos específicos do circuito, mais exatamente dois cortes transversais (cortes A e B - Figura 55-a). Os resultados obtidos estão apresentados na Figura-56. Para o corte A o erro lateral do controlador foi 0,10 m e o erro lateral real 0,70 m. Para o corte B o erro lateral do controlador foi 0,48 m e o erro lateral real 1,15 m.

É importante ressaltar que o desvio lateral real é causado por duas razões: erro de seguimento de caminho; e erro na localização. Os desvios observados nas Figuras 56 (a) e (c) são causados pelo erro de seguimento de caminho enquanto os desvios observados nas Figuras (b) e (d) são os erros reais observados, causados pela soma do erro de seguimento de caminho e o erro de localização. Pode-se notar que o erro real é bem maior que o erro de seguimento de caminho. Isso confirma o que foi discutido no início deste capítulo, que o erro de localização corresponde à maior parcela do erro total.

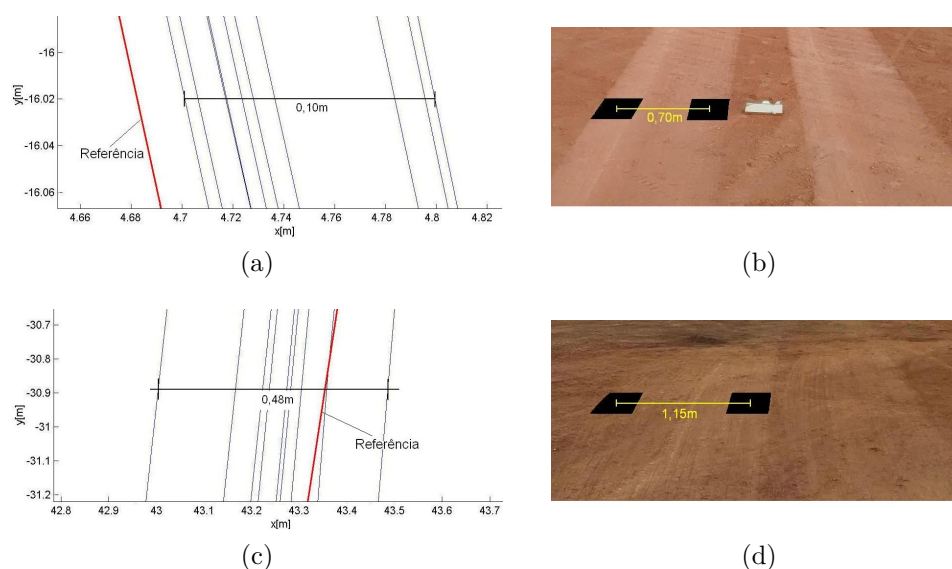


Figura 56 – Erros laterais: (a) Erro do controlador no corte A, (b) Erro real no corte A, (c) Erro do controlador no corte B, (d) Erro real no corte (B).

4.5 Conclusões

Neste capítulo foi apresentado o desenvolvimento de um sistema de localização baseado na fusão GNSS e *dead reckoning*. Os sensores para *dead reckoning* foram construídos a partir de um giroscópio MEMs, o odômetro original do veículo e um microcontrolador. O giroscópio operando em temperatura controlada permitiu uma precisa compensação do seu *bias*, e com isso minimizar a taxa de crescimento do erro de *dead reckoning* ($\approx 0,1\%$). Consideramos este valor um bom resultado tendo em vista que foram usados sensores comuns e de baixo custo.

O teste de direção autônoma em um circuito cíclico demonstrou, o que já era esperado, que a fusão GNSS com *dead reckoning* permite limitar o crescimento acumulativo de localização, intrínseco do *dead reckoning*. Com o erro acumulativo de *dead reckoning* controlado, o erro de localização passou a ser predominantemente definido pelo *bias* do giroscópio. O *bias* do receptor GNSS varia lentamente em função das condições atmosféricas e da geometria dos satélites, e só pode ser calculado com o uso de um sinal de correção, fornecido por uma estação fixa. Apesar do nosso experimento de direção autônoma ter sido conduzido em um tempo relativamente pequeno (10 min) o efeito da variação lenta do *bias* do GNSS já pôde ser percebido (deslocamento lateral dos cortes A e B - Figura 56).

Embora seja de conhecimento público as limitações da abordagem de localização baseada em GNSS, os estudos e experimentos realizados permitiram avaliar e aprimorar o sistema *dead reckoning* construído para o carro protótipo, preparando sua integração com o sistema desenvolvido no Capítulo 5. Além disso, também foi possível verificar na prática o funcionamento do controlador de seguimento de caminho proposto.

5 Direção Autônoma Baseada em Mapas de Marcadores de Pista

5.1 Introdução

Apesar dos bons resultados obtidos com o algoritmo de direção autônoma apresentado no Capítulo 3, ele estava limitado a ser um seguidor automático de pista, sem a capacidade de lidar com trechos com falta de marcação, rotatórias ou bifurcações. Os resultados obtidos com a abordagem do Capítulo 4, confirmaram que a precisão alcançada com este tipo de técnica não permite operação autônoma em vias públicas. Isso motivou o desenvolvimento do algoritmo apresentado neste capítulo, baseado nos conceitos estudados e na *hardware* e *software* desenvolvidos descritos nos capítulos anteriores. O presente algoritmo é baseado em mapas e localização, e é capaz de realizar a direção autônoma através do seguimento automático de uma trajetória pré-programada, criada em uma viagem de mapeamento manual.

Ao contrário do que normalmente é feito, nesta tese não houve a preocupação em se calcular a posição global do veículo, ou seja, a posição do veículo em relação ao globo terrestre (latitude e longitude). O termo localização se refere à postura do veículo (x, y, θ) em relação ao sistema de referência do mapa que está sendo utilizado. Para fazer direção autônoma, basta que se conheça a posição relativa entre o veículo e os limites laterais da pista, o que não varia para mapas em sistemas de referência diferentes.

O mapa de marcações de pista é construído por *dead reckoning* e não considera a informação de altitude. Se este mapa for comparado com outro construído a partir de GNSS de precisão (Figura 57), descartando a elevação e curvatura da terra, e, se forem sobrepostos dois pontos correspondentes quaisquer (aplicando rotação e translação), percebe-se que os dois mapas serão geometricamente muito parecidos em torno do ponto de sobreposição. É isso que permite que o método proposto nesta tese funcionar bem. Na medida que se afasta desse ponto, as diferenças aumentam devido ao acúmulo de erro no *dead reckoning*, o que não é um problema pois apenas a informação próxima a localização atual é usada para direção autônoma.

O novo sistema, apresentado neste capítulo, é baseado no detector de marcação do Capítulo 3, mas com duas diferenças importantes: (i) a detecção das marcações de pista passou a ser realizada apenas na região logo à frente do veículo; e (ii) o registro das marcações detectadas é armazenado num *buffer* de modo a ser utilizado para fins de localização (ver Figura 58).

Essas duas características juntas e a filosofia de baixo custo, herdada do seguidor de



Figura 57 – Distorção geométrica (erro acumulativo) em nosso esquema de mapeamento baseado em *dead reckoning*

pista, resultou na principal contribuição deste trabalho: um método baseado em sensores de baixo custo que permite construir uma percepção precisa, confiável e extensa das marcações de pista na vizinhança do veículo. Esta extensa percepção combinada com o algoritmo rápido de filtragem e *map-matching* levou a uma localização com precisão alta suficiente para operação em direção autônoma.

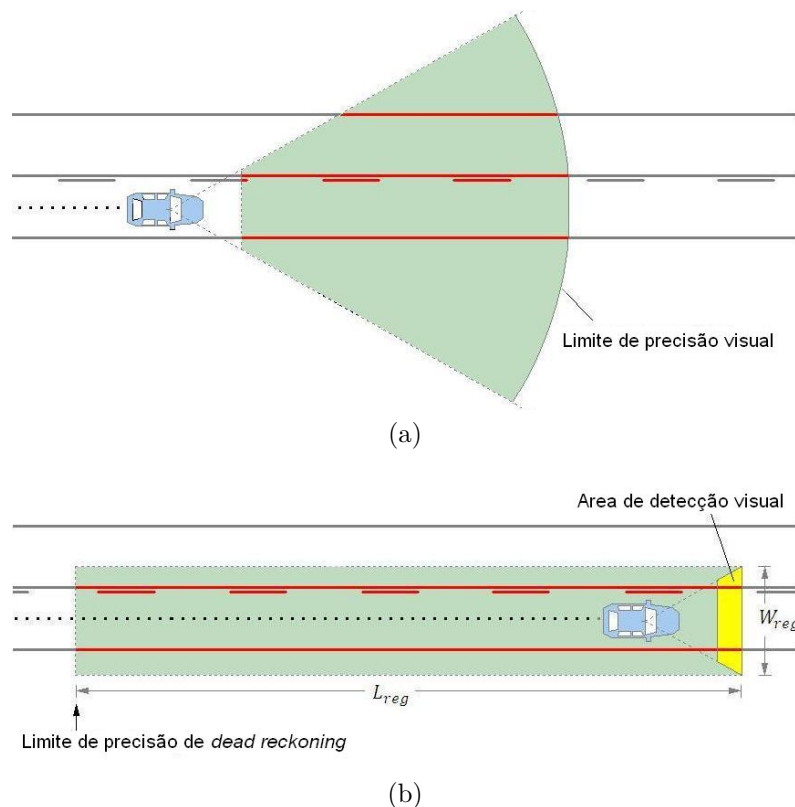


Figura 58 – (a) típica detecção frontal com alcance limitado pela precisão visual e (b) nossa proposta de detecção e reconstrução por *back registry* com o alcance limitado pela precisão do *dead reckoning*.

A escolha de detectar marcações de pista perto do veículo foi feita a fim de obter o máximo de precisão. De fato, para marcações de pista próximas ao veículo a resolução

óptica é maior. Outra vantagem é que, o cálculo da posição das marcas de pista é menos afetado pelos movimentos de veículos, como *pitch* e *roll*, que normalmente afetam dinamicamente a calibração do sistema de visão. A hipótese de uma estrada plana, usada na transformação perspectiva inversa, é mais exata nas regiões próximas ao veículo. Além disso, marcações de pista perto do veículo, geralmente não são afetadas por oclusões. No caso de operação noturna, a detecção de marcações de pista próximas ao veículo ainda oferece outra vantagem, ela focaliza a região que é mais bem iluminada pelo farol.

É importante ressaltar que, diferentemente dos seguidores de pista tradicionais, o novo sistema proposto utiliza as marcações de pista apenas como pontos de referência para o mapeamento e localização, e não para definir diretamente um ponto de alvo ou ponto de fuga, geralmente apresentados na literatura como uma ferramenta para o cálculo do comando de direção para o carro.

Portanto, quando presentes, as marcações de pista detectadas são armazenadas em um *buffer*, e sua posição em relação ao veículo continuamente atualizadas de acordo com os movimentos do veículo (ver Figura 58-b). Este conjunto de dados acumula as marcações de pista em um mapa, enquanto o veículo está viajando. Além da informação visual, dados sobre os movimentos do veículo e trajetória são obtidos por um processo de *dead reckoning*.

Uma vez que o mapa esteja construído, é possível usar o que foi chamado nesta tese de BLMR (*Back Lane Marking Registry*) para realizar a localização *online*. Os resultados preliminares demonstraram que a precisão é suficientemente alta para aplicação em direção autônoma. A ideia principal é que, enquanto o veículo está viajando, o sistema de localização *online* usa parte do caminho atual, já percorrido pelo carro, para localizar o veículo no mapa, ou seja, a localização é feita comparando o BLMR, com a estrada armazenada no mapa.

Como mencionado anteriormente, a abordagem proposta baseia-se em informações visuais e *dead reckoning*. Uma vez que a estimativa de posição por *dead reckoning* é um processo incremental e, portanto, afetado por acumulação de erro, o comprimento do BLMR deve ser limitado a um valor específico, a fim de manter um nível suficiente de precisão. Se o comprimento do BLMR for muito curto, haverá pouca informação e o processo de comparação com o mapa terá sua precisão reduzida. Por outro lado, se o comprimento do BLMR for muito longo, o erro cumulativo também vai reduzir a precisão do BLMR. Neste trabalho, através de testes experimentais definiu-se 240 m como melhor valor de comprimento para o BLMR. Este valor de comprimento resultou na melhor precisão de localização de acordo com a metodologia de avaliação apresentada na Seção 5.4.1.

O algoritmo BLMR usa um longo registro das marcações de pista como informação para a localização, por isso ele é robusto contra trechos de marcações ausentes. Uma vez

que o veículo esteja precisamente localizado no mapa, há sempre um caminho de referência que orienta o veículo através das seções de marcações de pista em falta.

Portanto, o método BLMR funciona mesmo quando marcações de pista não estão presentes ou não podem ser detectadas por uma curta distância (como em cruzamentos, bifurcações e túneis, por exemplo), e nenhum trabalho manual é necessário para completar as marcações ausentes após o mapa construído. Porque o BLMR pode lidar com tais situações e não depende diretamente da existência de marcações de pista em todas as partes do caminho, acredita-se que o método proposto aqui representa uma contribuição relevante para o estado da arte atual.

5.2 Mapeamento *Online*

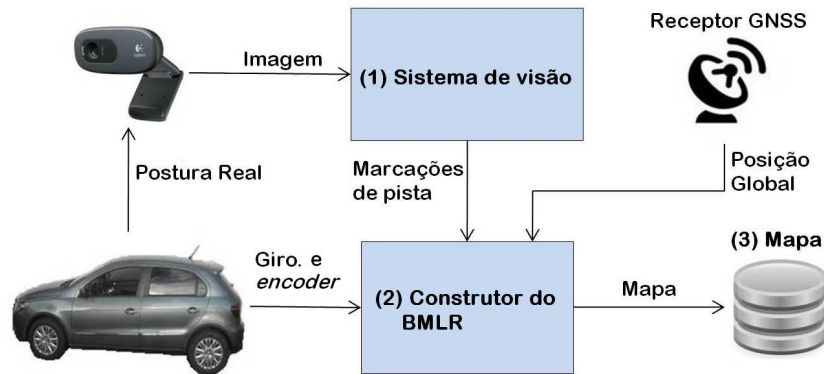
Algumas abordagens utilizadas para construção de mapas, coletam inicialmente os dados brutos da câmara e outros sensores, em uma viagem de mapeamento, e depois processam essas informações *offline*. Além disso, trabalho manual adicional pode ser necessário para se chegar à versão final do mapa.

Diferentemente destas abordagens, no nosso sistema o mapa é construído *online* durante uma viagem de mapeamento conduzida por um motorista humano ao longo do caminho desejado. As marcações de pista detectadas e os dados de *dead Reckoning* são utilizados para construir o mapa. O processo é totalmente automático e sem supervisão. Ele também não precisa de nenhum tipo de trabalho manual depois que o mapa é construído.

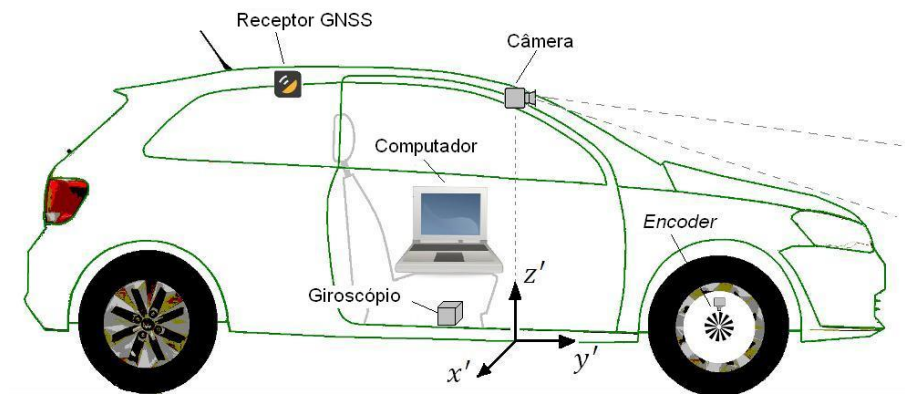
Durante a viagem de mapeamento, o computador registra a postura do veículo, estimada por *dead reckoning* 2D (x, y, θ) , e as posições de marcações de pista, detectadas pelo sistema de visão computacional. A Figura 59-a mostra a arquitetura do *software* de mapeamento que foi implementado. O Módulo 1, descrito no Capítulo 3, detecta visualmente as marcações de pista. O Módulo 2, coloca esta informação em um mapa 2D, construído a partir de *dead reckoning*. Os pontos desse mapa são vinculados à localização global, conforme detalhado na Seção 5.2.1.

O veículo protótipo usado para os experimentos é o mesmo apresentado em detalhes no Capítulo 2. A Figura 59-b destaca os sensores utilizados para construção do mapa, e o sistema de coordenadas do próprio do veículo $(x'/y'/z')$. Este arranjo de sensores também é usado para localização *online*, descrito na Seção 5.3.

Na abordagem adotada houve a preocupação em se criar um mapa no sistema de referência global (lat, lon) . Por esse motivo, a construção do mapa não exige um receptor GNSS de alta precisão ou qualquer tipo de pós-processamento. Cada mapa é construído no seu próprio sistema de referência, onde a postura inicial do veículo é



(a)



(b)

Figura 59 – Mapeamento *online*: (a) A arquitetura do *software*. (b) Montagem experimental.

considerada como a origem. O problema da direção autônoma exige principalmente que o veículo esteja corretamente posicionado sobre um caminho de referência, bem conhecido, e livre de obstáculos, não importando onde este caminho está em relação ao globo terrestre. No entanto, a posição global é registrada apenas para permitir uma estimativa grosseira da posição inicial de veículo no mapa, como será explicado na Seção 5.3.3.

5.2.1 A estrutura do mapa

Tanto o mapa quanto o BLMR respeitam a mesma estrutura de dados mostrada na Figura 60. Esta estrutura contém dois conjuntos de dados: o caminho de referência, que descreve o meio da pista e serve como caminho guia para a direção autônoma; e, naturalmente, as marcações de pista que servem para localização. A estrutura possui unidade métrica e respeita uma distância de discretização longitudinal fixa ($L_d = 1,33\text{ m}$). As marcações de pista ficam adiantadas em relação ao caminho de referência pela distância de operação do detector ($L_a = 7,2\text{ m}$). Esta padronização permite que um mapa seja usado por qualquer outro veículo, mesmo que esse use uma montagem diferente de câmera, e

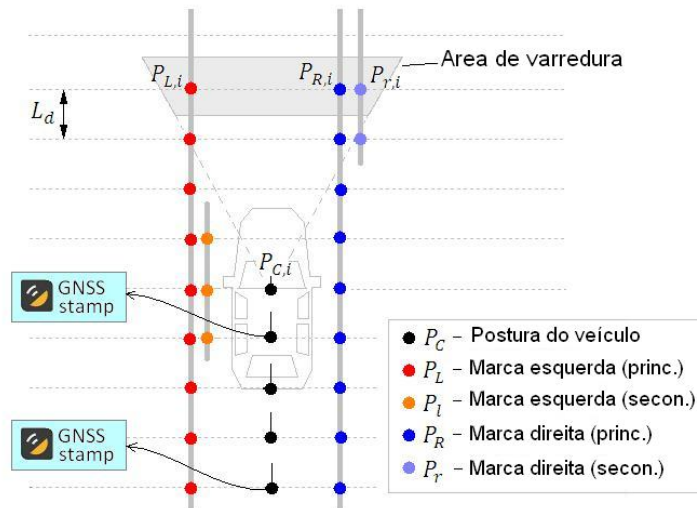


Figura 60 – Estrutura de dados do mapa e do BLMR: Caminho de referência e marcações de pista.

portanto, tenha uma distância de detecção diferente. Neste caso, o algoritmo de medição de postura (5.3.2) deve levar em conta uma eventual diferença na distância L_d e fazer a devida correção.

A discretização longitudinal simplifica e acelera a associação de pontos durante o processo de *map-matching* usado para localização. Idealmente, o valor de incremento (L_d) deveria ser tão pequeno quanto possível, mas isso produziria uma quantidade muito grande de pontos que torna o processo de localização mais pesado. Um valor de $L_d = 1,33$ m foi considerado uma boa solução de compromisso entre precisão e velocidade de execução.

Cada amostra contém informações de posição ($\mathbf{P}_c = (x_v, y_v)^T$) e orientação (*heading* θ_v) do centro do veículo (onde a câmera está montada), e a posição de até quatro marcações de pista ($\mathbf{P}_L, \mathbf{P}_l, \mathbf{P}_R, \mathbf{P}_r$). Um atributo de qualidade, gerado pelo sistema de visão durante a detecção de marcações de pista, é dada a cada ponto da pista. Este atributo é utilizado para indicar o grau de confiança da informação visual que pode variar de 0,0 a 1,0. Faixas faltando são registradas com valor de qualidade zero, e faixas bem detectadas têm qualidade próxima a 1,0, conforme explicado no Capítulo 3. O atributo de qualidade é usado no processo de *map-matching* para considerar o grau de confiabilidade da medição de postura.

Toda vez que o receptor GNSS mede sua posição atual, as informações de latitude e longitude são associadas à última amostra inserida no mapa, como uma etiqueta GNSS. Ao executar o algoritmo de localização *online*, esta associação permite ao veículo determinar sua localização inicial aproximada, nas coordenadas do mapa, com base na posição global fornecida pela leitura do GNSS.

O conjunto de n pontos $\mathbf{P}_{c,i}; i = 1..n$, chamado trilha de referência do mapa, descreve o caminho livre de obstáculos, onde o veículo passou durante a viagem de mapeamento. Esta trilha de referência pode ser usada diretamente como o caminho a ser seguido pelo veículo quando operando em modo autônomo.

5.2.2 Dead reckoning

Conforme já apresentado na Seção 4.2, o cálculo da postura do veículo é feita pela técnica de *dead reckoning*. O *dead reckoning* é calculado utilizando os sinais do *encoder* de roda e do giroscópio (*yaw-rate*). A nova postura do veículo ($\mathbf{V} = (x_v, y_v, \theta_v)^T$) em coordenadas do mapa é calculada pelas equações 4.1 a 4.3 toda vez que o veículo percorrer a distância fixa L_d .

É importante ressaltar que na abordagem proposta, o mapa e também o caminho de referência não precisam ser perfeitamente representados no sistema de referência global, ou seja, não há necessidade de serem geo-referenciados. Ao invés disso, o algoritmo constrói um mapa 2D em seu próprio sistema de referência usando *dead reckoning*. Em seguida, a localização do carro é precisamente determinada pelo algoritmo BLMR (a ser descrito na Seção 5.3) neste sistema de referência próprio. Se o mapa não tem distorções locais significativas, a direção autônoma é possível. Consideramos distorções locais significativas as que são causadas por um *bias* do giroscópio maior de 1 grau por minuto. Este valor foi determinado com base nos dados experimentais coletados, definindo-se um limiar para o erro de correspondência calculado entre trechos correspondentes. O valor escolhido foi 0,5 m.

Como o algoritmo BLMR foi projetado para ser usado em direção autônoma de acordo com um mapa pré-definido, e não precisa de uma localização global do veículo, pode-se usar o algoritmo BLMR para localizar precisamente o veículo no seu próprio mapa, mesmo usando um giroscópio de baixo custo. A análise da variação aleatória lenta do *bias* do giroscópio utilizado no carro protótipo demonstrou que, operando em temperatura controlada, a variação permanece menor que 1 grau por minuto num intervalo de uma hora, o que garante o correto funcionamento do algoritmo neste tempos.

De acordo com testes realizados, o algoritmo BLMR pode suportar um valor de *drift* de orientação de até 1 grau por minuto, mantendo uma localização precisa por até uma hora de direção.

5.2.3 Detector de Marcações de Pista

A detecção das marcações de pista a uma distância fixa na frente do veículo é feita pelo algoritmo de visão computacional herdado do programa seguidor de pista, descrito na Seção 3.2, otimizado para a operação de curto alcance. Como o tamanho da região de

processamento fica reduzido, o tempo de processamento também é menor. Os fragmentos mais distantes, acima de 8,4m, são descartados para a geração da melhor representação (Secção 3.2.4). A equação do cálculo da qualidade, expressão 3.7, foi ligeiramente alterada com a retirada do fator de penalidade de distância $(1 + |\frac{Z_{i,ini}}{0.5Z_{max}}|)$ e o parâmetro Z_{max} alterado para 8,4m. As duas melhores representações obtidas, com qualidade superior a 0,25, são usadas para avaliar a coordenada x' do centro de cada segmento na distância $y'=7,2$ m, conforme mostrado na Figura 61. Como o comprimento do segmento é curto, com apenas 2,4 m, sua curvatura é desconsiderada. Desse modo, o termo a_2 dos polinômios usados para sua representação (Seção 3.3.1) são zerados.

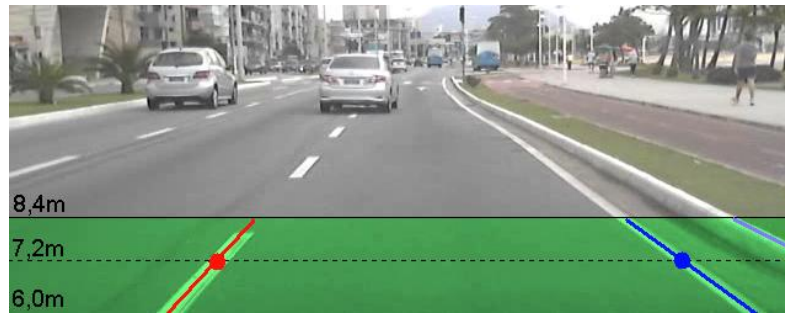


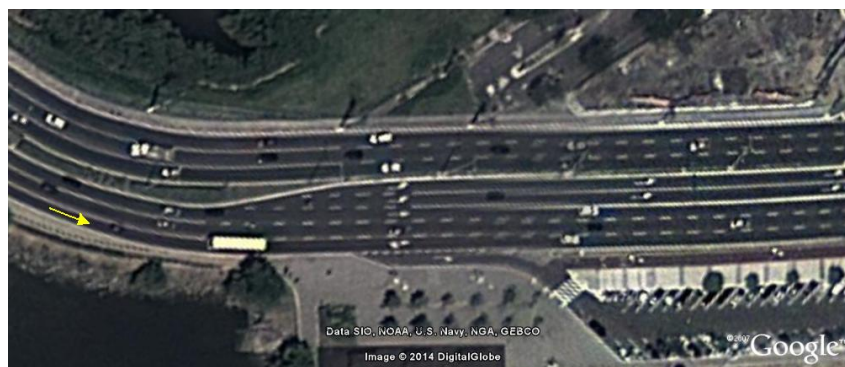
Figura 61 – O detector de marcação de pista de curto alcance operando: A informação visual entre 6,0m e 8,4m é usada para detectar os marcadores de pista.

A informação detectada no domínio da imagem da câmera é transformada para o sistema de referência do veículo (x' / y') através de transformação perspectiva inversa (*bird's eye view*) considerando a câmera na origem (Figura 60). A mudança do sistema de coordenadas do veículo (x' / y') para o sistema de coordenadas do mapa (x / y) é feita por uma operação de rotação e translação. Qualquer ponto de marcação da pista detectado no sistema de coordenadas do veículo (\mathbf{P}'_{lane}) tem a sua posição correspondente (\mathbf{P}_{lane}) no sistema de coordenadas do mapa calculado por

$$\mathbf{P}_{lane} = \mathbf{P}_c + R_{\theta_v} \cdot \mathbf{P}'_{lane}, \quad (5.1)$$

onde \mathbf{P}_c é a posição do veículo no momento da captura da imagem, R_{θ_v} é uma matriz de rotação em torno do eixo- z' em um ângulo θ_v , e θ_v é a orientação do veículo (*yaw*) no momento da captura da imagem. Figura 62 mostra um trecho de um mapa gerado de acordo com o método apresentado na presente seção.

Devido ao campo de visão ser próximo ao veículo, o mapa resultante contém apenas as marcações de pista que delimitam o corredor por onde o veículo passou, indicada pela seta amarela na Figura 62. Um mapa de múltiplas pistas poderia ser obtido por múltiplas viagens de mapeamento, cada uma em uma pista, que seriam unidas através da sobreposição das marcações de pista correspondentes.



(a)



(b)

Figura 62 – Exemplo da geração de mapa *online*: (a) foto de satélite do trecho mapeado; (b) marcações de pista resultantes.

5.3 Localização *Online*

O sistema de localização tem a função de estimar a posição do veículo (\mathbf{V}) dentro do mapa usando duas técnicas de medição absoluta de postura, uma baseada em GNSS e outra baseada no BLMR. Um filtro funde estas medições com *dead reckoning* para produzir uma estimativa de postura mais robusta e contínua.

A Figura 63 mostra a arquitetura do *software* do sistema de localização. O Módulo 1 e 2 são os mesmos usados no mapeamento (veja Seção 5.2). O Módulo 3 é o mapa, gerado em uma viagem anterior de mapeamento, e salva em arquivo. Os Módulos 4 e 5 fazem as medições de postura. O Módulo 6 estima a postura do veículo (V), filtrando e combinando as medições de postura (U or U_{gns}) com dados de *dead reckoning* (deslocamento incremental). O Módulo 7, em verde, não pertence ao sistema de localização. Ele é responsável por calcular o ângulo de posicionamento do volante (ψ), baseado na postura atual e no caminho de referência contido no mapa.

A medição de postura por GNSS é feita pelo Módulo 5 da Figura 63. Ele fica ativo somente durante a inicialização, e fornece a postura inicial aproximada do veículo no sistema de referência do mapa. A medição de posição global atual, fornecida pelo receptor GNSS, é usada para encontrar no mapa o selo GNSS mais próximo. A postura associada com este selo é definida com postura inicial ($\mathbf{V} = \mathbf{U}_{gps}$).

A medição de postura por BLMR é feita pelo Módulo 4 da Figura 63. Ele é

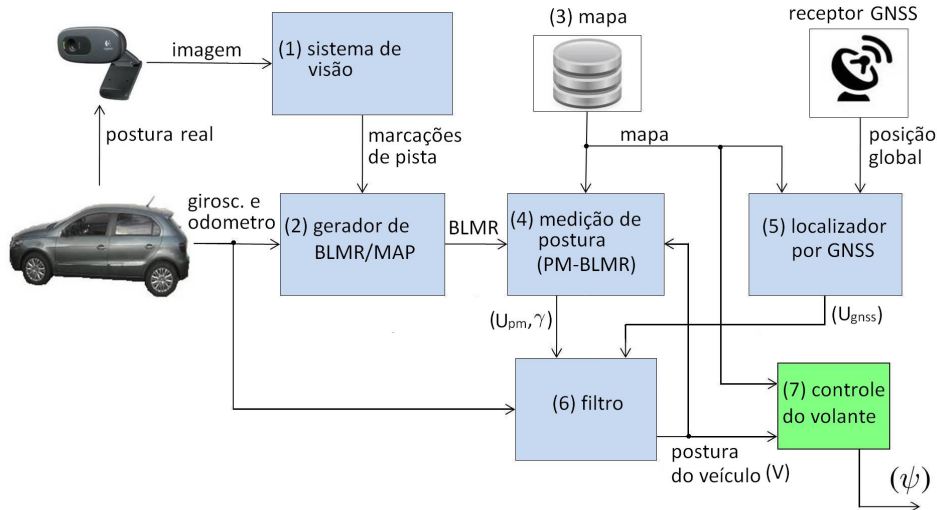


Figura 63 – A arquitetura do *software* de localização *online*.

necessário durante a operação de direção autônoma e, por causa dele, a alta precisão é obtida. O princípio, ilustrado na Figura 64, é relativamente simples: a medição de postura é feita encontrando-se a postura (\mathbf{U}_{blm}), no sistema de referência do mapa, que resulta na melhor sobreposição entre as marcações de pista armazenadas no mapa e o BLMR. A sobreposição (*map-matching*) é quantificada em termos do valor médio do desvio lateral, como explicado na Seção 5.3.2.

O filtro (Módulo 6 da Figura 63) funde as medições absolutas de postura com o *dead reckoning* para produzir uma estimativa de postura mais contínua e robusta (\mathbf{V}). Basicamente ele é um filtro passa-baixa, com diferentes ganhos para as componentes laterais, longitudinais e de orientação, e usa o odômetro e o giroscópio para calcular o deslocamento incremental (predição do estado). Note que a única informação necessária para direção autônoma é a postura do veículo (\mathbf{V}), fornecida pelo filtro, e o caminho de referência, fornecido pelo mapa.

5.3.1 O *Back Lane Marking Registry*

O *Back Lane Marking Registry* pode ser entendido como uma percepção estendida (longa) das marcações de pista, iniciando 7,2 m à frente (porque o detector usa imagens frontais) e terminando 240 m para trás. Este conceito é o núcleo de nosso algoritmo de localização, e que permite medições de postura muito precisas na referência do mapa. Esta longa e estreita (restrita à largura de uma pista) percepção é possível por causa da combinação de *dead reckoning* e um detector visual de marcações de pista de curto alcance. Em nossa abordagem, esta longa percepção (o BLMR) age como uma medida instantânea de um sensor, usada para fins de localização.

O BLMR é construído exatamente da mesma maneira que um mapa. De fato ele é um trecho de mapa. A diferença é que ele tem sua própria referência (x', y'), com a origem

fixa no centro do veículo ($\mathbf{U}' = [0, 0, 0]^T$). A outra diferença, é claro, é o número limitado de amostras. O retângulo localizado na parte superior esquerda da Figura 64 ilustra um exemplo de BLMR.

Encontrar o comprimento ideal do BLMR não é uma tarefa fácil. Se for muito longo, ele contém mais informação, o que é bom para melhorar a precisão da medição de postura. Por outro lado, o processo de *dead reckoning* acumula erro que degrada a confiabilidade dos dados mais distantes. Foi escolhido o valor de 240m para o BLMR por um processo de tentativa e erro, usando dados experimentais coletados com o veículo protótipo. Experimentou-se valores na faixa de 60m a 540m, e foi escolhido aquele que produziu a mais alta precisão, de acordo com a função de erro descrito na Seção 5.4.1. Dentro desse intervalo de valores testados a variação percentual do erro ($\epsilon_{min}/\epsilon_{max}$) foi apenas 56%, o que indica uma baixa sensibilidade do algoritmo em relação a este parâmetro.

5.3.2 Medição da Postura

A medição de postura por BLMR *map-matching* (PM-BLMR) é um algoritmo que gera medições simples de postura, encontrando a transformação de corpo rígido (rotação e translação) que melhor sobrepõe as marcações de faixa do BLMR às marcações de faixa do mapa. É um problema clássico de registro de formas, cujo princípio está ilustrado na Figura 64. Uma vez que a melhor transformação é encontrada, a medição de postura é dada pela transformação da postura atual do veículo, no sistema de referência do BLMR (\mathbf{U}'_{blm}), para o sistema de referência do mapa (\mathbf{U}_{blm}). Note que a medição de postura não é diretamente usada para direção autônoma, em vez disso ela é usada somente para atualizar a estimativa de postura do veículo, calculada pelo filtro (Módulo 6 da Figura 63). Note também que o caminho de referência não participa do cálculo do *map-matching*. A razão é porque o caminho exato percorrido pelo veículo durante o mapeamento manual e a direção autônoma podem ser diferentes.

A melhor transformação entre dois conjuntos de pontos 2D pode ser encontrada diretamente por qualquer algoritmo de registro de forma. *Iterative closest point* (ICP) (BESL; MCKAY, 1992) é um algoritmo poderoso e consolidado capaz de realizar o registro de formas 2D ou 3D de uma variedade de dados geométricos (conjunto de pontos, segmentos de linhas, curvas implícitas), sem conhecimento a priori da correspondência de dados. Atualmente existem muitas variantes do ICP básico, algumas adotam estratégias particulares como a seleção dos pontos da fonte, ponderação dos pares de correspondência e rejeição de pontos fora da curva. ICP requer uma transformação inicial que deve ser escolhida cuidadosamente para garantir que a solução não ficará presa em um mínimo local. No caso aqui abordado (BLMR e mapa), como a correspondência entre pontos do BLMR e do mapa é quase conhecida, foi desenvolvido um algoritmo de *map-matching* simples e leve, otimizado para as particularidades da estrutura de dados.

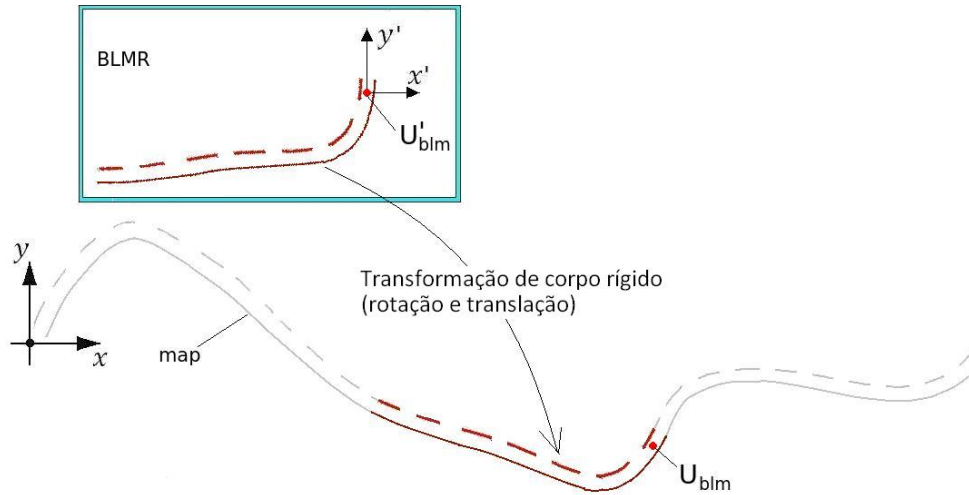


Figura 64 – Estratégia de medição de postura: encontrar a transformação de corpo rígido que resulta na melhor sobreposição entre as marcações de pista do BLMR e do mapa.

No caso do BLMR e do mapa, como eles são gerados dentro do mesmo espaçamento longitudinal, se for estabelecida a correspondência para o primeiro ponto (ponto cabeça - U_{blm}), automaticamente a correspondência será estabelecida para todos os outros pontos (ver Figura 60). Uma vez estabelecida a correspondência de pontos, a melhor transformação de corpo rígido pode ser encontrada de modo direto. Na abordagem aqui proposta, o processo de estabelecimento da correspondência para o ponto cabeça é feito por tentativa e verificação de erro, procurando a correspondência que resulta no menor erro de *matching*. Como a estimativa atual da postura do veículo (V) está disponível pelo módulo de filtragem, foram utilizados os 7 pontos do mapa mais próximos (Figura 65) como pontos cabeça candidatos. Note que as marcações de pista foram omitidas apenas para simplificar o desenho. Note também que, como os pontos cabeça candidatos são longitudinalmente distribuídos, encontrar o melhor candidato corresponde a resolver a componente longitudinal da medição de postura.

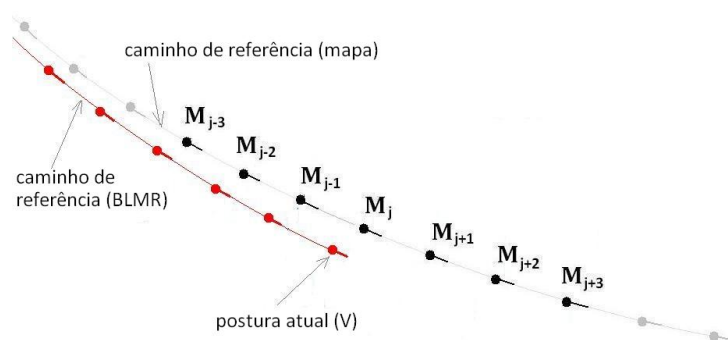


Figura 65 – Seleção dos *Head points* candidatos: utilizar o mais próximos a posição atual do veículo. As marcações de pista não estão mostradas aqui para simplificar o desenho.

Como já mencionado antes, se a correspondência entre dois conjuntos de pontos for conhecida, a melhor transformação de corpo rígido que minimiza o erro quadrático pode ser encontrada. Em (HORN, 1987) é apresentada uma forma fechada de solução baseada em quaternions unitários. A translação é calculada através da diferença dos centros de massa dos dois conjuntos de pontos, e a rotação calculada através da decomposição de valor singular da matriz de covariância.

No caso desta tese, além da uniformidade espacial, o conjunto de dados possui outras particularidades: (i) os dados são geometricamente alongados, ou seja se colocados num retângulo, a relação largura/comprimento é bem alta. (ii) cada ponto de marcação de pista possui um fator de confiabilidade (fator de qualidade - ver Seção 3.2.4.2) atribuído pelo detector visual de marcações. (iii) devido ao cálculo de *dead reckoning* usado para geração do BLMR, os pontos mais próximos ao ponto cabeça têm confiabilidade maior do que os pontos distantes.

A fim de manter todo o algoritmo de localização leve, foi proposto um algoritmo específico para encontrar a transformação de corpo rígido sem resolver a custosa decomposição de valor singular. O algoritmo é dividido em dois passos. Primeiro, a translação e rotação são calculadas usando dados do caminho de referência. Segundo, é aplicado um ajuste fino lateral usando os dados das marcações de pista. Depois disso, o erro de correspondência entre as marcações de pista é calculado, considerando o fator de qualidade das marcações e um fator de confiabilidade devido à incerteza do *dead reckoning*. Finalmente, é feita a análise do gráfico do erro de correspondência para gerar o parâmetro de confiabilidade da componente longitudinal da medição de postura.

As etapas da nossa transformação de corpo rígido são descritas nas subseções seguintes.

5.3.2.1 Translação e Rotação

A translação e rotação são executadas diretamente a partir de dados recuperados exclusivamente do caminho de referência do BLMR e do mapa. Este processo corresponde à primeira aproximação da transformação de corpo rígido. Primeiro, o BLMR é translado de modo que o ponto \mathbf{U}' coincida com ponto cabeça, então a rotação é aplicada de modo que o vetor \mathbf{MA} tenha a mesma direção que \mathbf{MB} (ver Figura 66).

O ângulo de rotação β é dado por

$$\beta = \arcsin \left(\frac{|\vec{MA} \times \vec{MB}|}{|\vec{MA}| |\vec{MB}|} \right). \quad (5.2)$$

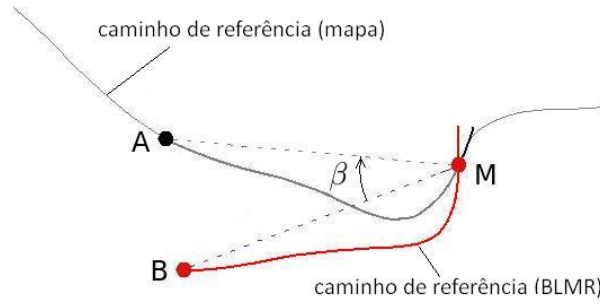


Figura 66 – Ajuste de orientação: cálculo do ângulo de rotação β necessário para alinhar os vetores MA e MB.

5.3.2.2 Ajuste Fino Lateral

O ajuste fino lateral corresponde à segunda aproximação da transformada de corpo rígido. Este processo mede a informação mais crítica para direção autônoma, o deslocamento lateral. O ajuste lateral fino é feito de acordo com o esquema ilustrado na Figura 67-a. O objetivo é calcular a translação lateral necessária que resulta na menor diferença lateral entre as marcações de pista do mapa e as do BLMR.

Este ajuste é calculado usando os 8 pontos mais recentes de marcação de pista de modo a obter uma resposta rápida. No exemplo da Figura 67-a, a marcação esquerda é intermitente e a direita é contínua. Quando as quatro marcações de pista estão disponíveis (R, r, L, l), todas elas são usadas no cálculo, ponderadas pelo fator de qualidade (Q), de acordo com

$$\delta_c = \frac{\sum_{j=1}^8 \sum_{i=[R,r,L,l]} Q_{map,i,j} Q_{blm,i,j} \delta_{i,j}}{\sum_{j=1}^8 \sum_{i=[R,r,L,l]} Q_{map,i,j} Q_{blm,i,j}}. \quad (5.3)$$

Finalmente, o deslocamento lateral δ_c é usado para transladar lateralmente todos os dados do BLMR (marcações de pista e caminho de referência), obtendo o resultado final mostrado na Figura 67-b.

Depois de obter a transformação de corpo rígido associada a cada ponto cabeça candidato, o erro de correspondência é calculado para todos os pontos. A Figura 68 ilustra o erro de correspondência normalizado calculado para três trechos dos locais de teste. A transformação final de corpo rígido considerada é obviamente aquela com o menor valor de erro.

O erro de correspondência é calculado como o valor médio do desvio lateral (valor absoluto) entre amostras correspondentes das marcações de pista do BLMR e do mapa. O fator de ponderação considera maior incerteza das amostras distantes, causado pelo *dead reckoning*. Ele é calculado por $G_j = \exp(-(\frac{j}{n})^2)$, onde n é o número de amostras do

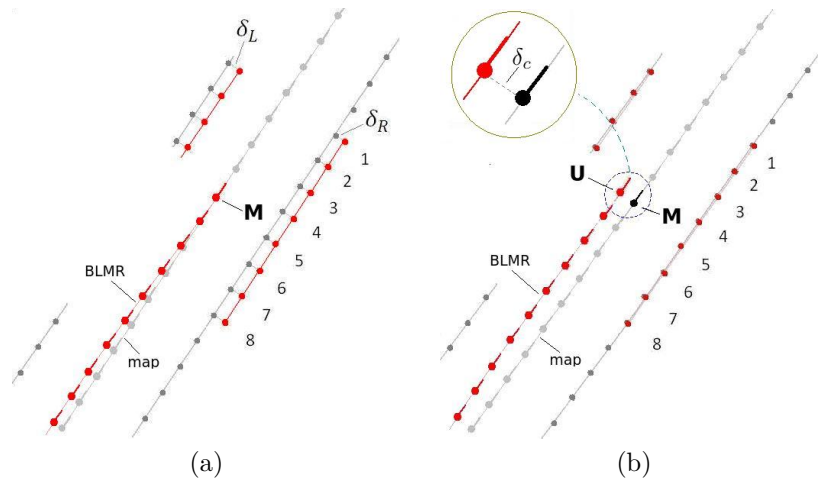


Figura 67 – Ajuste lateral: (a) cálculo do desvio lateral; (b) após efetuado o ajuste.

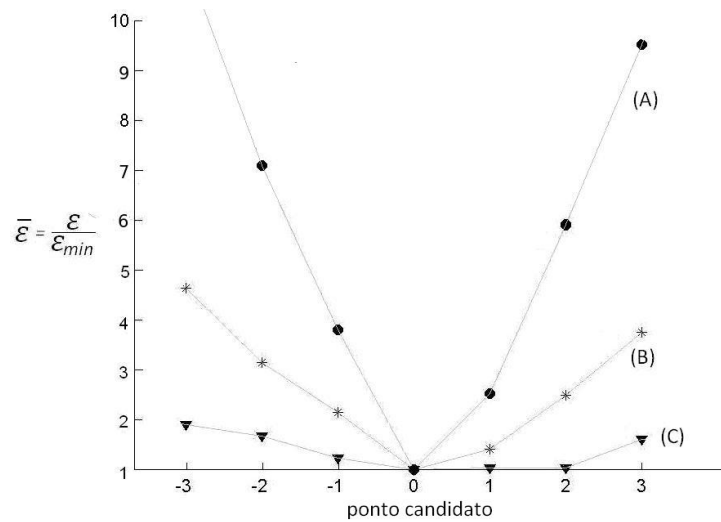


Figura 68 – Erro de correspondência normalizado para diferentes secções de nosso trajeto de teste: (A) o BLMR contém uma curva acentuada; (B) o BLMR contém uma curva média; (C) o BLMR contém um trecho reto.

BLMR ($n = 180$). O erro de correspondência é calculado por

$$\varepsilon = \frac{\sum_{j=1}^n G_j \sum_{i=[R,r,L,l]} Q_{map,i,j} Q_{blm,i,j} |\delta_{i,j}|}{\sum_{j=1}^n G_j \sum_{i=[R,r,L,l]} Q_{map,i,j} Q_{blm,i,j}}, \quad (5.4)$$

onde $Q_{map,i}$ é a qualidade da i -ésima amostra do mapa; $Q_{blm,i}$ é a qualidade da i -ésima amostra do BLMR; $|\delta_{i,j}|$ é o desvio lateral absoluto.

Analisando a inclinação do gráfico de erro de correspondência (Figura 68) pode-se extrair informações em relação à forma do BLMR e como ela afeta a confiabilidade da escolha do melhor ponto cabeça (discriminação do ponto de menor erro). O gráfico A corresponde a um trecho de estrada que contém uma curva acentuada, de modo que um pequeno deslocamento longitudinal do BLMR sobre o mapa (variação do ponto cabeça candidato) causa muita variação no erro de correspondência (altamente discriminativo).

Neste caso é muito fácil identificar o mínimo do gráfico, e portanto obtém-se uma alta confiabilidade na medição longitudinal da postura. Por outro lado, o gráfico C corresponde a um trecho aproximadamente reto. Neste caso, deslocamentos longitudinais causam pouca variação no erro de correspondência, que reflete em uma baixa confiabilidade da medição longitudinal. O gráfico B ilustra um caso intermediário.

A fim de modelar esta incerteza de modo que ela possa ser usada pelo filtro, foi proposta a relação matemática ilustrada na Figura 69, onde (γ) é o parâmetro de confiabilidade e $(max(\bar{\epsilon}))$ é o máximo valor do erro de correspondência normalizado. Foram ainda proposto dois limiares para $max(\bar{\epsilon})$. Quando ele for menor que 2, considera-se que o BLMR é muito reto, e a portanto o parâmetro de confiabilidade é nulo. Quando ele for maior que 6, considera-se que o BLMR contém curvas suficientes de modo que o parâmetro de confiabilidade é máximo.

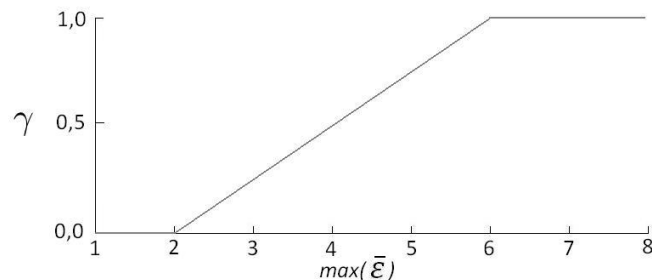


Figura 69 – Parâmetro de confiabilidade para a componente longitudinal da medição de postura.

5.3.3 Filtro

O filtro (Módulo 6 da Figura 63) calcula uma estimativa contínua e suave da postura do veículo. O filtro reduz o ruído (erro aleatório) da medição de postura e a funde com os dados do giroscópio e odômetro. Diferente de outras abordagens usadas em localização de robos móveis, foi usado um filtro passa baixa de primeira ordem com ganhos independentes para cada componente (k_c -lateral, k_l -longitudinal e k_h -heading). Por causa das particularidades do algoritmo de medição de postura, optou-se por fixar os ganhos laterais e de orientação fixos e, o ganho longitudinal ajustável de acordo com o parâmetro de confiabilidade γ .

Como discutido na secção anterior, a componente longitudinal da medição de postura (escolha do melhor ponto candidato) é um processo em que o erro depende da forma do BLMR. Para considerar o parâmetro de confiabilidade no processo de filtragem, foi usado o parâmetro (γ) Este parâmetro atua diretamente no ganho de atualização longi-

tudinal através da expressão

$$k_l = \gamma k_{lmax}, \quad (5.5)$$

onde k_{lmax} é o valor máximo do ganho longitudinal.

Quando o BLMR é quase uma linha reta, o parâmetro de confiabilidade γ é nulo, portando atualizações longitudinais de localização não são feitas. Por outro lado, quando o BLMR contém curvas acentuadas, o parâmetro de confiabilidade γ será aproximadamente um, portanto a atualização será feita com seu valor máximo de ganho (k_{lmax}). Além disso a componente longitudinal da medição de postura está restrita ao espaçamento uniforme do mapa (veja Figuras 60 e 65), o que adiciona ruído de discretização de alta frequência.

Para remover adequadamente o ruído de alta frequência (frequência espacial) contido na componente longitudinal da medição de postura, e produzir um estimativa mais suave, foi definido um valor do ganho máximo (k_{lmax}) baixo, para uma ação de filtragem lenta. A ação lenta de correção sobre a componente longitudinal não atrapalha a direção autônoma porque esta componente é menos crítica do que a componente lateral. Além disso, como o carro está andando em uma estrada pavimentada (onde o escorregamento dos pneus é mínimo), o erro acumulativo de localização na direção longitudinal cresce lentamente de modo que a ação lenta de correção é suficiente para manter o erro produzido por escorregamento limitado.

A estimativa lateral da postura é a informação mais crítica para manter um veículo dirigindo automaticamente no centro de uma pista. Portanto, para essa componente uma resposta rápida do filtro é necessária. Além disso, a componente lateral da medição de postura, feita pelo detector de marcações de pista de curto alcance, produz valores pequenos de erro. Por causa disso, definiu-se o valor do ganho de atualização lateral mais alto do que para a componente longitudinal. Em relação à componente de orientação (ângulo yaw), as medições de orientação são muito estáveis e, portanto, ação de filtragem rápida foi empregada assim como no caso da componente lateral.

Toda vez que uma nova amostra é inserida no BLMR a atualização da localização do veículo é feita por

$$\mathbf{V}_i = \mathbf{V}_{i-1} + R_\theta K R_\theta^T (\mathbf{U}_i - \mathbf{V}_{i-1}), \quad (5.6)$$

$$R_\theta = \begin{bmatrix} \cos V_\theta & -\sin V_\theta & 0 \\ \sin V_\theta & \cos V_\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (5.7)$$

$$K = \begin{bmatrix} k_c & 0 & 0 \\ 0 & k_l & 0 \\ 0 & 0 & k_h \end{bmatrix}, \quad (5.8)$$

onde K é a matriz de ganho para as componentes separadas, e R_θ é a matriz que opera uma rotação 2D somente para as componentes x-y.

Os valores iniciais de ganho foram definidos para $k_c = 0.1$, $k_{lmax} = 0.01$, e $k_h = 0.1$. O ajuste fino desses valores foi feito experimentalmente usando dados coletados pelo veículo protótipo. Os melhores resultados foram alcançados com $k_c = 0.25$, $k_{lmax} = 0.008$, e $k_h = 0.25$. Apesar de termos variado bem os parâmetros do filtro, por exemplo, o ganho transversal de 0.1 para 0.25, esta variação teve pouco efeito na precisão do algoritmo medido pelo critério de erro adotado (ver Seção 5.4.1), o que indica uma baixa sensibilidade.

Antes do filtro iniciar sua operação usando as medições de postura precisas fornecidas pelo Módulo 4 (PM-BLMR), ele deve passar por três diferentes modos de operação. Primeiro, a localização inicial aproximada é fornecida pelo receptor de GNSS. Segundo, o sistema deve esperar até que o BLMR esteja preenchido com um número mínimo de amostras para que a PM-BLMR possa operar. Finalmente, quando o BLMR está pronto, o filtro opera em modo preciso. Independente do modo de filtragem *dead reckoning* é calculada para fazer a predição da postura.

5.3.3.1 Modo 1 - Localização Desconhecida

O primeiro modo é ativo quando o filtro não tem nenhuma estimativa de sua localização, ou seja, quando o sistema é ativado. Quando o receptor GNSS recebe sua primeira localização o módulo de localização por GNSS (Módulo 5 da Figura 63) encontra a posição mais próxima (\mathbf{U}_{gns}) no mapa, usando a informação dos selos GNSS (Figure 60). O filtro considera este ponto como a primeira estimativa de localização ($\mathbf{V}_0 = \mathbf{U}_{gns}$). Depois disso, o filtro comuta para o Modo 2.

5.3.3.2 Modo 2 - Localização Aproximada

Neste modo o filtro espera o BLMR alcançar seu comprimento mínimo operacional, que foi definido como metade do comprimento máximo. Quando isso ocorre, o módulo PM-BLMR entra em operação e, se o erro mínimo de correspondência for menor que o limiar de (0.5 m), o filtro redefine a localização atual como esta medição ($\mathbf{V}_0 = \mathbf{U}_{blm}$), e comuta para o Modo 3. Enquanto estiver no Modo 2 o filtro continua a usar as medições de GNSS para atualizar seu estado.

5.3.3.3 Modo 3 - Localização Precisa

O Modo 3 é ativado depois que o BLMR entra em operação, como explicado anteriormente. Neste modo o sistema de localização não usa mais as medições de GNSS. A localização é atualizada exclusivamente a partir das medições de postura feitas pelo módulo PM-BLMR. Somente nesse módulo o veículo é capaz de fazer direção autônoma.

5.4 Testes e Resultados

A avaliação do sistema de localização foi feita em duas partes. A primeira parte dos testes foi feita no Brasil com o objetivo de analisar a precisão da localização alcançada com o método proposto. A segunda parte dos testes foi feita na Itália, com o objetivo de verificar o desempenho de direção autônoma baseado no método de localização. No final desta seção também é apresentado o desempenho computacional do algoritmo.

5.4.1 Precisão de localização

O cálculo da precisão da localização de veículos operando em condições reais de tráfego é uma questão complicada devido à dificuldade prática de se obter o *ground truth*. Sistemas normais de GNSS não podem ser usados como referência *ground truth* porque seu erro é muito maior do que o erro do sistema investigado. Sistemas GNSS de alta precisão, como GNSS com correção RTK, poderiam ser utilizados em certas circunstâncias.

O algoritmo de localização proposto nesta tese foi projetado principalmente para fins de direção autônoma, uma das aplicações que requer o mais alto nível de precisão e confiabilidade. Por esta razão, a metodologia usada para calcular a precisão preocupou-se em analisar a informação que afeta diretamente esta tarefa.

Uma maneira simples de realizar direção autônoma é usar o desvio lateral de um ponto alvo (ξ), em relação ao eixo longitudinal do veículo, para calcular o ângulo de acionamento do volante. A ideia está ilustrada na Figura 70. O raio de curvatura, e conseqüentemente o ângulo do volante (ψ), necessário para que o veículo alcance o ponto alvo, é calculado pelas equações 5.9 e 5.10, onde f é uma função dos parâmetros cinemáticos do veículo.

$$R(\xi, L_\xi) = \frac{\xi^2 + L_\xi^2}{2\xi} \quad (5.9)$$

$$\psi(\xi, L_\xi) = f(R) \quad (5.10)$$

Para uma dada distância L_ξ , o desvio lateral ξ pode ser obtido a partir do estado atual do veículo (\mathbf{V}), fornecido pelo módulo de filtragem, e o caminho de referência do

mapa, fornecido pelo arquivo do mapa. Note que a única variável suscetível a incertezas, que pode causar erro aleatório no cálculo do ângulo do volante (ψ), é a variável ξ . Por esta razão, foi proposto um método para avaliar a precisão da localização analisando o erro da variável ξ (erro lateral do ponto alvo - ELPA).

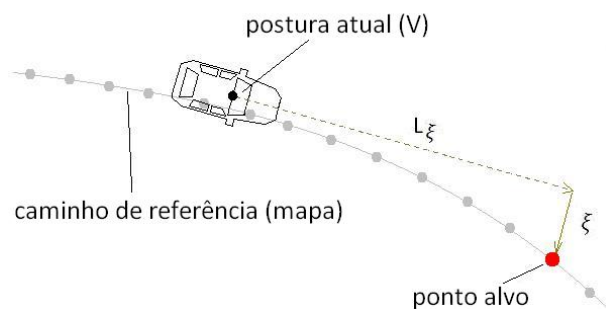


Figura 70 – Direção autônoma baseada no seguimento de um ponto alvo no caminho de referência.

A distância L_ξ , usada na análise, é fixa e foi escolhida considerando a velocidade de operação de 60 km/h e o tempo de alcance do alvo de 1,5s. O ponto alvo é tomado diretamente do caminho de referência do mapa, escolhendo aquele cuja distância é mais próxima de L_ξ .

Os dados usados para o cálculo de precisão foram coletados em dois locais com diferentes características geométricas (ver Figura 71-a-b). O total de dados coletados nestas estradas, em 21 viagens, correspondem a 177 km.

O primeiro local escolhido fica localizado na cidade de Vitória, ES, Brasil, (S20.2971567, W40.2920133) com um comprimento total de 5,1 km. É uma avenida larga com 3 pistas em cada direção, localizada à beira mar, em terreno plano com poucas e suaves curvas. A viagem de mapeamento e as duas primeiras viagens de localização foram feitas no mesmo dia, em condições similares de iluminação (um dia nublado). Duas outras viagens de localização foram feitas 15 dias depois, também em um dia nublado. Finalmente, as duas últimas viagens de localização foram feitas um mês depois, em um dia claro de céu azul.

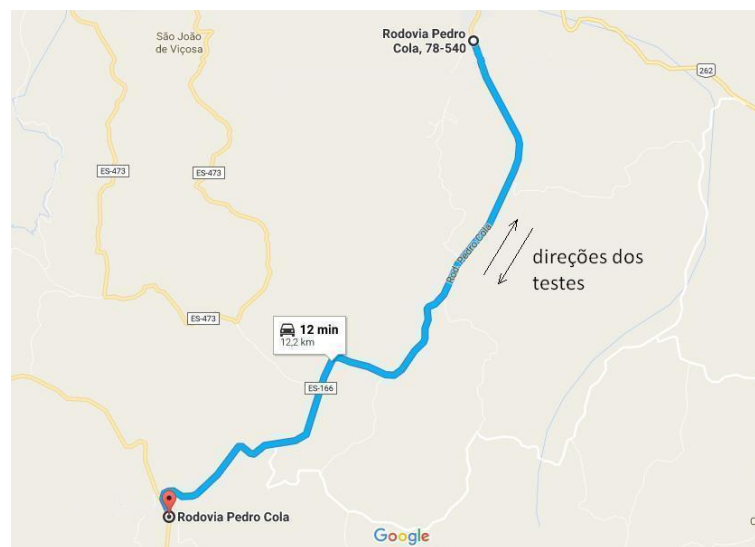
O segundo local fica localizado na cidade de Venda Nova do Imigrante, ES, Brasil (S20.348685, W41.1297919) com um comprimento total de 12,2 km. Esta estrada apresenta uma geometria muito mais complexa do que a primeira. É uma rodovia rural com mais curvas, curvas mais acentuadas e curvas em ladeira. Os testes nesta estrada foram realizados em ambas as direções. As primeiras 3 viagens de localização foram feitas um dia após a viagem de mapeamento, em um dia nublado. Duas outras viagens foram feitas dois dias após, em um dia chuvoso.

O protocolo experimental adotado foi:

- Primeiro, o veículo protótipo foi dirigido manualmente pelo caminho desejado, para



(a) Local 1 (6 x 5.1km)



(b) Local 2 (6 x 2 x 12.2km)

Figura 71 – Estradas escolhidas para a avaliação experimental da precisão.

construir o mapa, em velocidades entre 50km/h e 60km/h.

- Exatamente como antes, outras viagens foram feitas, simulando as viagens em direção autônoma. Os dados foram salvos em arquivo para análise posterior.
- O programa de localização processou *offline* os dados gravados e calculou a precisão, representada na forma de gráficos de violino do ELPA.

5.4.1.1 Metodologia para o *Ground truth*

Para calcular o erro de ξ , o ELPA, é necessário compará-lo com um valor de referência correto (*ground truth*). No caso deste trabalho, o erro é calculado em relação a uma referência quase correta (RQC). O cálculo foi feito em pós-processamento, usando os dados de *dead reckoning* salvos nas viagens de coleta de dados. Esta informação foi usada para reconstruir a posição das marcações de pista até 25 m a frente do carro, considerando isso como RQC (Figure 72). Esta informação corresponde ao *forward lane*

marking registry (FLMR). Baseado na análise do sistema *dead reckoning*, o erro lateral introduzido por ele, na faixa de 25 m, é somente 1,2 cm (0,05%). Outra fonte de erro que afeta ξ é a variação aleatória do ângulo de *pitch* do veículo devido ao deslocamento em superfícies imperfeitas. Baseado nos parâmetros da transformação perspectiva inversa, e na medição da variação de *pitch* tomada com o eixo- x' do giroscópio, estimou-se que esta parcela de erro seja em torno de 2,0 cm. Desse modo, foi considerado que o RQC tem um erro próprio de 2,33 cm em relação à referência exata, o que é perfeitamente aceitável no caso em questão.

O ELPA foi obtido diretamente a partir da diferença lateral entre o RQC e as marcações de pista tomadas a uma distância (L_ξ).

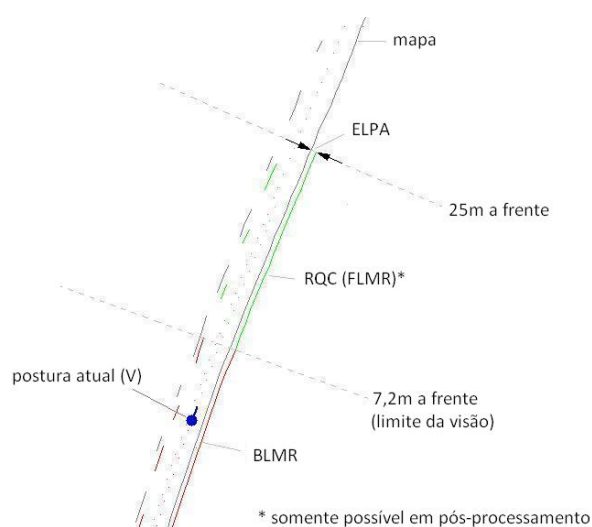


Figura 72 – Registro de *dead reckoning* usado para estimativa do erro lateral do ponto alvo (δ_ξ).

5.4.1.2 Resultados

A Figura 73 mostra os gráficos de violino (histograma vertical) do valor absoluto do erro lateral do ponto alvo (ELPA), obtido em cada local de teste. Ele representa a distribuição de erro em nossa variável de direção autônoma ξ .

Para o local 1 foi considerada a velocidade de operação de 60km/h e o ponto alvo 25 m à frente. O valor médio obtido foi de 0,056 m e permanece abaixo de 0,290 m em 99,9% do trajeto. Os valores de erro angular correspondente, calculados de acordo com $\frac{\delta_\xi}{L_\xi}$, foram $0,13^\circ$ e $0,66^\circ$ respectivamente. Para o local 2 foi considerado a velocidade de operação de 80km/h, e o ponto alvo 33 m à frente. O erro médio obtido foi 0.064 m e permanece abaixo de 0,410 m em 99,9% do trajeto. Os valores de erro angular correspondente foram $0,11^\circ$ e $0,71^\circ$ respectivamente. Os baixos valores de ELPA obtidos após o processamento de 177 km de dados reais, obtidos em dois locais diferentes, foram um forte indicador que o sistema seria capaz de realizar direção autônoma com sucesso. Isso

foi um grande motivador para a realização dos testes de direção autônoma, apresentados na próxima seção.

Os valores médios de ELPA obtidos nos dois locais de teste foram bem próximos, com o ELPA do local 2 sendo ligeiramente maior que o do local 1. Esta diferença era esperada porque o ponto alvo para o local 2 fica mais distante do carro. Mas, note que para o erro angular ocorreu o oposto. É importante mencionar que a distribuição do erro vai variar dependendo da forma do caminho mapeado (características geométricas). O erro lateral pode ser causado por: (a) o erro do detector das marcações de pista; ou (b) o erro de localização.

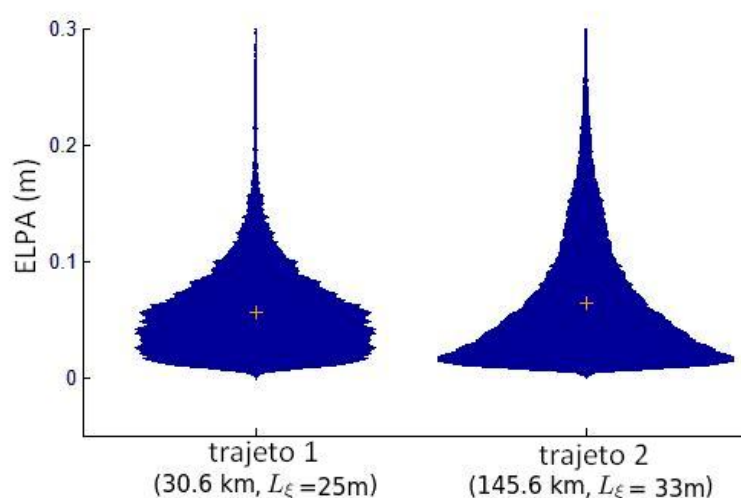


Figura 73 – Precisão de localização: Gráficos violino (histogramas verticais) do erro lateral do ponto alvo (ELPA) das duas estradas testadas. A cruz corresponde ao valor médio.

A componente de erro devido ao detector das marcações de pista é causada pela variação do ângulo de *pitch* (um movimento natural quando o carro se move), deficiências aleatórias na pintura das faixas e ruído interno do algoritmo de detecção. Ele ocorre uniformemente em toda a extensão do caminho e, portanto, age como um ruído de fundo. Isto explica porque a maior parte da energia do histograma da Figura 73 está concentrada na faixa de 0 - 10 cm.

Por outro lado, a componente de erro devido à localização ocorre principalmente nas entradas de curvas. Por exemplo, considere que o veículo esteja no início de uma curva após um longo trecho reto. Devido à impossibilidade de realizar medições de localização longitudinal nos trechos retos, erros de localização longitudinal podem ocorrer. Suponha que o veículo esteja adiantado no mapa, o que significa neste caso que, ele "acredita" que a curva começará antes do ponto real. Isso causará um alto valor de ELPA transitório. Quando o carro iniciar a execução da curva, a informação de curvatura entrará no BLMR e permitirá a correção da localização longitudinal. Mas, durante este tempo transitório, haverá um alto valor de ELPA. Isso explica a ocorrência dos mais altos (mas pouco

frequentes) valores de ELPA no histograma ($>0,1$ m). Note que quanto mais acentuada for a curva, maior será o ELPA.

Finalmente, como estradas diferentes possuem quantidades e intensidades diferentes de curvas, a distribuição dessa parcela do ELPA vai variar em cada caso.

A Tabela 5 apresenta um resumo comparativo dos valores de precisão de localização obtidos com sistemas de localização baseado em visão. Note que as abordagens diferentes usam metodologias diferentes para calcular o erro de localização, portanto a comparação numérica deve ser feita com cuidado. Por exemplo, alguns trabalhos informam todos os componentes do erro (lateral, longitudinal e total), outros somente o erro total.

A metodologia de erro usada nesse trabalho (ELPA) combina as componentes longitudinal, lateral e de orientação de modo que elas afetam diretamente a precisão da direção autônoma. Por outro lado, alguns trabalhos apresentados na tabela apresentam somente o erro médio, enquanto outros, o médio e máximo e, outros um gráfico do erro em função do tempo (o qual foi usado para estimar o máximo valor de erro quando não foi declarado no texto do artigo). Neste trabalho de tese foi apresentado o erro médio, o valor de percentil 99,9% e o histograma (gráfico de violino), que permite uma visualização completa.

Note que o erro lateral é mais crítico do que o longitudinal para fins de direção autônoma. Note também que um dado sistema de localização pode apresentar um valor de erro médio baixo, mas se, mesmo por um período curto de tempo, o erro subir acima de um máximo de segurança, o sistema não é capaz de realizar direção autônoma. Uma maneira mais eficiente de um sistema de localização de alta precisão demonstrar sua capacidade é através da realização de viagens de direção autônoma em complexas e variadas rodovias reais. A realização segura de tais viagens é uma tarefa cara, complexa e que demanda muito tempo, assumida apenas por uma pequena parte dos trabalhos relacionados.

Todos os dados experimentais coletados pelo veículo de teste no Brasil (mapas de acordo com a estrutura descrita na Seção 5.2) estão disponíveis em <http://bit.ly/1MH1Pqb>. Além disso foi incluído um *script* de MATLAB para acessar os dados originais (ELPA) de modo que outra análise estatística possa ser feita por quem estiver interessado.

5.4.2 Direção Autônoma

Uma vez que o sistema de localização demonstrou boa precisão, o equipamento foi preparado para realização dos testes de direção autônoma. Os testes de direção autônoma foram realizados na Itália, com o apoio do laboratório VisLab, que pertencia a *Università degli studi di Parma*. O laboratório VisLab é reconhecido mundialmente por seus projetos e produtos na área de visão computacional e veículos autônomos. Entre suas contribuições

Tabela 5 – Comparação de resultados reportados por trabalhos similares de localização por visão em veículos automotivos ($a, b, c, d, e, f =$ (HATA; WOLF, 2014; ZIEGLER et al., 2014; BADINO; HUBER; KANADE, 2011; LATEGAHN; STILLER, 2012; CUI; XUE; ZHENG, 2016; LYRIO et al., 2015)).

	Nossa abordagem	a	b	c	d	e	f
Método de Localização	BLMR <i>map-matching</i> e filtragem com ganho variável	monte carlo (MCL)	três métodos são usados: (i) minimização de uma função de erro específica (para PFL); (ii) filtro de Kalman (para LFL) e; (iii) filtro de Kalman sigma-point (fusão de PFL e LFL).	filtro de bayes discreto	3D features map-matching	iterative closet point (ICP)	virtual generalizing random access memory weightless neural networks (VG-RAM)
Localização online	sim @ 130Hz	não declarado	sim @ 10Hz	não	sim @ 17Hz	sim @ 20Hz	não
Ground truth (dispositivo ou método)	relativo ao mapa - pós processamento de <i>dead reckoning</i>	Xsens MTI-G GPS	relativo ao mapa	GPS padrão	high precision DGPS	marcações de pista rotuladas manualmente	repetibilidade (comparação entre duas viagens)
Ground truth (precisão)	0.02m (por causa do <i>dead reckoning</i>)	não reportado	0	reportado 3.0 m	não reportado	não reportado	0
Erro (método)	lateral of target point (LETG)	lateral/longitudinal /total	lateral	total	total	lateral por faixas de distância: (faixa considerada aqui é de 20m a 30m)	total
Erro (médio)	0.063 m	0.31m / 1.20m / 1.25m	0.100m	2.70m	0.340m	3 pixels (0.150m)	2.61m
Erro (max em 99.9%)	0.401m	-	0.200m	não reportado	não reportado	não reportado	não reportado
Error (max in 100%)	0.540 m	1.50m / 0.75m / 1.40m	não reportado	18.0 m	não reportado	não reportado	6.50m
Validação com direção autônoma	sim	não	sim	não	não	sim	não
Comprimento total dos testes	6 x 5.1km + 12 x 12.2 km = 177km	2 x 0.82km = 1.64km	103km	1 x 8.8km = 8.8km	2 x 1km = 2km	testado em FC2011 contest (competição de carros autônomos na china)	2 x 3.57km = 7.14km

na área de veículos autônomos pode-se destacar: o desenvolvimento do sistema de visão do veículo TERRAMAX (BROGGI et al., 2010), qualificado nas competições DARPA 2005 e 2007; a realização do projeto VIAC (BERTOZZI et al., 2011), que foi um desafio intercontinental de direção autônoma com 13000km; e o projeto PROUD (VISLAB, 2013), que foi uma exibição de direção autônoma na cidade de Parma sem nenhuma intervenção humana. Em meados de 2015, o laboratório foi vendido para empresa americana Umbarella, especializada na fabricação de semicondutores para processamento de vídeo.

5.4.2.1 Preparação dos Testes

O plano de trabalho traçado e executado foi o seguinte: adaptação do algoritmo ao ambiente de programação da Itália; adaptação de veículo de testes (construção dos sensores), testes de desempenho do sistema *dead reckoning* construído; escolha do trajeto de testes; realização de viagens de mapeamento; avaliação da precisão de localização; e finalmente a realização das viagens de direção autônoma.

A primeira fase do trabalho foi a adaptação dos códigos escritos no Brasil, em C++ e ambiente Windows, para a plataforma de trabalho do VisLab. O laboratório trabalha em ambiente Linux e usa uma plataforma para desenvolvimento de aplicações de visão para veículos inteligentes, chamada GOLD (BERTOZZI et al., 2008). A plataforma GOLD, desenvolvida pelo próprio laboratório, oferece diversas funções de processamento de imagem, visão computacional, e interface de sensores. A plataforma também possui

ferramentas de registro de toda a informação bruta coletada pelos sensores do veículo de modo que os algoritmos possam ser testados e ajustados *offline*.

Na segunda fase, o foco do trabalho foi a escolha e preparação do veículo para os testes. O laboratório dispõe de 3 veículos para teste, apresentados na Figura 74. Os veículos são compartilhados entre os pesquisadores e alunos do laboratório seguindo uma agenda de alocação, mas somente 5 pessoas têm autorização para dirigi-los.

O veículo denominado Deeva, o mais novo e sofisticado do laboratório, ainda estava em fase de montagem, por isso não estava disponível para testes de direção autônoma. Ele possui 25 câmeras e 25 computadores, além de sensores LASER e uma INS/GNSS de alta precisão. Em um teste experimental de direção em circuito fechado para avaliação da precisão da unidade INS/GNSS, similar ao realizado com o carro brasileiro (Seção 4.4.1), o erro médio de fechamento de caminho foi muito baixo, apenas 0,016%. O circuito usado para este teste foi o anel viário do campus da universidade de Parma, com 1870 m de comprimento.

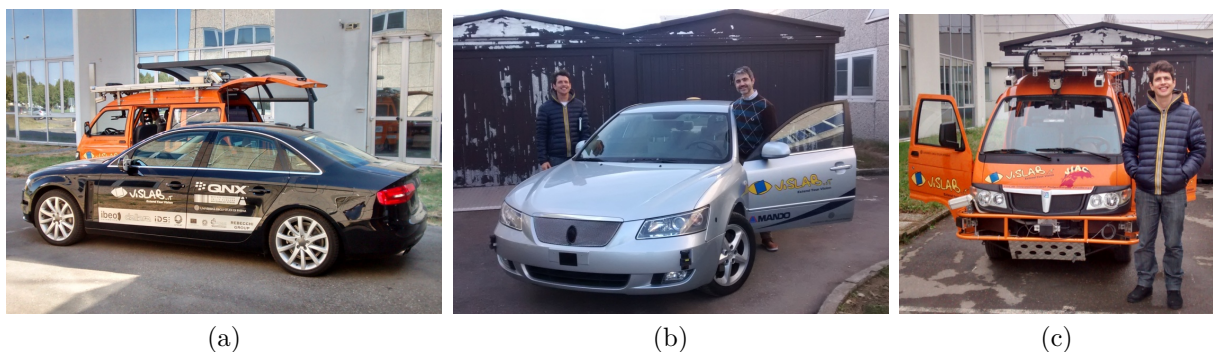


Figura 74 – Veículos de teste do laboratório: (a) Deeva; (b) BRAIVE; (c) PORTER.

O primeiro veículo a ser trabalhado para realização dos testes foi o denominado BRAIVE, o mesmo que foi usado na demonstração pública de direção autônoma PROUD (BROGGI et al., 2014). Este veículo foi preparado cerca de 2 meses mas os resultados obtidos nos testes do sistema de *dead reckoning* em circuito fechado não apresentaram bons resultados. A odometria com o BRAIVE foi feita aproveitando-se o sinal de velocidade das rodas, lido a partir dos sensores do freio ABS. O sinal de *yaw rate*, foi lido a partir dos dois giroscópios disponíveis mas ambos não apresentaram precisão suficiente devido a problemas de compensação térmica. Nos testes realizados ao redor do anel viário do campus da universidade de Parma, o erro médio de fechamento foi de 0,54%. Comparando-se com o valor de 0,1%, obtido com o carro brasileiro (Seção 4.4.1), constatou-se que a precisão ficou bem abaixo do esperado. Como já foi discutido na Seção 5.3.2, o bom desempenho da medição de postura pelo *map-matching* do BLMR, depende de uma alta repetitividade do sistema *dead reckoning*. Esta limitação inviabilizou a utilização do BRAIVE. A modificação da arquitetura de sensores do BRAIVE não foi permitida pois o veículo tem de estar

constantemente disponível para apresentações a empresas e pesquisadores visitantes.

A solução então foi utilizar o veículo denominado PORTER (Figura 74-c), o mesmo utilizado no desafio Intercontinental VIAC. Este veículo é o mais antigo do laboratório ainda em operação, e tem permissão do laboratório para que pequenas modificações no seu *hardware* sejam feitas. O veículo conta com 7 câmeras, 3 computadores, sensores LASER de varredura simples, e uma unidade de localização baseada em uma unidade GNSS com correção RTK. A câmera aproveitada para os experimentos com o algoritmo BLMR foi a câmera central de visão larga. O amplo campo de visão desta câmera em conjunto com o formato do veículo, capô curto, resultou numa captação de imagem muito favorável, o que contribuiu para uma ótima detecção das marcações de pista. O veículo possui propulsão elétrica e o atual desgaste das baterias limita sua autonomia a pouco mais de 30km por carga. Este fato dificultou um pouco a realização dos testes já que, em virtude do comprimento do trecho escolhido para os testes, isso representava no máximo duas viagens por cada dia de testes.

A unidade GNSS do PORTER é a mesma utilizada pelo BRAIVE, portanto, as medições de *yaw-rate* apresentaram o mesmo problema de baixa precisão para o cálculo de *dead reckoning*. Outro problema encontrado na utilização do PORTER foi que o veículo não possui *encoder* instalado nas rodas para medição direta de deslocamento e velocidade. Por estas razões foi necessário a construção de uma unidade de interface de veículo (Figura 75-a), similar à usada no carro do Brasil, e um *encoder* externo para uma das rodas (75-b). O encoder foi construído com peças aproveitadas de sucata e foi montado na roda traseira direita, conforme mostrado na Figura 75-c. A resolução obtida foi de 0,16 m, um pouco melhor que do carro brasileiro.

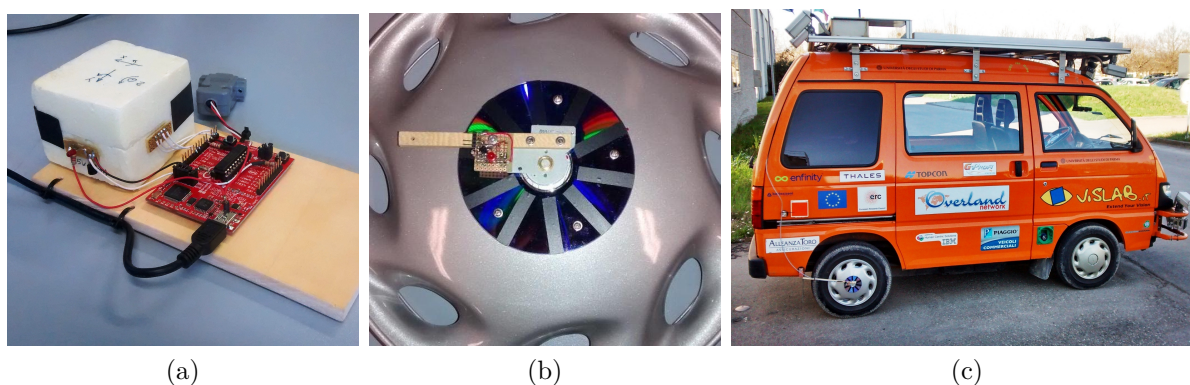


Figura 75 – Adaptações feitas no veículo PORTER: (a) Unidade de interface e giroscópio; (b) Encoder; (c) Montagem do encoder no carro.

Este novo arranjo de sensores do PORTER também foi submetido ao mesmo teste de fechamento de caminho no anél viário da Universidade de Parma (Figura 76-a). A Figura 76-b mostra o resultado comparativo entre os testes com o BRAIVE e o PORTER. O valor médio do erro encontrado para o BRAIVE foi de 0,54% contra o valor 0,08% para

o PORTER. O resultado obtido com o PORTER foi muito positivo, sendo o valor de erro menor do que os 0,1% obtidos com o carro brasileiro. Mais seis viagens foram realizadas no sentido horário e o resultado foi ainda melhor, 0,04%.

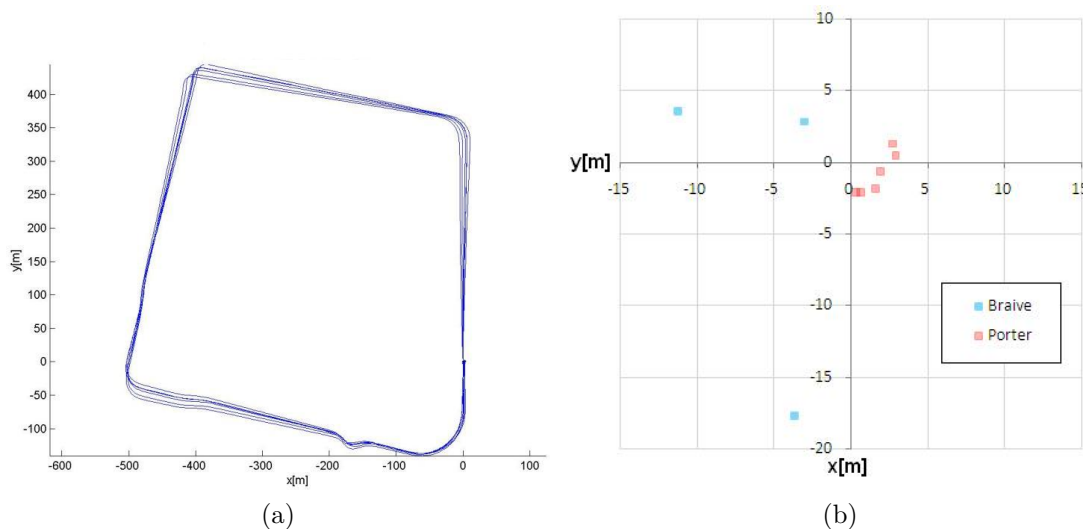


Figura 76 – Testes de caminho fechado: (a) D.R. registrado pelo veículo PORTER; (b) Posições finais PORTER e BRAIVE.

5.4.2.2 Viagem de Mapeamento

Uma vez que o sistema de *dead reckoning* e de detecção de marcações de pista estavam devidamente instalados e funcionando no PORTER, o veículo estava pronto para os testes do algoritmo de direção autônoma por BLMR, propriamente dito. O passo seguinte foi a realização da viagem de mapeamento para construção do mapa. O local escolhido para o mapeamento foi um trecho de 4,4 km da Estrada Montanara que fica na vizinhança da Universidade de Parma (Figura 77-a), iniciando na coordenada (N44,773166; L10,307766). Por se tratar de uma estrada rural, estreita, de mão dupla, sem acostamento e ainda com um trecho urbano, considerou-se a condição suficientemente complexa para avaliar rigidamente o desempenho do algoritmo BLMR. A escolha do local também levou em conta questões de logística pois a limitada autonomia do veículo exigiria um caminhão para transportá-lo até o local dos testes, caso fosse longe da Universidade.

Duas viagens de mapeamento foram realizadas, sendo o mapa resultante produzido por uma delas mostrado na Figura 77-b. O mapa possui um comprimento de 4,4km e está rotacionado para facilitar a comparação visual com o mapa rodoviário. A linha cinza representa o caminho percorrido pelo veículo, ou seja o caminho de referência para ser usado como guia na direção autônoma. As marcações de pista não podem ser visualizadas nesta escala pois fundem-se ao caminho de referência. O trecho mapeado apresenta uma boa combinação de curvas e retas. As curvas estão destacadas em preto e numeradas.

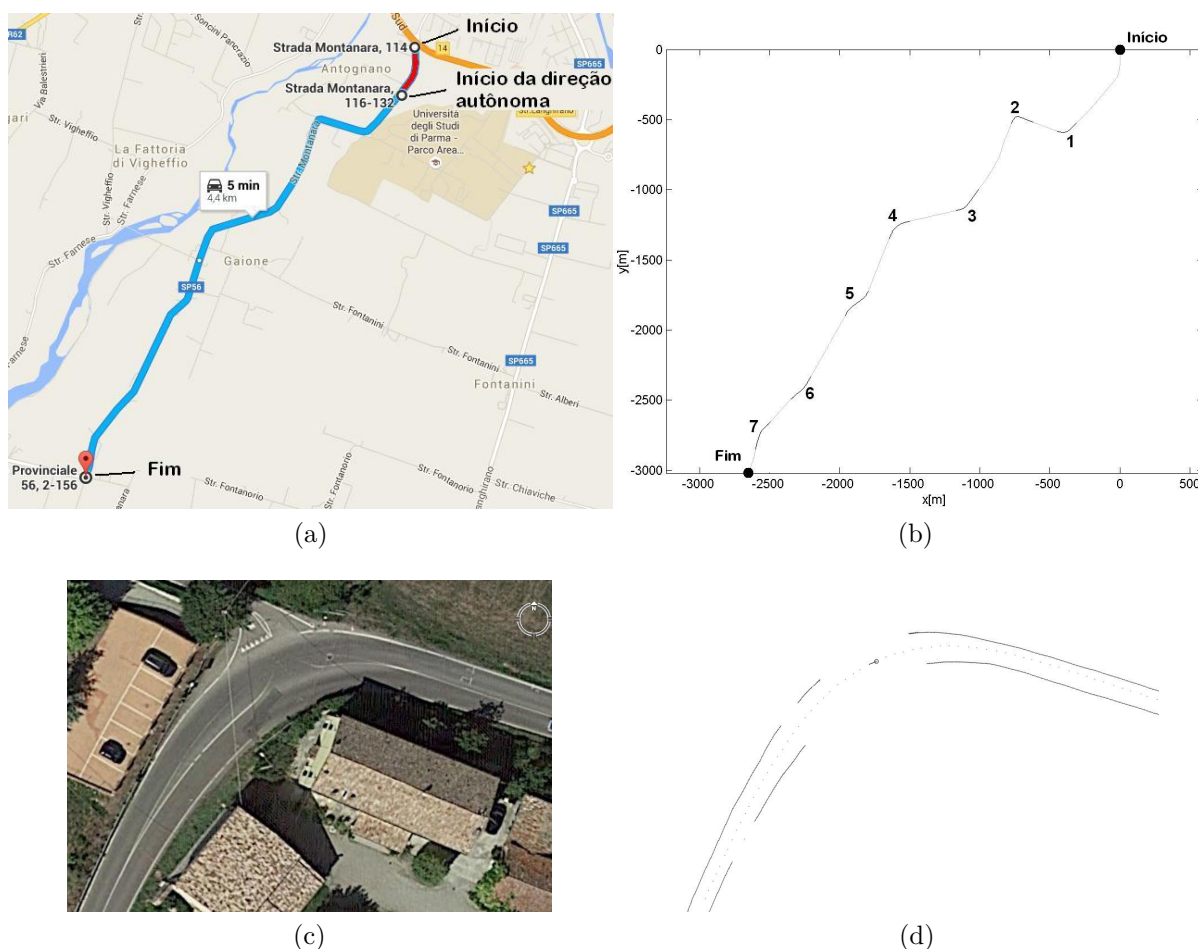


Figura 77 – Percurso mapeado para o teste de direção autônoma: (a) Mapa rodoviário; (b) mapa construído por *dead reckoning*; (c) foto de satélite da curva 2; (d) marcações detectadas na curva 2.

A curva 2 está mostrada em detalhes nas Figuras 77-c e 77-d. A Figura 77-c mostra a imagem de satélite desta curva e a Figura 77-d o mapa construído. Por se tratar de uma curva fechada (pequeno raio de curvatura) e ainda pelo fato de exatamente no meio desta curva existir um trecho faltando as marcações, este trecho do percurso foi considerado o mais desafiador para o algoritmo de direção autônoma por BLMR.

Para verificar a repetitividade do processo de mapeamento, a comparação entre os dois mapas construídos foi feita por *map-matching* em cada um dos trechos enumerados na Figura 77-b. O comprimento dos trechos comparados é igual ao comprimento determinado para o BLMR, ou seja, 240m. O *map-matching* calcula o módulo da diferença lateral entre os pontos correspondente dos dois mapas. Os valores encontrados (em cm) foram: 9,4; 6,6; 5,1; 12,6; 6,4; 8,7; 2,7. Este baixo valor de erro indicou a alta repetibilidade do sistema na construção dos mapas, o que é um requisito fundamental para o correto funcionamento do método BLMR.

5.4.2.3 Viagens com Direção Autônoma

Com todos os pré-requisitos atendidos passou-se então para a etapa final dos testes, ou seja a verificação da direção autônoma.

Foram realizados diversos testes de direção autônoma em diferentes condições de iluminação. Em <<https://youtu.be/AjMVzoUjv-Y>> pode-se acessar a gravação de uma viagem autônoma em um dia de céu azul com muitas sombras projetadas sobre a pista. Já em <<https://youtu.be/i9UNKuL5V8E>> pode-se acessar outra viagem no onde o veículo dirige um trecho diretamente contra o sol poente. Nestes vídeos pode-se notar que os sistema se mantém estável mesmo nos momentos mais difíceis, tolerando trechos sem linha central, sombras, falhas na marcação de pista. Dos 15 testes realizados todos foram completados com sucesso, exceto o teste 11, quando houve a falsa detecção do *guard rail* da curva 2, como se fosse marcação de pista. A sensação causada aos passageiros foi de uma direção estável e suave (sem guinadas).

Cada teste se divide nas seguintes etapas: estacionamento manual do veículo no início do mapa; inicialização do *software* e carregamento do mapa; espera do sinal de GNSS e confirmação da localização inicial, início de direção em modo manual até confirmação da localização precisa, habilitação do acionador do volante, supervisão da direção autônoma, desabilitação do acionador do volante. A Figura 78 mostra uma sequência de fotos registrando um desses testes. As fotos correspondem aos pontos do trajeto indicados na Figura 77-b.

Os dados coletados nestas viagens estão disponíveis em <<http://bit.ly/1MH1Pqb>>. Assim como no caso anterior, foram incluídos arquivos de script do MATLAB para facilitar a análise dos resultados.

Em relação ao sistema de segurança, todos os testes foram realizados sempre com duas pessoas a bordo do veículo. O carro utilizado dispõe de um servomotor para atuação do volante que permite limitar o torque aplicado no acionamento. Desse modo, sempre que alguma intervenção manual fosse necessária, bastava o motorista de segurança mover o volante com as próprias mãos, sem grande dificuldade. As primeiras viagens autônomas foram feitas aos domingos e em baixa velocidade (40km/h) para não atrapalhar o trânsito normal e aumentar a segurança. Em seguida a velocidade dos testes foi aumentada até 60 km/h e depois do quinto teste, o sistema mostrou-se estável o suficiente para continuar os testes em condições de tráfego normal.

5.4.3 Desempenho computacional

A análise do desempenho computacional do algoritmo de localização foi feita em termos do tempo gasto para processar o algoritmo em um computador específico. O objetivo desta análise é ter uma ideia aproximada da carga de processamento do algoritmo

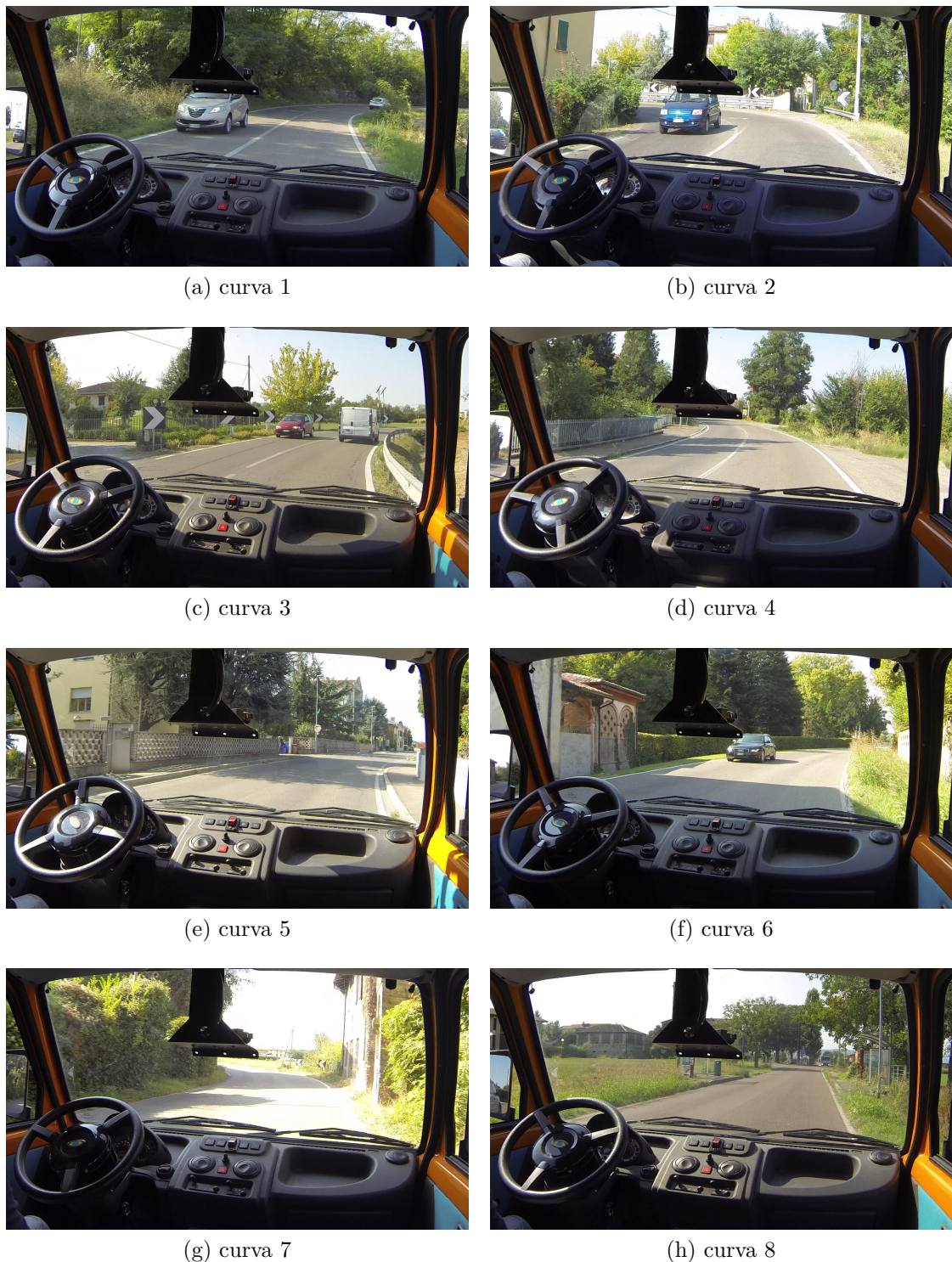


Figura 78 – Fotos de um teste de direção autônoma.

implementado, a fim de verificar a possibilidade de ele ser aplicado em sistemas de computadores menores e mais baratos, o que favorece o uso comercial. Foi utilizado um *notebook* com processador Intel Celeron de 1,86 GHz. A Tabela 6 mostra o tempo gasto em cada etapa do processo de localização.

Considerando que a aquisição de um novo quadro de imagem leva 10 ms e o

Tabela 6 – Desempenho Computacional: tempo gasto em cada processo do sistema de localização.

#	Processo	Tempo (ms)
1	Detecção das marcações de pista	3,90
2	Construção do BLMR	0,01
3	Localização por BLMR	2,38
4	Filtragem (predição e atualização em modo 3)	0,37
5	Outros	1,00
	Total	7,66

tempo de processamento do algoritmo mostrado na Tabela 6, o sistema de localização está apto a operar *online* com o carro dirigindo a velocidades de até 271 km/h (1,33 m em 17,66 ms). O tempo necessário somente para o algoritmo de localização pode ser computado considerando apenas os passos 2 a 5. O valor obtido é de 3,76 ms. Este resultado demonstra que o algoritmo proposto é adequado para rodar em computadores compactos e de baixo custo.

5.5 Conclusão e Discussão

Este capítulo apresenta uma nova abordagem de localização e direção autônoma baseada em sensores de baixo custo. Mais especificamente, o sistema proposto propõe se a resolver o problema de seguimento autônomo de um trajeto rodoviário pré mapeado em vias sinalizadas.

A abordagem usa uma câmera frontal e uma sistema *dead reckoning* para construir um registro contínuo das marcações de pista por onde o veículo já passou (*Back Lane Marking Registry* - BLMR). Esse registro é utilizado para a localização precisa do veículo no mapa, o que permite a realização de direção autônoma. Pelo fato do BLMR conter a informação de um trecho relativamente grande (240 m), e, o sistema usar *dead reckoning*, a abordagem proposta pode lidar com situações onde as marcações de pista não estão presentes ou não podem ser detectadas por curtas distâncias (150m). O BLMR mostrou-se um método preciso e confiável capaz de construir uma extensa percepção do entorno do veículo a partir de uma câmera monocular e cálculo de *dead reckoning*. Esta, sem dúvida, é a grande contribuição deste trabalho.

As marcações de pista contidas no BLMR são usada para localizar o veículo por meio de um algoritmo de *map-matching* otimizado para velocidade, em função das características da estrutura de dados do mapa/BLMR e das exigências da direção autônoma (alta precisão lateral e média longitudinal). Além disso, a estimativa de posição foi melhorada com o esquema de filtragem proposto, que usa ganhos independentes para cada componente de localização (lateral, longitudinal, orientação), e considera a incerteza da

medição longitudinal de posição.

Todos os cuidados para otimizar e baixar o custo da arquitetura do algoritmo resultaram em um sistema de localização de custo muito baixo quando comparado com qualquer sistema similar. Isto é uma importante questão quando se pensa em aplicações comerciais em larga escala.

O algoritmo de localização usa diversos parâmetros empíricos para funcionar, tais como comprimento do BLMR, ganhos do filtro (k_c, k_{lmax}, k_h), valor de corte da função de incerteza longitudinal γ , entre outros. Todos estes parâmetros, assim como a geometria particular da estrada, podem influenciar na precisão alcançada, medida através do erro lateral do ponto alvo (ELPA). Apesar de não ter sido feito nenhum estudo quantitativo mais profundo sobre a sensibilidade do algoritmo (variação de precisão / variação dos parâmetros), alguns resultados preliminares indicam uma baixa sensibilidade. Por exemplo, o comprimento do BLMR foi testado para comprimentos variando entre 60m e 540m e a variação correspondente do erro (ELPA) nesse intervalo foi apenas 0,56 ($ELPA_{min}/ELPA_{max}$). Já os parâmetros do filtro com seus valores inicialmente arbitrados já produziram um resultado próximo ao valor otimizado mesmo com a variação de 2,5 vezes no ganho transversal.

6 Conclusão

Este trabalho é uma contribuição na área de sistemas inteligentes de transporte, mais especificamente através do desenvolvimento de um sistema de direção autônoma confiável e de baixo custo, aplicável a veículos rodoviários (carros, caminhões). Para atender à restrição de custo, optou-se por uma abordagem de percepção puramente visual, e o único sensor exteroceptivo utilizado após a localização inicial é uma câmera. Apesar de não ser o foco principal desse trabalho, os esforços dedicados à construção de um veículo protótipo próprio foram decisivos para alcançar uma arquitetura de baixo custo. O trabalho principal se desenvolveu através do estudo de três técnicas amplamente utilizadas em trabalhos relacionados: direção autônoma por seguimento de marcadores de pista, direção autônoma por sistemas de localização global e direção autônoma por técnicas de mapeamento e localização.

No Capítulo 3, estudou-se a direção autônoma por seguimento de marcadores de pista, uma das primeiras técnicas aplicadas a este problema que alcançou resultados expressivos. Apesar de suas limitações no controle direto do veículo, o detector de faixas desenvolvido neste estudo serviu de base para a abordagem baseada em localização e mapas, do Capítulo 5. No Capítulo 4, estudou-se a direção autônoma por localização global. Apesar de ser conhecido que este tipo de abordagem não alcança precisão suficiente para direção autônoma em qualquer condição, este estudo serviu tanto para avaliar o desempenho de uma implementação de um sistema *dead reckoning* de baixo custo quanto para desenvolver e testar o controlador de seguimento de caminho de referência aproveitados na abordagem do Capítulo 5. Por fim, no Capítulo 5, estudou-se a direção autônoma por técnicas de mapeamento e localização, técnica que tem sido usada nos trabalhos mais recentes e produzido os melhores resultados. Este estudo constitui o corpo principal desta tese, que produziram os melhores e mais significativos resultados deste trabalho.

As técnicas de mapeamento e localização estudadas nos trabalhos relacionados usam as informações instantâneas dos sensores externos (câmeras, LASER) para, através de um processo de filtragem (Kalman ou partículas), determinar a localização do veículo no mapa. A grande novidade deste trabalho está na maneira como a percepção do ambiente é feita. O método se baseia no registro das marcações de pista por onde o veículo já passou (*Back Lane Marking Registry* - BLMR). O BLMR mostrou-se um método leve, preciso e confiável, capaz de construir uma extensa percepção das marcações de pista no entorno do veículo a partir de uma câmera monocular e cálculo de *dead reckoning*.

Os testes experimentais de precisão feitos no Brasil e os testes de direção autônoma realizados na Itália demonstraram que, apesar da arquitetura baseada em sensores muito

simples e baratos, o sistema é plenamente capaz de operar com grande confiabilidade mesmo em percursos considerados difíceis. Além disso, é importante destacar que apesar do sistema usar visão computacional, considerado uma técnica computacionalmente pesada, o algoritmo foi capaz de rodar em um computador bastante defasado. Essa combinação de uma arquitetura leve, de baixo custo, alta precisão e confiabilidade constitui uma relevante contribuição ao estado da arte.

É claro que o algoritmo proposto ainda precisa ser testado em diferentes percursos e por muito mais tempo para se ter uma visão mais ampla de suas peculiaridades. Também é importante frisar que o algoritmo BLMR não é um sistema de direção autônoma completo, já que a capacidade de detectar e reagir a obstáculos móveis também precisa ser incorporada, que está previsto como trabalho futuro.

Entretanto, acredita-se que os dois objetivos mais importantes do trabalho foram alcançados. A concepção de um sistema de localização leve e de baixo custo, mas com precisão suficiente para realizar direção autônoma em situações complexas.

6.1 Trabalhos futuros

Considerando as opções de trabalho futuro, as questões principais podem ser descritas em quatro tópicos: realização de mais testes; uso de um esquema de filtragem mais elaborado, uso de mais tipos de *landmarks* para superar a limitação dos trechos retos de estradas, detecção e tratamento de obstáculos móveis, e implementação do sistema em *hardware* compacto.

Espera-se que o uso de um esquema mais elaborado de filtragem, como filtragem probabilística, conduzirá a uma estimativa de localização mais precisa e robusta, já que a confiabilidade da medição de pose por BLMR é variável (Seção 5.3.2). Como dito anteriormente, a localização longitudinal não é possível de ser realizada em trechos retos longos. De um modo geral, a confiabilidade da medição longitudinal será maior de acordo com a quantidade e intensidade das curvas contidas no BLMR. A Figura 79 ilustra esta ideia. O trecho A é a condição mais favorável com a mais alta confiabilidade, o trecho B é uma condição intermediária, e o trecho C o pior caso. Além disso a confiabilidade também depende da quantidade trechos de falta de marcadores.

Outros tipos de *landmarks* comuns e presentes em estradas padrões podem ser usados para melhorar a precisão e a disponibilidade do sistema de localização, especialmente para a componente longitudinal. Um sistema de visão capaz de detectar artefatos rodoviários, como faixa de parada ou faixa de pedestre, poderiam ser usados para eliminar o problema de longos trechos retos, onde o BLMR não consegue calcular com precisão sua localização.

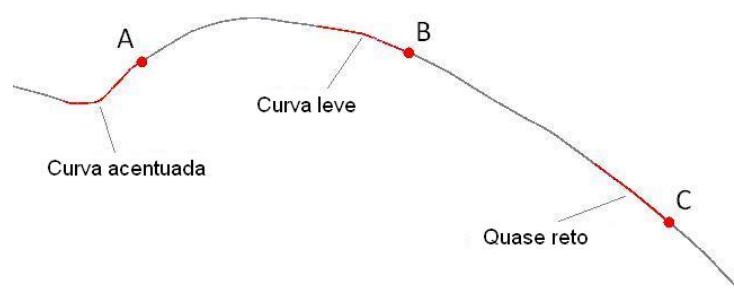


Figura 79 – Exemplo de um trecho de uma estrada com diferentes características que afetam a localização longitudinal: (a) baixa incerteza, (b) media incerteza, (c) alta incerteza.

A última linha de trabalho futuro é a detecção e o tratamento dos obstáculos dinâmicos, tais como pedestres, outros automóveis, sinalização semafórica. Está é uma questão fundamental para se chegar a um sistema de direção autônoma completo.

6.2 Considerações sobre o Desempenho

A avaliação de desempenho de um sistema de direção autônoma é uma questão um tanto subjetiva. Basta se observar como as próprias pessoas "medem" o quão bem elas mesmas dirigem. Enquanto para uns dirigir bem significa ser rápido, para outras dirigir bem significa ser seguro. Nos projetos de carro autônomo estudados na revisão bibliográfica, o desempenho geralmente é apresentado na forma de características como o comprimento da viagem, velocidade média e percentual do percurso feito em modo autônomo.

O fato de um carro dirigir, por exemplo, 98% de um percurso em modo autônomo pode dizer muito mais sobre as características da estrada do que do próprio sistema de direção autônoma. Os resultados dos testes feitos com o sistema de direção do carro protótipo revelaram que em condições boas de sinalização, o carro é capaz de dirigir indefinidamente sem falha.

A questão é, de certo modo, até filosófica. O que é melhor: um carro que faz uma determinada viagem em 20 minutos mas a probabilidade de um acidente é $1/1000000$ ou um carro que faz a mesma viagem em 15 minutos com probabilidade de acidente de $1/10000$? A resposta é pessoal e depende da disposição de cada um em assumir riscos para obter alguma vantagem. Ao tentar medir o desempenho de um sistema de direção, algumas questões devem ser respondidas. Como lidar com as situações adversas inesperadas? Como garantir que o sistema desenvolvido vá reagir com segurança a estas situações (algumas vezes inimagináveis)? Como medir a segurança?

Ainda não há um protocolo padronizado para medir o desempenho do processo de direção e essa provavelmente será uma questão bastante complexa a ser tratada no futuro.

Um protocolo para medição de desempenho de direção autônoma justo deveria submeter o veículo candidato a um rico e variado conjunto de situações de trânsito a fim de avaliar o comportamento do carro e eventuais situações de risco. Para que um veículo candidato não seja beneficiado em relação a outro, tanto o circuito de testes quanto as situações de trânsito deveriam ser sempre exatamente as mesmas. Repetir um circuito de testes não é um problema, porém repetir uma determinada situação de trânsito é. Imagine estabelecer uma área de testes padronizada, rica o suficiente para reproduzir as situações de rua onde o carro candidato vai operar. Agora imagine ainda que todos os obstáculos móveis devam estar se movendo sempre da mesma maneira em cada realização de teste. Sem dúvida não é uma questão simples de se resolver, porém é o mais próximo do ideal para afirmar que um sistema de direção autônoma é melhor do que outro.

Com o progresso e a popularização inevitável dos sistemas de direção autônoma, outro problema que terá que ser tratado, é a questão da responsabilidade por acidentes e prejuízos e suas implicações legais.

Referências

- A380, W. A. *Airbus A380*. 2014. Disponível em: <http://en.wikipedia.org/wiki/Airbus_A380>. Citado na página 17.
- ADVANCED Avionics HandBook. [S.l.]: Federal Aviation Administration, 2009. Citado na página 17.
- ASIRT, A. for S. I. R. T. *Annual Global Road Crash Statistics*. 2013. Disponível em: <<http://www.asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics.aspx>>. Citado na página 16.
- AUFRERE, R.; CHAPUIS, R.; CHAUSSE, F. A fast and robust vision based road following algorithm. In: *Intelligent Vehicles Symposium, 2000. IV 2000. Proceedings of the IEEE*. [S.l.: s.n.], 2000. p. 192–197. Citado 5 vezes nas páginas 31, 32, 59, 66 e 72.
- BADINO, H.; HUBER, D.; KANADE, T. Visual topometric localization. In: *IEEE Intelligent Vehicles Symposium (IV), 2011 IEEE*. [S.l.], 2011. p. 794–799. Citado 4 vezes nas páginas 12, 33, 34 e 120.
- BAY, H. et al. Speeded-up robust features (surf). *Computer Vision and Image Understanding*, v. 110, n. 3, p. 346 – 359, 2008. ISSN 1077-3142. Similarity Matching in Computer Vision and Multimedia. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1077314207001555>>. Citado na página 25.
- BERTOZZI, M. et al. Gold: A framework for developing intelligent-vehicle vision applications. *IEEE Intelligent Systems*, v. 23, n. 1, p. 69–71, Jan 2008. ISSN 1541-1672. Citado na página 120.
- BERTOZZI, M. et al. VIAC Expedition Toward Autonomous Mobility. *Robotics and Automation Magazine*, IEEE, v. 18, n. 3, p. 120–124, set. 2011. ISSN: 1070-9932. Citado 2 vezes nas páginas 24 e 120.
- BERTOZZI, M.; BROGGI, A.; FASCIOLI, A. VisLab and the Evolution of Vision-Based UGVs. *IEEE Computer*, IEEE Computer Society, Los Alamitos, CA, USA, v. 39, n. 12, p. 31–38, dez. 2006. ISSN: 0018-9162. Citado na página 25.
- BESL, P. J.; MCKAY, N. D. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 14, n. 2, p. 239–256, Feb 1992. ISSN 0162-8828. Citado 2 vezes nas páginas 35 e 106.
- BORKAR, A.; HAYES, M.; SMITH, M. T. A novel lane detection system with efficient ground truth generation. *Intelligent Transportation Systems, IEEE Transactions on*, IEEE, v. 13, n. 1, p. 365–374, 2012. Citado 2 vezes nas páginas 25 e 31.
- BROGGI, A.; BERTOZZI, M.; FASCIOLI, A. ARGO and the MilleMiglia in Automatico Tour. *IEEE Intelligent Systems*, IEEE Press, Piscataway, USA, v. 14, n. 1, p. 55–64, jan.–fev. 1999. Citado na página 30.

- BROGGI, A. et al. The argo autonomous vehicle's vision and control systems. *International Journal of Intelligent Control and Systems*, p. 409–441, 1999. Citado 9 vezes nas páginas 21, 31, 32, 60, 64, 65, 66, 69 e 72.
- BROGGI, A. et al. TerraMax Vision at the Urban Challenge 2007. *IEEE Trans. on Intelligent Transportation Systems*, IEEE Press, v. 11, n. 1, p. 194–205, mar. 2010. Citado 2 vezes nas páginas 23 e 120.
- BROGGI, A. et al. Terrain Mapping for Off-Road Autonomous Ground Vehicles Using Rational B-Spline Surfaces and Stereo Vision. In: *Procs. IEEE Intelligent Vehicles Symposium 2013*. Gold Coast, Australia: [s.n.], 2013. p. 648–653. Citado na página 31.
- BROGGI, A. et al. PROUD-Public road urban driverless test: Architecture and results. In: *Procs. IEEE Intelligent Vehicles Symposium 2014*. Dearborn, MI, USA: [s.n.], 2014. p. 684–654. Citado 2 vezes nas páginas 33 e 121.
- CAR, W. A. *Autonomous car 1960*. 1960. Disponível em: <http://en.wikipedia.org/wiki/Autonomous_car#1960s>. Citado na página 21.
- CARROS, P. O. F. *Informações Modelo Fusion*. 2014. Disponível em: <<http://www.ford.com.br/carros/fusion>>. Citado na página 17.
- CHOI, H.-C. et al. Vision-based fusion of robust lane tracking and forward vehicle detection in a real driving environment. *International Journal of Automotive Technology*, The Korean Society of Automotive Engineers, v. 13, n. 4, p. 653–669, 2012. ISSN 1229-9138. Disponível em: <<http://dx.doi.org/10.1007/s12239-012-0064-x>>. Citado 9 vezes nas páginas 31, 32, 33, 60, 61, 64, 66, 69 e 72.
- CHOI, S.; KIM, T.; YU, W. Performance evaluation of ransac family. *Journal of Computer Vision*, v. 24, n. 3, p. 271–300, 1997. Citado na página 64.
- Color Vision For Road Following*, v. 1007. 175-185 p. Disponível em: <<http://dx.doi.org/10.1117/12.949096>>. Citado na página 21.
- CUI, D.; XUE, J.; ZHENG, N. Real-time global localization of robotic cars in lane level via lane marking detection and shape registration. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 4, p. 1039–1050, April 2016. ISSN 1524-9050. Citado 3 vezes nas páginas 12, 34 e 120.
- CUI, G.; WANG, J.; LI, J. Robust multilane detection and tracking in urban scenarios based on lidar and mono-vision. *Image Processing, IET, IET*, v. 8, n. 5, p. 269–279, 2014. Citado 2 vezes nas páginas 25 e 31.
- DAHLKAMP, H. Selfsupervised monocular road detection in desert terrain. In: *In Proc. of Robotics: Science and Systems (RSS)*. [S.l.: s.n.], 2006. Citado na página 22.
- DARPA, D. A. R. P. A. *DARPA Grand Challenge 2004*. 2004. Disponível em: <<http://archive.darpa.mil/grandchallenge04/>>. Citado na página 21.
- DARPA, D. A. R. P. A. *DARPA Grand Challenge 2005*. 2005. Disponível em: <<http://archive.darpa.mil/grandchallenge05/>>. Citado na página 21.

DESOUZA, G. N.; KAK, A. C. Vision for mobile robot navigation: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, IEEE, v. 24, n. 2, p. 237–267, 2002. Citado 2 vezes nas páginas 33 e 34.

DIETMAYER, K. et al. Roadway detection and lane detection using multilayer laserscanner. In: *Advanced Microsystems for Automotive Applications 2005*. [S.l.]: Springer, 2005. p. 197–213. Citado 2 vezes nas páginas 25 e 30.

EUREKA, i. a. b. *EUREKA PROMETHEUS*. 1987. Disponível em: <<http://www.eurekanetwork.org/project/-/id/45>>. Citado na página 21.

FACTORY, M. *Bleriot XI Trainer / Reconnaissance Monoplane (1909)*. 2003. Disponível em: <http://www.militaryfactory.com/aircraft/detail.asp?aircraft_id=719>. Citado na página 17.

FISCHLER, M. A.; BOLLES, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, ACM, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/358669.358692>>. Citado na página 32.

GOPALAN, R. et al. A learning approach towards detection and tracking of lane markings. *Intelligent Transportation Systems, IEEE Transactions on*, IEEE, v. 13, n. 3, p. 1088–1098, 2012. Citado 2 vezes nas páginas 25 e 31.

HATA, A.; WOLF, D. Road marking detection using lidar reflective intensity data and its application to vehicle localization. In: IEEE. *Intelligent Transportation Systems (ITSC), 2014 IEEE 17th International Conference on*. [S.l.], 2014. p. 584–589. Citado 6 vezes nas páginas 12, 25, 30, 33, 34 e 120.

HILLEL, A. B. et al. Recent progress in road and lane detection: a survey. *Machine Vision and Applications*, Springer, v. 25, n. 3, p. 727–745, 2014. Citado na página 30.

HORN, B. K. P. Closed-form solution of absolute orientation using unit quaternions. *J. Opt. Soc. Am. A*, OSA, v. 4, n. 4, p. 629–642, Apr 1987. Disponível em: <<http://josaa.osa.org/abstract.cfm?URI=josaa-4-4-629>>. Citado na página 108.

HUANG, S.-S.; FU, L.-C. Driver assistance system for lane detection and vehicle recognition with night vision. In: *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on*. [S.l.: s.n.], 2005. p. 3530–3535. Citado na página 30.

ICCM-USP, L. d. R. M. *Projeto CARINA*. 2010. Disponível em: <http://www.lrm.icmc.usp.br/?page=projetos&projeto=carina_home>. Citado na página 24.

INOVATION, P. O. M. B. *Inovações tecnológicas*. 2014. Disponível em: <<http://www5.mercedes-benz.com/en/innovation/>>. Citado na página 17.

KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Fluids Engineering*, American Society of Mechanical Engineers, v. 82, n. 1, p. 35–45, 1960. Citado na página 33.

KAUR, G.; KUMAR, D. Lane detection techniques: A review. *International Journal of Computer Applications*, Foundation of Computer Science, v. 112, n. 10, 2015. Citado na página 30.

KIM, H. et al. Robust vehicle localization using entropy-weighted particle filter-based data fusion of vertical and road intensity information for a large scale urban area. *IEEE Robotics and Automation Letters*, v. 2, n. 3, p. 1518–1524, July 2017. ISSN 2377-3766. Citado 2 vezes nas páginas 24 e 25.

LABS, A. *Autonomos Project*. 2011. Disponível em: <<http://www.autonomos.inf.fu-berlin.de/>>. Citado na página 23.

LATEGAHN, H.; STILLER, C. City gps using stereo vision. In: IEEE. *Vehicular Electronics and Safety (ICVES), 2012 IEEE International Conference on*. [S.l.], 2012. p. 1–6. Citado 4 vezes nas páginas 12, 33, 34 e 120.

LCAD, L. de Computação de A. D. *Projeto Veiculo Autonomo do LCAD*. 2010. Disponível em: <<http://lcadufes.wordpress.com/>>. Citado na página 24.

LEARN, Y. S. *LBS building blocks: Global Positioning System*. 2013. Disponível em: <<http://yourstory.com/2013/03/ys-learn-lbs-building-blocks-global-positioning-system-gps/>>. Citado 2 vezes nas páginas 8 e 27.

LEVINSON, J.; MONTEMERLO, M.; THRUN, S. Map-based precision vehicle localization in urban environments. In: *Robotics: Science and Systems*. [S.l.: s.n.], 2007. v. 4, p. 1. Citado 4 vezes nas páginas 24, 25, 30 e 33.

LEVINSON, J.; THRUN, S. Robust vehicle localization in urban environments using probabilistic maps. In: IEEE. *Robotics and Automation (ICRA), 2010 IEEE International Conference on*. [S.l.], 2010. p. 4372–4378. Citado 3 vezes nas páginas 24, 30 e 33.

LIMA, D. de; PEREIRA, G. Navegação segura de um carro autônomo utilizando campos vetoriais e o método da janela dinâmica. In: *Anais do X Simpósio Brasileiro de Automação Inteligente*. Universidade Federal de São João del-Rei, São João del-Rei, MG, Brasil: [s.n.], 2011. p. 1167–1172. Disponível em: <<http://www.sbai2011.ufsj.edu.br>>. Citado na página 23.

LU, G. *A Lane Detection, Tracking and Recognition System for Smart Vehicles*. Tese (Doutorado) — University of Ottawa, 2015. Citado na página 30.

Vision-based real-time road detection in urban traffic, v. 4666. 75–82 p. Disponível em: <<http://dx.doi.org/10.1117/12.458518>>. Citado 3 vezes nas páginas 21, 32 e 72.

LYRIO, L. J. et al. Image-based mapping, global localization and position tracking using vg-ram weightless neural networks. In: *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. [S.l.: s.n.], 2015. p. 3603–3610. Citado 4 vezes nas páginas 12, 33, 34 e 120.

MATTERN, N.; SCHUBERT, R.; WANIELIK, G. High-accurate vehicle localization using digital maps and coherency images. In: *Intelligent Vehicles Symposium (IV), 2010 IEEE*. [S.l.: s.n.], 2010. p. 462–469. Citado 2 vezes nas páginas 33 e 34.

MCCALL, J.; TRIVEDI, M. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *Intelligent Transportation Systems, IEEE Transactions on*, v. 7, n. 1, p. 20–37, March 2006. Citado na página 30.

- MIAO, X.; LI, S.; SHEN, H. On-board lane detection system for intelligent vehicle based on monocular vision. *International journal on smart sensing and intelligent systems*, v. 5, n. 4, p. 957–972, 2012. Citado na página 30.
- MOTORS, P. O. T. *Informações de desempenho Tesla Motors*. 2014. Disponível em: <<http://www.teslamotors.com/models>>. Citado na página 17.
- NASHMAN, M.; SCHNEIDERMAN, H. Real-time visual processing for autonomous driving. In: *Intelligent Vehicles '93 Symposium*. [S.l.: s.n.], 1993. p. 373–378. Citado 3 vezes nas páginas 31, 32 e 72.
- ONIGA, F.; NEDEVSKI, S. Processing dense stereo data using elevation maps: Road surface, traffic isle, and obstacle detection. *Vehicular Technology, IEEE Transactions on*, IEEE, v. 59, n. 3, p. 1172–1182, 2010. Citado na página 31.
- PATON, M. et al. Expanding the limits of vision-based localization for long-term route-following autonomy. *Journal of Field Robotics*, v. 34, n. 1, p. 98–122, 2017. ISSN 1556-4967. Disponível em: <<http://dx.doi.org/10.1002/rob.21669>>. Citado 2 vezes nas páginas 24 e 25.
- PINK, O. Visual map matching and localization using a global feature map. In: IEEE. *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*. [S.l.], 2008. p. 1–7. Citado na página 33.
- PIŠI, D.; KALOVÁ, I. *Generating stepper motor speed profiles using FPGA*. 2011. Citado na página 50.
- POMERLEAU, D.; JOCHEM, T. Rapidly adapting machine vision for automated vehicle steering. *IEEE Expert*, v. 11, n. 2, p. 19–27, Apr 1996. ISSN 0885-9000. Citado 3 vezes nas páginas 21, 31 e 72.
- S500, P. O. M. B. *The S 500 intelligent drive concept car*. 2013. Disponível em: <<http://www5.mercedes-benz.com/en/innovation/tv/?p=4>>. Citado na página 23.
- SABBAGH, V. B. et al. Desenvolvimento de um sistema de controle para um carro de passeio autônomo. In: *XVIII Congresso Brasileiro de Automática*. [S.l.: s.n.], 2010. Citado na página 24.
- SCANIABRASIL, N. *Scania e USP desenvolvem caminhão autônomo no Brasil*. 2015. Disponível em: <<http://newsroom.scania.com/br/2015/07/29/scania-e-usp-desenvolvem-caminhao-autonomo-no-brasil/>>. Citado na página 24.
- SCHNEIDERMAN, H.; NASHMAN, M. A discriminating feature tracker for vision-based autonomous driving. *Robotics and Automation, IEEE Transactions on*, v. 10, n. 6, p. 769–775, Dec 1994. ISSN 1042-296X. Citado 2 vezes nas páginas 21 e 72.
- SEO, Y.-W.; HWANGBO, M. A computer vision system for lateral localization. *Journal of Field Robotics*, 2015. Citado na página 30.
- SHEN, M. et al. Improving localization accuracy in connected vehicle networks using rao-blackwellized particle filters: Theory, simulations, and experiments. *CoRR*, abs/1702.05792, 2017. Disponível em: <<http://arxiv.org/abs/1702.05792>>. Citado na página 26.

- SPANGENBERG, M.; CALMETTES, V.; TOURNERET, J.-Y. Fusion of gps, ins and odometric data for automotive navigation. In: *Proc. of EUSIPCO*. [S.l.: s.n.], 2007. Citado na página 29.
- SPECTRUM, I. *How Google's Self-Driving Car Works*. 2011. Disponível em: <<http://spectrum.ieee.org/automaton/robotics/artificial-intelligence/how-google-self-driving-car-works>>. Citado na página 23.
- T, W. F. M. *Ford Model T*. 1908. Disponível em: <http://en.wikipedia.org/wiki/Ford_Model_T>. Citado na página 17.
- TESLAMOTORS. *AutoPilot Software for Model S*. 2016. Disponível em: <<https://www.teslamotors.com/presskit/autopilot>>. Citado na página 23.
- THRAPP, R.; WESTBROOK, C.; SUBRAMANIAN, D. Robust localization algorithms for an autonomous campus tour guide. In: *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*. [S.l.: s.n.], 2001. v. 2, p. 2065–2071 vol.2. ISSN 1050-4729. Citado na página 29.
- THRUN, S. et al. Stanley: The robot that won the darpa grand challenge: Research articles. *J. Robot. Syst.*, John Wiley and Sons Ltd., Chichester, UK, v. 23, n. 9, p. 661–692, set. 2006. ISSN 0741-2223. Disponível em: <<http://dx.doi.org/10.1002/rob.v23:9>>. Citado 2 vezes nas páginas 22 e 29.
- UBLOX. *u-blox 3D Automotive Dead Reckoning technology*. 2014. Disponível em: <<http://www.u-blox.com/en/dead-reckoning.html>>. Citado na página 29.
- VAMP, W. *VaMP*. 1994. Disponível em: <<https://en.wikipedia.org/wiki/VaMP>>. Citado na página 21.
- VIOLA, P.; JONES, M. Rapid object detection using a boosted cascade of simple features. In: *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*. [S.l.: s.n.], 2001. v. 1, p. I-511–I-518 vol.1. ISSN 1063-6919. Citado na página 61.
- VIRGINIATECH. *Precision Farming Tools: Global Position System*. Disponível em: <<http://pubs.ext.vt.edu/442/442-503/442-503.html>>. Citado 2 vezes nas páginas 8 e 27.
- VISLAB, A. V. I. S. L. *Public ROad Urban Driverless-Car Test 2013*. 2013. Disponível em: <<http://vislab.it/proud-en/>>. Citado 2 vezes nas páginas 23 e 120.
- VIVACQUA, R.; VASSALLO, R.; MARTINS, F. A low cost sensors approach for accurate vehicle localization and autonomous driving application. *Sensors*, v. 17, n. 10, 2017. Citado na página 37.
- VIVACQUA, R. P. D. et al. Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, PP, n. 99, p. 1–16, 2017. ISSN 1524-9050. Citado na página 37.
- VIVACQUA, R. P. D.; MARTINS, F. N.; VASSALLO, R. F. Direção autônoma baseada em visão computacional. In: *Anais do XIX Congresso Brasileiro de Automática, CBA 2012*. [S.l.: s.n.], 2012. p. 1430–1425. Citado 2 vezes nas páginas 37 e 82.

- VIVACQUA, R. P. D.; MARTINS, F. N.; VASSALLO, R. F. Visual lane tracking and prediction for autonomous vehicles. In: *Simpósio Brasileiro de Automação Inteligente, SBAI 2013*. [S.l.: s.n.], 2013. Citado 2 vezes nas páginas 37 e 72.
- VOLVOCARS. *Volvo Intellisafe AutoPilot Systems*. 2016. Disponível em: <<http://www.volvocars.com/au/about/innovations/intellisafe/autopilot>>. Citado na página 23.
- WINGS, H. *George the Auto Pilot*. 2012. Disponível em: <<http://fly.historicwings.com/2012/08/george-the-autopilot/>>. Citado na página 17.
- XUE, J.-r. et al. A vision-centered multi-sensor fusing approach to self-localization and obstacle perception for robotic cars. *Frontiers of Information Technology & Electronic Engineering*, v. 18, n. 1, p. 122–138, 2017. ISSN 2095-9230. Disponível em: <<http://dx.doi.org/10.1631/FITEE.1601873>>. Citado na página 25.
- YANG, S. et al. Road constrained monocular visual localization using gaussian-gaussian cloud model. *IEEE Transactions on Intelligent Transportation Systems*, PP, n. 99, p. 1–8, 2017. ISSN 1524-9050. Citado na página 25.
- YENIKAYA, S.; YENIKAYA, G.; DÜVEN, E. Keeping the vehicle on the road: A survey on on-road lane detection systems. *ACM Computing Surveys (CSUR)*, ACM, v. 46, n. 1, p. 2, 2013. Citado na página 30.
- ZIEGLER, J. et al. Video based localization for bertha. In: *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. [S.l.: s.n.], 2014. p. 1231–1238. Citado 6 vezes nas páginas 12, 24, 26, 33, 35 e 120.