Universidade Federal de Minas Gerais (UFMG)
Engenharia de Controle e Automação
Projeto Final de Curso

Desenvolvimento de um Sistema de Controle para um Veículo Autônomo

Vitor Brandão Sabbagh

Belo Horizonte
Novembro de 2009

UNIVERSIDADE FEDERAL DE MINAS GERAIS (UFMG)
ENGENHARIA DE CONTROLE E AUTOMAÇÃO
PROJETO FINAL DE CURSO

Desenvolvimento de um Sistema de Controle para um Veículo Autônomo

Vitor Brandão Sabbagh

Monografia relativa ao Projeto Final de Curso, apresentado ao Colegiado do Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Minas Gerais, como requisito para obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Guilherme Augusto Silva Pereira / DEE - UFMG
Supervisor: Prof. Paulo H. Iscold A. de Oliveira / CEA - UFMG

Belo Horizonte
Novembro de 2009

Resumo

Palavras chave: Veículos Autônomos, Robótica, Controle de Direção.

Este trabalho apresenta o desenvolvimento de um sistema de controle de direção para um veículo autônomo. Desde 2007 vem sendo desenvolvido pelo Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos da Universidade Federal de Minas Gerais o CADU (Carro Autônomo Desenvolvido na UFMG), que é um veículo autônomo baseado em um automóvel de passeio. Tornar este carro autônomo consiste em prover ao mesmo a capacidade de percorrer determinadas trajetórias sem a presença de um condutor.

Atualmente o CADU conta com mecanismos de controle automático de frenagem, aceleração, câmbio e direção. Foi desenvolvido também um sistema de fusão sensorial que provê informações sobre a posição e atitude do carro a partir de quatro sistemas sensoriais: GPS, IMU (Unidade de Medidas Inerciais), medição do ângulo do volante e medição da velocidade das rodas dianteiras.

Neste projeto de fim de curso foi desenvolvido um software que integra todos os sistemas de controle e de sensoriamento. Além disso, foi desenvolvido um módulo que controla a posição e atitude do veículo. Para a validação do sistema foi realizado um experimento onde o carro percorre um conjunto determinado de coordenadas geográficas sem nenhuma intervenção humana.

Abstract

Keywords: Autonomous Vehicles, Robotics, Steering Control.

This work presents the development of a steering control system for an autonomous vehicle. The PDVA, a group for Research and Development of Autonomous Vehicles, in the Universidade Federal de Minas Gerais, has been developing the Cadu, an autonomous vehicle based on a conventional car. Making this car autonomous is providing it with capability to follow a determined trajectory without a human driver.

Cadu has mechanisms for automatic braking, acceleration, gear shifting and steering control. It was also developed a sensor fusion system that provides information about position e orientation of the car, taking into account data from four sensor systems: GPS, IMU, steering wheel angle and velocity of the front wheels.

In this project, it was developed a software that integrates control and sensor systems. Furthermore, it was developed a closed-loop control module that controls the vehicle's position and orientation. Afterwards, the system is validated in field tests, where the car drives through determined geographic coordinates without any human intervention.

Agradecimentos

Considero este projeto de pesquisa e desenvolvimento uma grande realização. Foi necessário muito trabalho e conhecimento de um grupo de pessoas que há algum tempo queria ver o Cadu andando autonomamente. Por isso agradeço aos colegas do Laboratório de Sistemas de Computação e Robótica (CORO) e do Centro de Estudos Aeronáuticos (CEA). Em especial aos colegas que muito contribuíram e apoiaram, Elias Freitas, Guilherme Castro, engenheiro Tiago Mendonça, professora Michelle Mendes, engenheiro Danilo Alves e engenheiro Marco Aurelio Moreira.

Agradeço muito ao meu orientador professor Guilherme Pereira pelos ensinamentos e pela constante cobrança, potencializando o nível dos trabalhos. Ao supervisor do projeto, professor Paulo Iscold, por disponibilizar o material e o espaço para realizar o trabalho, e pelo suporte dado até mesmo em feriados. Obrigado também ao professor Leonardo Torres pela excelente didática ao explicar problemas chaves.

Finalmente, acima de tudo agradeço e dedico esta realização à minha família pelo apoio contínuo e aos meus amigos pela paciência frente a minha ausência em muitos finais de semana.

Sumário

Resumo **i**

Abstract **ii**

Agradecimentos **iii**

Sumário **v**

Lista de Figuras **viii**

Siglas e Abreviaturas **ix**

1 Introdução **1**
1.1 Motivação e Relevância do Projeto 2
1.2 Escopo e Objetivos do Projeto 4
1.3 Revisão Bibliográfica 4
1.4 Organização do Texto 7

2 Descrição do Processo **9**
2.1 Automóvel Astra 9
2.2 Sistemas de Atuação 10
2.2.1 Câmbio 10
2.2.2 Freio 11
2.2.3 Direção 12
2.2.4 Aceleração 12
2.3 Sistemas Sensoriais 12
2.3.1 GPS 13
2.3.2 IMU 13
2.3.3 Sensor de orientação das rodas dianteiras 14
2.3.4 Sensor de velocidade das rodas 15
2.3.5 Fusão Sensorial 15

3 Sistemas de Operação e Controle do Veículo **17**
3.1 Integração do sistema de operação dos atuadores 17
3.2 Modelagem do veículo 18
3.3 Projeto do Controlador de Posição e Atitude 20

3.4 Simulação em ambiente Matlab Simulink 21

4 Implementação e Validação do Sistema de Controle 30
4.1 *Software* de Operação 30
4.1.1 Classes 32
4.2 Validação do Controlador de Posição e Atitude 33

5 Conclusões Finais 40
5.1 Resultados 40
5.2 Propostas de continuidade 41

A Manual de Operação do Sistema 43

B Método da Seção Áurea 46

Referências 49

Lista de Figuras

1.1 Stanley, veículo autônomo vencedor do Desafio DARPA 2005. 5
1.2 Modelo cinemático do veículo. 6

2.1 Automóvel Chevrolet Astra no qual este trabalho foi desenvolvido 9
2.2 Automatização de um automóvel Chevrolet Astra - adaptada de [Santos et al., 2008]. 10
2.3 Atuador utilizado para acionar o câmbio do veículo, modelo MTrack01 da Warner. 11
2.4 Receptor GPS Garmin modelo GPS18. 13
2.5 IMU utilizado no Cadu, modelo 3DM-GX1 da fabricante Microstrain . . . 14
2.6 Controlador do sistema de acionamento do volante, modelo 24/5 da fabricante EPOS 14
2.7 Microcontrolador Microship PIC18F2550, responsável pela leitura dos sensores de velocidade das rodas 15
2.8 Diagrama de comunicação entre hardware e software do sistema sensorial 16

3.1 Arquitetura do hardware e software do sistema. 18
3.2 Modelo de Ackerman para um automóvel 18
3.3 Simulação do controlador com $v(t) = 2m/s$, $k_1 = 1$ e $k_2 = 3$. 21
3.4 Malha de controle da direção modelada em Matlab 22
3.5 Bloco de simulação do modelo cinemático de Ackerman para um automóvel. 23
3.6 Simulação do movimento do veículo para o controlador com parâmetros $k_1 = 1$ e $k_2 = 3$. A curva tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo. 24
3.7 Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com parâmetros $k_1 = 1$ e $k_2 = 3$. 24
3.8 Simulação do movimento do veículo para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$. 26

3.9 Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$. 26
3.10 Simulação do movimento do veículo para o controlador com previsão de 40 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$. Observa-se um adiantamento do veículo em relação ao caminho. 27
3.11 Simulação do movimento do veículo para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 5$ e $k_2 = 3$. 28
3.12 Simulação do movimento do veículo para o controlador deslocamento de 20 posições na busca pela tangente do caminho e parâmetros $k_1 = 2,1$ e $k_2 = 3$. 28
3.13 Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com previsão de 40 posições no cálculo da tangente do caminho e parâmetros $k_1 = 2,1$ e $k_2 = 3$. 29

4.1 Interface do software do operação do veículo. 31
4.2 Biblioteca criada para a operação dos sistemas atuadores. 32
4.3 Classe responsável pela interface com o sistema de localização do veículo. 33
4.4 Área da UFMG em que foi realizado o experimento. 34
4.5 Avaliação experimental do controlador com parâmetros $k_1 = 1$ e $k_2 = 3$, sem previsão no cálculo da tangente do caminho. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo. 34
4.6 Teste prático do controlador com parâmetros $k_1 = 2,1$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo. 35
4.7 Comportamento da variável manipulada ($\delta(t)$) em um experimento com parâmetros do controlador $k_1 = 2,1$ e $k_2 = 3$. 36
4.8 Avaliação experimental do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo. 36
4.9 Teste prático do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia o comportamento do veículo. 37
4.10 Erro de distância obtido no teste prático do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. 37
4.11 Comportamento do veículo em um experimento realizado em um caminho de 240m de extensão. Controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. 38
4.12 Erro de distância obtido em um caminho de 240m de extensão coberto por terra e grama. Controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. . . 39

A.1 Interface do Samba, programa principal de operação do veículo. 43

B.1 Método da Seção Áurea 47

Siglas e Abreviaturas

DARPA	Do inglês *Defense Advanced Research Projects Agency*
PDVA	Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos
UFMG	Universidade Federal de Minas Gerais
CORO	Laboratório de Computação e Robótica
CC,DC	Corrente Contínua, *Direct Current*
CAN	Do inglês *Controller-Area Network*
GPS	Do inglês *Global Positioning System*
USB	Do inglês *Universal Serial Bus*
IDE	Do inglês *Integrated Development Environment*
PWM	Do inglês *Pulse Width Modulation*
SW	Do inglês *Software*

CAPÍTULO 1

Introdução

O conceito de veículo autônomo é baseado em um conjunto de tecnologias cognitivas e de controle altamente automatizadas. Em termos práticos, é um veículo que pode se mover de um ponto a outro, sem a necessidade de um motorista.

O Projeto Final de Curso apresentado nesta monografia trata do projeto e desenvolvimento de um sistema de controle de direção para um carro autônomo. Para tanto, é feita a integração entre sistemas sensoriais e de controle. Ademais, é desenvolvido um módulo que controla a posição e atitude do veículo a partir do controle da orientação de suas rodas dianteiras.

A história dos carros autônomos teve início em 1977, no Japão, onde cientistas do *Tsukuba Mechanical Engineering Lab* construiram um carro que, seguindo marcas brancas na pista, alcançava 30 km/h [Gingichashvili, 2007]. No fim da década de 80, a *European Comission* (Comissão das Comunidades Européias) investiu uma enorme quantia no projeto *Eureka Prometheus Project*, dedicado ao desenvolvimento de carros autônomos.

Na década de 90, diversas universidades americanas, européias e asiáticas iniciaram pesquisas nessa área. Destacaram-se a Carnegie Mellon University, a University of Michigan e a Stanford University.

Há dois anos aconteceu nos Estados Unidos o DARPA Urban Challenge, uma competição de carros autônomos, onde os carros deveriam dirigir por 96Km numa área urbana, respeitando todas as leis de trânsito. Seis equipes atingiram este objetivo, o

que pode ser considerado um marco na história dos carros autônomos, e representa o atual estado da arte dessa tecnologia.

Por sua vez, o Brasil ainda é um lugar de poucos projetos de pesquisa nessa área. O Laboratório de Sistemas de Computação e Robótica (CORO), pertencente ao Departamento de Engenharia Elétrica da Universidade Federal de Minas Gerais (UFMG), em 2007, iniciou pesquisas na área de carros autônomos. O CORO é um laboratório de pesquisa e desenvolvimento que atua nas áreas de robótica, visão computacional, processamento de imagens digitais, sistemas integrados de hardware e software, sistemas a eventos discretos, instrumentação e controle por computador [CORO, 2009].

O Grupo de Pesquisa e Desenvolvimento de Veículos Autônomos (PDVA) da Universidade Federal de Minas Gerais constitui-se de um grupo de professores e alunos da referida instituição engajados no desenvolvimento de sistemas embarcados de instrumentação e controle para veículos aéreos e terrestres, tripulados ou autônomos.

Um dos principais trabalhos atualmente em desenvolvimento pelo grupo envolve a construção de um sistema de instrumentação e controle para prover completa autonomia à um veículo de passeio terrestre de modo que o mesmo possa percorrer determinadas trajetórias sem a presença de um condutor [Santos et al., 2008]. Neste projeto está envolvido o laboratório CORO da Universidade Federal de Minas Gerais, localizado no Campus Pampulha da referida instituição e associado ao PDVA.

1.1. Motivação e Relevância do Projeto

As principais motivações para a realização deste projeto são de longo prazo e estão relacionadas ao desenvolvimento de tecnologias de veículos autônomos encarado como uma futura realidade, considerando um nível já avançado de autonomia [Thrun, 2006]. Estas são:

- **Salvar vidas**: Atualmente, uma das maiores causas de morte no mundo são acidentes de trânsito. De acordo com um levantamento do Ministério da Saúde, morreram no Brasil em conseqüência desses acidentes 32.753 pessoas em 2002, e 35.753 pessoas em 2005 [Online, 2007]. Ao considerarmos o uso de carros autônomos, esses números seriam drasticamente reduzidos, tendo em vista que as principais causas dos acidentes estão relacionadas ao consumo excessivo de bebidas alcoólicas e alta velocidade.;
- **Reduzir gastos**: Em 2006, a internação de 123.061 vítimas de acidentes no SUS

(Sistema Único de Saúde) custaram R$ 118 milhões. A maioria das internações ocorreu por atropelamentos (41.517), seguida pelos acidentes com motociclistas (34.767). Seria possível reduzir significativamente estes gastos, visto que um carro autônomo dificilmente cometeria um atropelamento[Thrun, 2006].

- **Tornar as pessoas mais produtivas**: Nas grandes cidades as pessoas gastam em media 1,5 horas no trânsito diariamente. Considerando uma jornada de trabalho de 8 horas, o tempo gasto no trânsito representa 18% do horário de trabalho. Nesse tempo as pessoas poderiam dormir, trabalhar, enviar emails, enfim, realizar tarefas mais produtivas, uma vez que não teriam de prestar atenção na direção do carro.

- **Melhorar a qualidade de vida de pessoas de idade**: As mesmas não teriam que conduzir o veículo ou depender de transporte público ou privado para se deslocarem em uma cidade ou realizarem uma viagem.

- **Aumentar a capacidade de tráfego nas vias expressas e rodovias**: Em diversas metrópoles brasileiras, o tráfego de automóveis já excede a capacidade para a qual foram projetadas, resultando em trânsito lento nas vias principais. Segundo Thrun[Thrun, 2006], a utilização de veículos autônomos poderia aumentar a capacidade dessas vias em cerca de 100%, considerando que os mesmos utilizariam melhor o espaço da pista, mantendo menor distância dos carros vizinhos.

Considerando-se motivações de curto e médio prazo para o desenvolvimento deste trabalho, pode-se também destacar benefícios trazidos pelo mesmo em um contexto mais atual. Segundo [Silva, 2009], o desenvolvimento de veículos autônomos tem sido explorado em diversas universidades e empresas automobilísticas no mundo. Apesar da lenta convergência dos esforços realizados nesta área de pesquisa em direção a uma eventual produção em massa de veículos autônomos, os estudos e progressos realizados proporcionam avanços no setor automobilístico no que se refere, principalmente, a aspectos ergonômicos e segurança. Em [Santos et al., 2008], Santos e colaboradores prevêem ainda, como contribuição à segurança dos passageiros, a concepção de sistemas capazes de detectar situações de perigo e alertar o condutor. Também pode ser proporcionado ao motorista maior conforto, a partir do desenvolvimento de sistemas de localização do veículo facilitando a condução do mesmo, e também acessibilidade, por meio de soluções para usuários com necessidades especiais.

A formação de um Engenheiro de Controle e Automação pela Universidade Federal de Minas Gerais inclui a realização do chamado Projeto Final de Curso (PFC). Neste, o aluno deve engajar-se na realização de um trabalho que lhe permita aplicar conceitos, métodos e ferramentas estudados ao longo do curso. O desenvolvimento do sistema de

controle da direção do veículo em questão exige a aplicação de vários desses conhecimentos, dos quais pode-se destacar a utilização de instrumentação eletrônica e digital e o projeto, simulação e validação de sistemas de controle.

1.2. Escopo e Objetivos do Projeto

Desde 2007 vem sendo desenvolvido pelo PDVA o Cadu (Carro Autônomo Desenvolvido na UFMG), que é um veículo autônomo baseado em um automóvel de passeio. Para o controle da posição e orientação do Cadu, objetiva-se a implementação de leis de controle da direção do veículo sobre uma plataforma que permite a aquisição das informações sobre posição, velocidade e orientação do veículo, bem como o controle automático dos mecanismos de condução do mesmo. Neste contexto, os sistemas de controle e sensoriais devem ser analisados e integrados. É descrito um controlador para a direção, simulado, implementado no software de operação do veículo e posteriormente validado em testes práticos.

A realização desses objetivos é cumprida de maneira sequencial e organizada segundo as seguintes etapas:

1. Integração dos sistemas de controle do freio, direção, câmbio e aceleração, centralizando as interfaces de operação em um software de controle geral.
2. Implementação da interface de comunicação TCP entre o software de controle e o sistema de fusão de dados.
3. Projeto, simulação em ambiente Matlab/Simulink[1] e implementação do módulo de controle de direção para o veículo.
4. Validação do sistema em testes práticos de operação autônoma do veículo.

1.3. Revisão Bibliográfica

Em 2003, a Organização Central de Pesquisa e Desenvolvimento do Departamento de Defesa dos Estados Unidos (DARPA) anunciou o primeiro Desafio DARPA (*DARPA Grand Challenge*), visando ao desenvolvimento de um veículo robótico autônomo capaz de atravessar um terreno inóspito *off-road*. Com o evento, diversos grupos de pesquisas envolvendo empresas e universidades se formaram com o objetivo de desenvolver sistemas veiculares robustos e seguros para aplicações comerciais e militares

[1] Os softwares The MathWorks Matlab® e Simulink® foram utilizados ao longo do trabalho para a simulação do sistema.

[McBride et al., 2008]. O número de participantes a cada edição do desafio (2004, 2005 e 2007) aumentou, refletindo o crescente interesse e progresso na área de pesquisa e desenvolvimento de veículos autônomos.

Em 2005, a Universidade de Stanford venceu o segundo Desafio DARPA. O veículo Stanley [Thrun et al., 2006], mostrado na Figura 1.1, desenvolvido em parceria com as empresas Volkswagen of America, Mohr Davidow Ventures e Intel Research, conta com um sistema embarcado multi-processado interligado a uma rede de sensores e atuadores necessários à autonomia do veículo. Com o projeto, novas tecnologias foram criadas e outras expandidas. O veículo abriga um sistema de hardware personalizado capaz de acionar eletronicamente o acelerador e os freios. A rotação do volante é controlada por meio de um motor CC acoplado ao mesmo por uma corrente, e o chamamento entre as posições do câmbio (automático) é provido por um atuador linear robótico. Informações como velocidade individual das rodas e ângulo de direção são sensoriadas automaticamente e enviadas a um sistema de computação via rede CAN.

Figura 1.1: Stanley, veículo autônomo vencedor do Desafio DARPA 2005.

O controle de direção do Stanley tem como entradas a trajetória gerada pelo sistema de planejamento de caminho, a pose, a velocidade e o ângulo das rodas dianteiras em relação ao veículo. A saída são comandos de ângulo das rodas dianteiras à uma frequência de 20Hz. A função do controlador é prover um seguimento em malha fechada do caminho desejado, que foi determinado pelo planejamento de caminho.

A principal métrica de erro é o erro transversal de caminho, $e(t)$, que mede a distância lateral entre do centro das rodas dianteiras e o ponto mais próximo no caminho, conforme mostrado na Figura 1.2. A direção é comandada por uma lei de controle cujo resultado é um $e(t)$ que converge para zero.

O controlador de direção do Stanley é baseado em uma função de realimentação não-linear do erro de distância transversal, para a qual a convergência pode ser provada. Na ausência de erro de distância, as rodas dianteiras do Stanley possuem a mesma

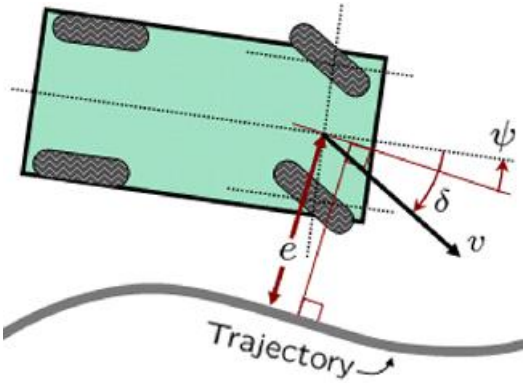

Figura 1.2: Modelo cinemático do veículo.

orientação que o caminho desejado. O ângulo ψ na Figura 1.2 descreve a orientação do veículo, medido relativamente à orientação do ponto mais próximo no caminho. Na ausência de erro de distância, as leis de controle fazem com que as rodas dianteiras estejam apontadas paralelamente à trajetória planejada.

A lei de controle da direção é dada basicamente por:

$$\delta(t) = \psi(t) + atan\frac{k.e(t)}{v(t)}, \tag{1.1}$$

onde k é o parâmetro de ganho e $v(t)$ é a velocidade do veículo. O segundo termo ajusta a direção proporcionalmente (não-linear) ao erro de distância $e(t)$: quanto maior esse erro, maior é a resposta da direção em direção à trajetória.

Usando o modelo linear do biciclo, considerando as rodas com rigidez infinita e limitações na direção, obtém-se o seguinte efeito da lei de controle:

$$\dot{x}(t) = -v(t).sen(atan\frac{k.e(t)}{v(t)}) = \frac{-k.e(t)}{\sqrt{1 + (\frac{k.e(t)}{v(t)})^2}}, \tag{1.2}$$

convergindo então para um pequeno erro de distância transversal,

$$e(t) = e(0).e^{-k.t}. \tag{1.3}$$

O erro converge portanto exponencialmente para $e(t) = 0$. O parâmetro k determina a taxa de convergência. À medida que o erro de distância aumenta, a função arco-tangente tende a virar as rodas dianteiras em direção à trajetória.

Outro participante do Desafio DARPA 2005 foi o Espírito de Berlim[Rojo et al., 2007],

um automóvel de passeio modificado para *Drive-by-Wire*, que utiliza tecnologias comerciais disponíveis para motoristas portadores de deficiência física. Para sua localização em relação à terra foi empregado uma combinação de GPS e IMU capaz de fornecer a posição do veículo com precisão de 1 metro ou melhor. O veículo ajusta a sua condução a partir de informações locais, a partir de sistema de visão computacional e lasers.

O hardware utilizado integra sensores e atuadores de qualidade industrial de custo moderado. O software possui grande robustez, pois o *framework* de sistemas robóticos checa continuamente a execução dos processos. Um algoritmo de fusão sensorial foi implementado. O controle de alto nível é realizado por um algoritmo de planejamento de caminho, que calcula um caminho ótimo, inclusive em torno de obstáculos. Componentes de controle reativos garantem que o carro possa reagir seguramente à situação inesperadas. Um simulador permite o teste e desenvolvimento de controladores adaptativos a partir de métodos de aprendizado.

Para o controle da direção deste veículo, foi utilizado um controlador PD, similar ao controlador utilizado pelo carro autônomo Stanley, abordado nesta seção. Dada uma trajetória, a distância das rodas dianteiras ao segmento mais próximo e o ângulo das rodas em relação ao referido segmento são determinados. A diferença entre os controladores implementados no Stanley e no Spirit of Berlim é um termo diferencial para a suavização da variável manipulada, evitando oscilação devido ao atraso entre o envio do comando e o alcance do ângulo das rodas desejado:

$$\delta(i) = \psi(i) + atan(\frac{k_1.e(i)}{v(i) + k_2}) - k_3(\delta(i) - \delta(i-1)). \tag{1.4}$$

O controlador de direção apresentado neste trabalho é uma modificação dos dois controladores citados nesta seção. Em comparação ao veículo Spirit of Berlim, são utilizados igualmente os ganhos k_1 e k_2. Em contrapartida, o termo diferencial (segunda parcela da Equação (1.4)) não é utilizado, igualmente ao controlador do veículo Stanley (Equação (1.1)).

1.4. Organização do Texto

Esta monografia está organizada em 5 capítulos. O presente capítulo apresenta os objetivos, motivação e relevância do Projeto Final de Curso em questão. É também apresentada uma revisão bibliográfica apontando o atual estado da arte do tema abordado. O Capítulo 2 apresenta o processo para o qual é desenvolvido o sistema de controle proposto, ou seja, o automóvel Chevrolet Astra 2003/2004, os sistemas de acionamento do freio, direção, câmbio e aceleração e os sistemas sensoriais.

O Capítulo 3 descreve inicialmente os a integração dos sistemas de acionamento. Em seguida é apresentado um modelo cinemático para o veículo, que é então validado via simulação e posteriormente aplicado ao projeto de um controlador de direção. O Capítulo 4 descreve o desenvolvimento do software de operação, a validação do mesmo em testes práticos em malha aberta e a configuração e validação do sistema de controle de direção, incluindo localização e controle de posição, em um experimento realizado com o veículo operando autonomamente. Por fim, como conclusões finais, avalia-se, no último capítulo, os resultados obtidos e são apresentadas sugestões para trabalhos futuros.

CAPÍTULO 2

Descrição do Processo

O Projeto Final de Curso em questão visa à implementação de um sistema de controle da direção do veículo de passeio Chevrolet Astra, nomeado Cadu. Este capítulo apresenta o processo em questão. Inicialmente, é descrito o atual estágio de automatização do veículo e discute-se a arquitetura dos sistemas de acionamento e sensoriais.

2.1. Automóvel Astra

O carro de passeio no no qual este trabalho foi desenvolvido é um Chevrolet Astra Sedan 2.0 retratado na Figura 2.1. O mesmo possui motor com 16 válvulas, 130 cavalos de potência, câmbio automático e direção eletro-hidráulica.

Figura 2.1: Automóvel Chevrolet Astra no qual este trabalho foi desenvolvido

Desde o início do projeto de pesquisa em 2007, este veículo vem sendo equipado com a instrumentação necessária para a idealização do sistema cognitivo e de controle do mesmo. Atualmente, acredita-se que ele já possui instrumentação suficiente para

prover ao veículo um nível intermediário de autonomia, tal que seria possível trafegar em ruas vazias, onde não haja trânsito de veículos ou pessoas.

A seções a seguir descrevem os sistemas de controle e sensoriais presentes no veículo. Os referidos sistemas foram integrados, possibilitando o seguimento de trajetórias em malha fechada, sendo a localização do veículo provida por uma fusão dos sistemas sensoriais e o controle de posição e orientação executada pela atuação dos sistemas de controle.

2.2. Sistemas de Atuação

Foram desenvolvidos para os quatro mecanismos de condução básicos do veículo sistemas de acionamento que permitem a operação dos mesmos por software. A direção, o câmbio, o acelerador e o freio do veículo possuem acoplados a sí atuadores eletromecânicos ou eletrônicos, como mostrado na Figura 2.2. Estes atuadores são controlados por microcontroladores embarcados no veículo, que por sua vez, permitem comunicação serial em tempo-real com um computador, por meio de portas USB.

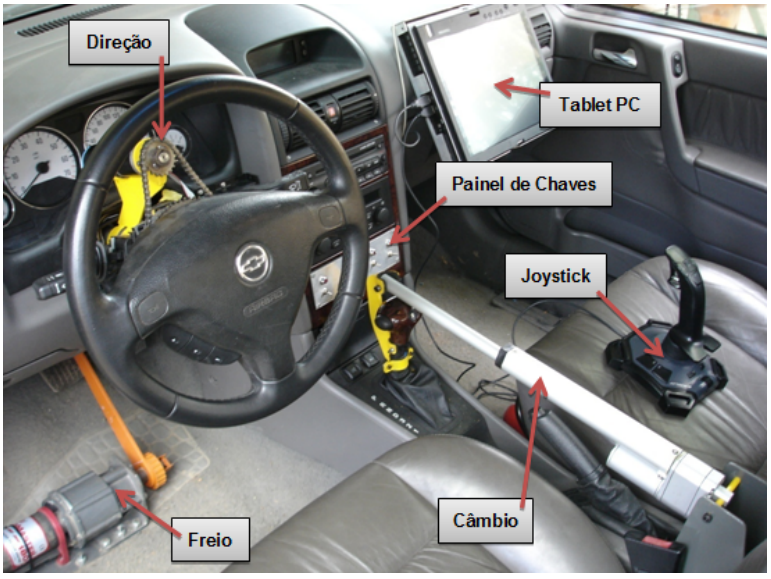

Figura 2.2: Automatização de um automóvel Chevrolet Astra - adaptada de [Santos et al., 2008].

2.2.1. Câmbio

O veículo Astra, sujeito deste trabalho, possui câmbio automático. Isso simplifica muito a automatização deste mecanismo de duas formas: 1) por não possuir embreagem, é dispensado um acionamento para este mecanismo; 2) não é necessária a mudança

manual entre a primeira e a quinta marcha. No caso de um câmbio automático, um acionamento eletro-mecânico realiza o controle da marcha de forma transparente ao usuário, baseado na frequência de rotação do motor [Nice, 2000].

O resultado disso é que a única atuação necessária no câmbio é uma atuação linear entre os modos *Park* (carro estacionado), *Reverse* (marcha ré), *Neutral* (neutro, ponto morto) e *Drive*. Para isso, foi empregado o atuador linear modelo MTrack01 (Figura 2.3) da fabricante Warner. O mesmo possui capacidade de carga de até 25 libras (aproximadamente 110 Newtons), velocidade mínima de 1,75 polegada/segundo, tensão de alimentação de 12V e extensão de atuação de 8 polegadas (aproximadamente 20cm). Possui um potênciometro em que a resistência é proporcional à posição do atuador, servindo como realimentação para o controle da posição do mesmo, e também chaves de fim de curso que cortam a alimentação do motor ao atingir as extensões limites.

Figura 2.3: Atuador utilizado para acionar o câmbio do veículo, modelo MTrack01 da Warner.

Este atuador é acionado por uma ponte-H, que por sua vez é alimentada pela bateria do veículo e controlada por um microcontrolador PIC18F2550, o que permite comunicação serial com um computador em tempo-real. Foi desenvolvido um protocolo de comunicação próprio para a operação do acionamento. Este protocolo foi então implementado em uma interface de comunicação no software de operação do veículo, podendo o câmbio ser comandado paralelamente aos outros sistemas de atuação.

2.2.2. Freio

Foi desenvolvido pelo aluno Tiago Mendonça como Projeto de Final de Curso um sistema de controle automático do mecanismo de frenagem do Cadu[Silva, 2009]. Um motor de corrente contínua, acoplado à uma caixa de redução e a uma alavanca, é responsável por transmitir ao pedal do freio o torque necessário à desaceleração do veículo. Um controlador de torque foi implementado em um microcontrolador digital e um circuito de potência, configurado como ponte-H, entregando ao motor, por meio da modulação PWM da tensão aplicada ao mesmo, a corrente de armadura necessária à

produção de um dado torque. Tal dispositivo fornece ainda uma medida desta corrente permitindo a realimentação da malha de controle.

Este sistema possui uma interface de comunicação serial com um computador mestre, pela qual o freio é operado neste trabalho. O sistema também possui uma interface com o sistema de emergência do veículo.

2.2.3. Direção

Para o controle da direção das rodas dianteiras do veículo, foi utilizado um motor de corrente contínua Maxon RE40 acoplado a uma caixa de redução. Foram acopladas ao volante e ao motor coroas dentadas conectadas por uma corrente de redução 2:1, como mostrado na Figura 2.2.

Este motor é controlador pelo controlador digital EPOS 24/5, que é capaz de controlar a posição, velocidade e corrente do motor. A comunicação é feita serialmente (RS-232). O acionamento requer alimentação de 11 a 70 volts e suporta corrente de até 5 amperes continuamente.

2.2.4. Aceleração

A aceleração do veículo, por sua vez, é obtida mediante atuação direta sobre o sistema eletrônico *drive-by-wire*, interno ao veículo. Este sistema é controlado por uma Unidade de Comando Eletrônico (UCE) responsável pelo avanço da ignição, controle da mistura ar/combustível e ventilador de arrefecimento. Em modo automático, um microcontrolador PIC18F2550 envia à UCE um sinal PWM correspondente à aceleração desejada. Em modo manual, o controle de aceleração é devolvido ao veículo, sendo o sinal elétrico referente à aceleração desejada traduzido por um sensor de posição do pedal do acelerador.

2.3. Sistemas Sensoriais

O sistema de localização do Cadu foi desenvolvido como projeto de mestrado pela engenheira Michelle Mendes[**?**], e consiste na fusão das informações fornecidas por um conjunto de sensores, seguido de uma filtragem de Kalman.

As informações de posição e atitude do veículo são obtidas a partir da fusão dos dados fornecidos por quatro sistemas sensoriais distintos: GPS, IMU, sensor de velocidade das rodas e sensor de posição do volante, descritos a seguir:

2.3.1. GPS

O principal sensor utilizado para a localização é o GPS, um sistema de posicionamento global baseado em tecnologia de satélites. A técnica fundamental do GPS é medir a distância entre o receptor e alguns satélites observados simultaneamente. As posições dos satélites são previstas e emitidas juntamente ao sinal que é emitido ao usuário. A partir de diversas posições conhecidas (dos satélites) e as distâncias medidas entre o satélite e o usuário, a posição do receptor pode ser determinada. A variação da posição, que também pode ser determinada, é então a velocidade do receptor. As aplicações mais importantes do GPS são posicionamento e navegação [Xu, 2007].

O Cadu é equipado com um GPS modelo GPS 18 PC [Garmin, 2005], da fabricante Garmin. Esse instrumento fornece informações a uma taxa de amostragem de 1Hz, com uma incerteza de 15 metros. O receptor disponibiliza as medidas por meio de uma saída serial sob o protocolo de comunicação RS-232. A essa saída serial é conectado um conversor USB/Serial que possibilita que o sensor seja conectado a uma porta USB no computador portátil[Santos, 2009].

Figura 2.4: Receptor GPS Garmin modelo GPS18.

2.3.2. IMU

O Sistema de Medições Inerciais (SMI ou IMU, do inglês *Inertial Measurement System*) utilizado neste trabalho, mostrado na Figura 2.5, é um instrumento eletrônico que informa medidas de velocidade angular, orientação e aceleração linear utilizando uma combinação de acelerômetros e girômetros. SMIs são tipicamente usados para assistir na pilotagem de aeronaves.

O instrumento utilizado no Cadu combina três girômetros com três acelerômetros ortogonais, três magnetômetros ortogonais, um multiplexador, um conversor AD de 16 bits e um microcontrolador embarcado para comunicar serialmente as medidas a uma

frequência de ate 350Hz [MicroStrain, 2006].

Figura 2.5: IMU utilizado no Cadu, modelo 3DM-GX1 da fabricante Microstrain

2.3.3. Sensor de orientação das rodas dianteiras

A orientação das rodas dianteiras do veículo não é medida diretamente nas rodas, mas a partir do ângulo do volante. Foi considerado que o mesmo possui um ângulo de giro linearmente proporcional ao ângulo das rodas. A medida deste ângulo é fornecida pelo mesmo equipamento utilizado para o acionamento do volante, o controlador EPOS 25/5, mostrado na Figura 2.6. Um *encoder* é responsável por medir o deslocamento angular do volante a partir do momento que o equipamento é ligado. A comunicação entre este dispositivo e o computador é feita serialmente. Um conjunto de funções foi fornecida em uma biblioteca DLL pelo fabricante.

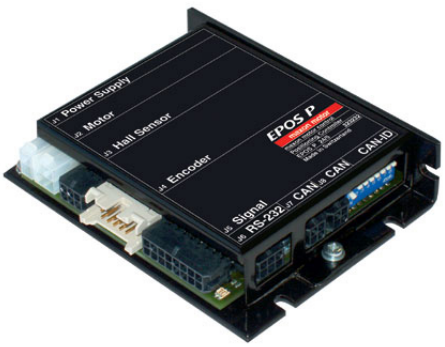

Figura 2.6: Controlador do sistema de acionamento do volante, modelo 24/5 da fabricante EPOS

2.3.4. Sensor de velocidade das rodas

A velocidade das rodas dianteiras do veículo é medida em um sensor já existente no sistema de freios ABS [1] no veículo. Este sensor fornece um sinal em uma frequência proporcional a velocidade angular da roda. Foi desenvolvido um circuito que realiza a conversão do sinal do sensor para um sinal de tensão proporcional à frequência.

O sinal de tensão fornecido pelo circuito de conversão é então lido por um microcontrolador PIC18F2550 (Figura 2.7), que por sua vez se comunica serialmente com o computador responsável pela localização do veículo.

Figura 2.7: Microcontrolador Microship PIC18F2550, responsável pela leitura dos sensores de velocidade das rodas

2.3.5. Fusão Sensorial

Foi desenvolvido pela mestranda Michelle Mendes o sistema de fusão sensorial baseada em filtragem de Kalman modificada [Santos, 2009].

No software responsável pela fusão sensorial foi implementado um servidor UDP, capaz de receber mensagens de diversos clientes UDP. Para cada um dos sistemas sensoriais, foi desenvolvido um software cujo papel é receber a partir de uma conexão serial as medidas dos sensores. Estes softwares atuam como clientes UDP, enviando simultaneamente as informações recebidas dos sensores para o servidor UDP, onde é executado o algoritmo de fusão sensorial, conforme mostrado na Figura 2.8.

Conhecidos os recursos e ferramentas disponíveis para a automação e controle do Cadu, apresentados no presente capítulo, segue-se a modelagem do veículo e o projeto do sistema de controle desejado, descritos no capítulo seguinte.

[1] Do inglês, Antilock Braking System

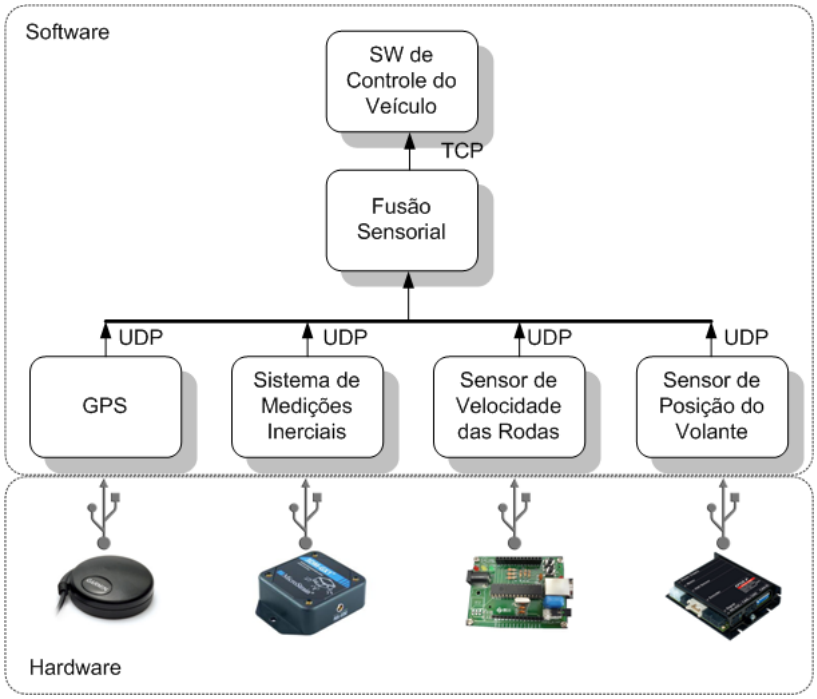

Figura 2.8: Diagrama de comunicação entre hardware e software do sistema sensorial

CAPÍTULO
3

Sistemas de Operação e Controle do Veículo

Este capítulo apresenta, inicialmente, uma discussão acerca da integração entre os sistemas atuadores e destes com o sistema de localização. Em seguida é proposto um modelo cinemático para o movimento do veículo. A partir deste modelo é apresentado um controlador para a direção. Por fim, são apresentadas simulações do processo em ambiente Matlab/Simulink.

3.1. Integração do sistema de operação dos atuadores

A condução automática do veiculo depende de um sistema capaz de operar simultaneamente todos os atuadores do veículo. Para tanto, fez-se necessário o desenvolvimento de um software único com o poder de controlar os quatro sistemas de atuação.

Conforme mostrado na Figura 3.1, o programa possui no total seis interfaces, sendo quatro interfaces de comunicação serial com os sistemas atuadores, uma de comunicação TCP com o sistema de localização e uma interface gráfica de comunicação com o usuário.

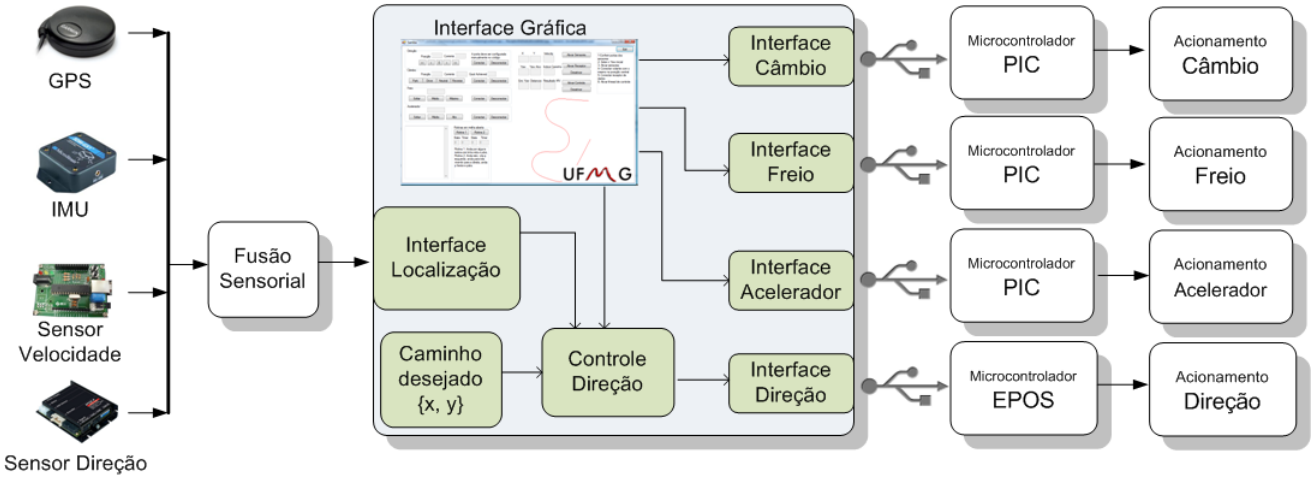

Figura 3.1: Arquitetura do hardware e software do sistema.

3.2. Modelagem do veículo

Esta seção descreve um modelo para o movimento do veículo. O modelo utilizado é o modelo de Ackerman, descrito em [Choset et al., 2005], onde o veículo é representado por um volume retangular sobre quatro rodas, conforme a Figura 3.2. As rodas traseiras são fixas e as rodas dianteiras são atuadas por meio do volante. A tração do veículo é dianteira e as velocidades das rodas dianteiras é conhecida por meio de sensores. É assumido que o veículo possui inércia desprezível, permitindo o projeto de um controlador de direção globalmente estável. Segundo Lynch, esta suposição é eficaz para o veículo em baixas velocidades [Choset et al., 2005].

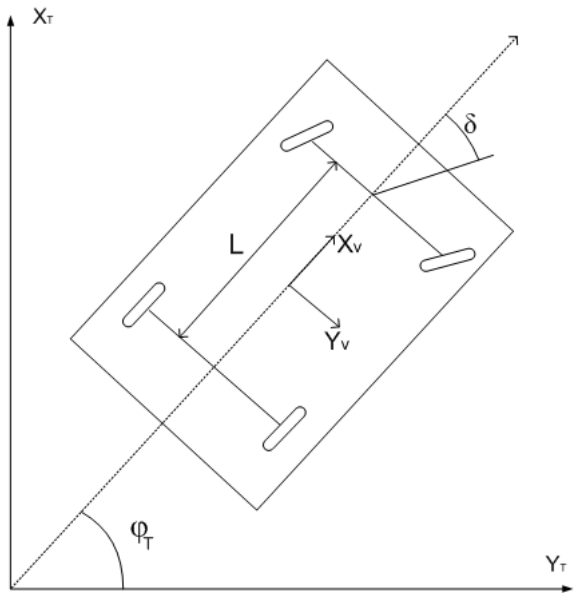

Figura 3.2: Modelo de Ackerman para um automóvel

No modelo de Ackerman, o ângulo da roda $\delta(t)$ é representado por uma roda virtual localizada entre as duas rodas dianteiras e o ângulo dessa roda virtual é dado pela média dos ângulos das duas rodas (que são diferentes entre si). A velocidade dessa roda virtual ($v(t)$) também é dada pela média das velocidades das duas rodas. A distância entre os eixos L deve ser considerada na estimação da orientação do veículo [1].

Os estados do modelo são as posições $x(t)$ e $y(t)$, a velocidade linear $v(t)$ e a orientação absoluta do veículo ψ_T no plano horizontal. A velocidade das rodas dianteiras $v(t)$ e o ângulo das rodas dianteiras $\delta(t)$(em relação ao veículo) são utilizados como entradas do modelo. Assim, os estados deste modelo, considerando o movimento em duas dimensões (apenas nos eixos X e Y) podem ser descritos como:

$$\dot{x}(t) = v(t).\cos(\psi(t)).\cos(\delta(t)), \tag{3.1}$$

$$\dot{y}(t) = v(t).\operatorname{sen}(\psi(t)).\cos(\delta(t)), \tag{3.2}$$

$$\dot{\psi}_T(t) = \frac{v(t)}{L}.\operatorname{sen}(\delta(t)). \tag{3.3}$$

Considerando um caminho definido sobre o referencial da terra composto por pontos espaçados de aproximadamente 0,05m, em um determinado ponto do caminho, sua orientação é dada por $\psi_C(t)$, equivalente ao ângulo absoluto formado pelo ponto em questão e o próximo ponto do caminho. Em outras palavras, $\psi_C(t)$ é a orientação absoluta do caminho em um determinado segmento (dois pontos do caminho consecutivos), sendo o mesmo o mais próximo da posição do veículo no tempo t. Foi definido então o ângulo de guinada do veículo em relação ao caminho $\psi(t)$, descrito por

$$\psi(t) = \psi_T(t) - \psi_C(t). \tag{3.4}$$

É possível descrever o modelo cinemático de um automóvel com velocidade $v(t)$ pelo erro de distância transversal $e(t)$ das rodas dianteiras e o ângulo dessas rodas em relação ao segmento mais próximo do caminho a ser seguido, $(\psi(t) - \delta(t))$, como mostrado na Figura 1.2. Essa estrutura de referência permite ao sistema atuar diretamente onde o erro é medido. Para a condução para frente, a derivada do erro de distância transversal é

$$\dot{e}(t) = v(t).\operatorname{sen}(\psi(t) - \delta(t)), \tag{3.5}$$

onde o ângulo das rodas dianteiras é mecanicamente limitado a $|\delta(t)| < 29.5°$. A deri-

[1] No veículo utilizado, a distância entre os eixos é de 2,61m.

vada do ângulo de guinada é

$$\dot{\psi}(t) = -\frac{v(t).\,\mathrm{sen}(\psi(t) - \delta(t))}{L}. \tag{3.6}$$

3.3. Projeto do Controlador de Posição e Atitude

Esta seção detalha as leis de controle da direção, que possuem como entradas as variáveis de estado do veículo e o caminho desejado e como saída o ângulo de giro do volante δ a uma taxa de 10Hz. O sistema de fusão sensorial fornece as variáveis de estado: localização $x(t)$ e $y(t)$, orientação $\psi(t)$ em relação ao referencial da terra e a velocidade do veículo.

Este é um controlador proporcional não-linear, sendo o mesmo uma adaptação de [Rojo et al., 2007] e [Hoffmann et al., 2007]. Dada uma trajetória (Figura 1.2), a distância $e(t)$ entre o ponto médio das rodas dianteiras e o seguimento da trajetória mais próximo e o ângulo $\psi(t)$ das rodas em relação ao referido segmento são determinados. A fórmula para computar a saída (o ângulo δ das rodas dianteiras) também inclui a velocidade do veículo e as constantes k_1 e k_2.

A equação para o controle da direção é

$$\delta(t) = \psi(t) + \mathrm{atan}(\frac{k_1.e(t)}{v(t) + k_2}). \tag{3.7}$$

A constante k_2 é incluída na expressão originalmente proposta por Hoffmann para obter-se melhor desempenho em baixa velocidade e para prevenir que o argumento da função arco-tangente se torne muito grande [Hoffmann et al., 2007]. O atraso entre o envio do comando e o alcance do ângulo de direção desejado pode causar oscilações, segundo [Rojo et al., 2007]. Uma saturação na taxa de variação de $\delta(t)$ é adicionada para evitar esta condição. Além disso, o ângulo das rodas $\delta(t)$ é limitado mecanicamente à $-\delta_{max} < \delta < \delta_{max}$. No caso do veículo Astra que foi utilizado, $\delta_{max} = 29,5°$. Considerando as duas restrições citadas, as leis de controle são

$$\delta(t) = \begin{cases} \psi(t) + \mathrm{atan}\,\frac{k_1.e(t)}{v(t)+k_2} \text{ se } \left|\delta(t) = \psi(t) + \mathrm{atan}\,\frac{k_1.e(t)}{v(t)+k_2}\right| < \delta_{max} \\ \delta_{max} \text{ se } \psi(t) + \mathrm{atan}\,\frac{k_1.e(t)}{v(t)+k_2} \geq \delta_{max} \\ -\delta_{max} \text{ se } \psi(t) + \mathrm{atan}\,\frac{k_1.e(t)}{v(t)+k_2} \leq -\delta_{max} \end{cases} \tag{3.8}$$

e

$$\dot{\delta}(t) = \begin{cases} \frac{\partial\psi(t)+\text{atan}\,\frac{k_1.e(t)}{v(t)+k_2}}{\partial t} \text{ se } \left|\frac{(\delta(t)-\delta(t-\triangle t))}{\triangle t}\right| < \dot{\delta}_{max} \\ \dot{\delta}_{max} \text{ se } \frac{(\delta(t)-\delta(t-\triangle t))}{\triangle t} \geq \dot{\delta}_{max} \\ -\dot{\delta}_{max} \text{ se } \frac{(\delta(t)-\delta(t-\triangle t))}{\triangle t} \leq -\dot{\delta}_{max}. \end{cases} \tag{3.9}$$

Para uma avaliação preliminar do controlador, o mesmo foi implementado em ambiente Matlab e teve seu resultado simulado para posições diversas no espaço próximo a um determinado caminho. Esta simulação está representada na Figura 3.3 em forma de um campo vetorial. Cada vetor no espaço representa a direção em que estariam apontadas as rodas dianteiras do veículo, caso o mesmo se encontrasse na posição marcada pela base do vetor.

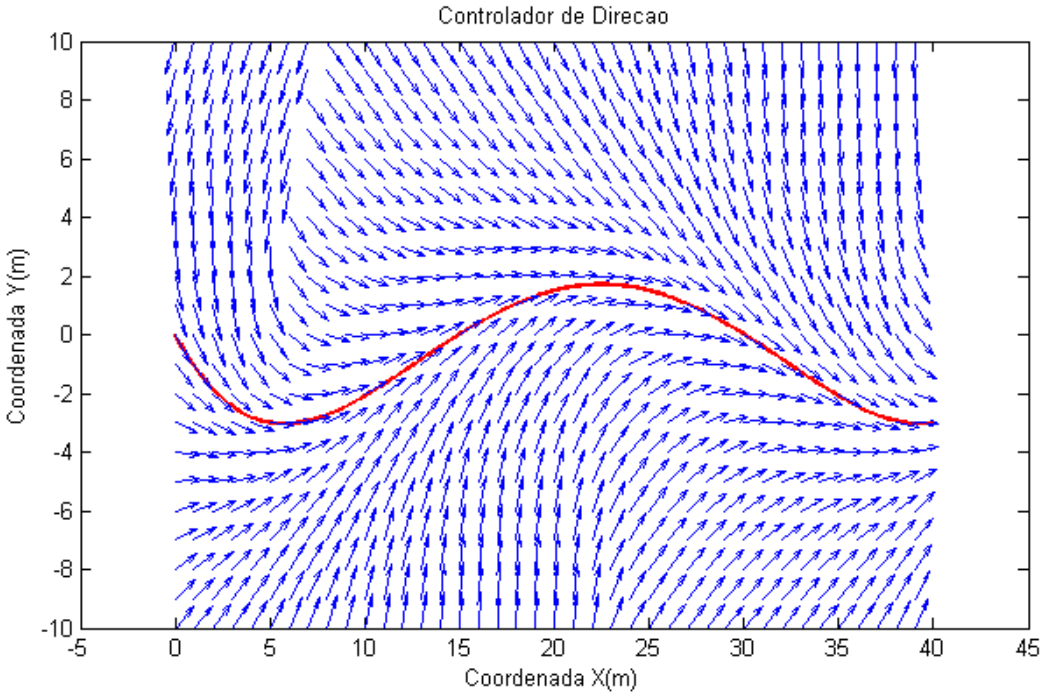

Figura 3.3: Simulação do controlador com $v(t) = 2m/s$, $k_1 = 1$ e $k_2 = 3$.

3.4. Simulação em ambiente Matlab Simulink

Para a validação do modelo e simulação do controlador proposto, os mesmos foram implementados em ambiente Matlab Simulink, como mostrado na Figura 3.4.

Considerando a entrada de velocidade $v(t)$, é utilizada na simulação um sinal semelhante ao que ocorre nos experimentos práticos. Primeiramente, o sinal de velocidade é

uma rampa de 0 à 3 m/s. Após alcançar este valor, a velocidade é mantida com a média de 3 m/s, adicionada de um ruído senoidal de amplitude 0,5 m/s. A velocidade oscila portanto entre 2,5 e 3,5 m/s.

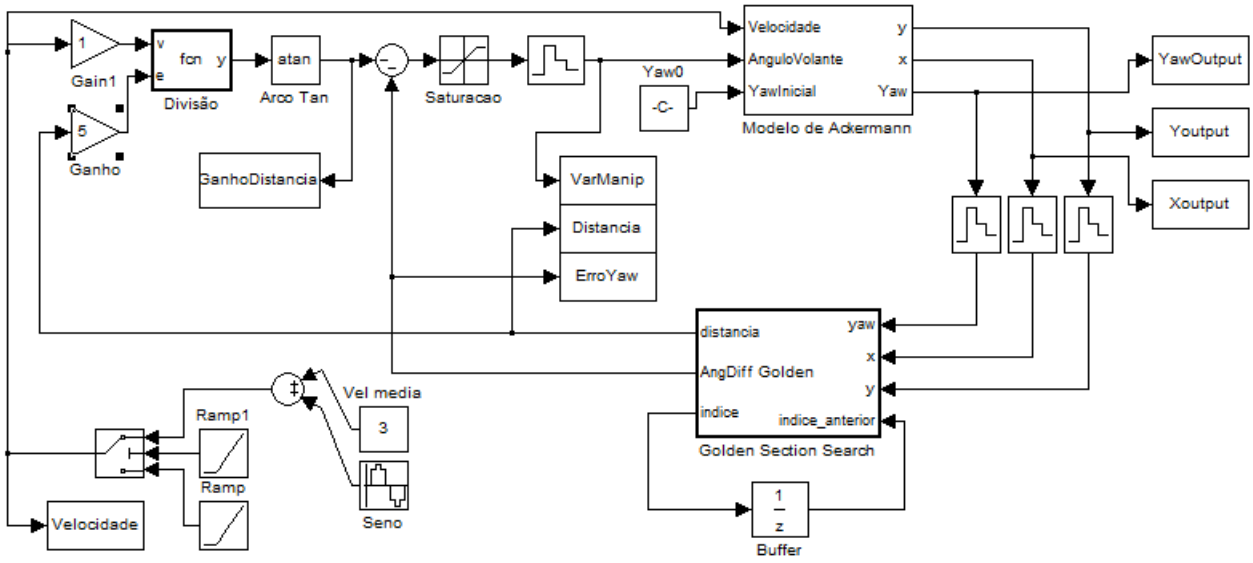

Figura 3.4: Malha de controle da direção modelada em Matlab

As variáveis de estado relativas a posição do veículo, $x(t)$ e $y(t)$, são inicialmente iguais à zero. O ângulo de guinada inicial do veículo em relação ao referencial da terra, referido na Figura 3.4 como Yaw0, foi inserido manualmente. Nas simulações feitas, o ângulo de guinada inicial foi igualado ao ângulo da tangente formada pelos dois primeiros pontos do caminho a ser seguido.

A atualização da posição do veículo, baseada no modelo de Ackerman, é realizada na simulação pelo bloco referido como Modelo de Ackerman. Este bloco é baseado nas Equações (3.1), (3.2) e (3.3) e está retratado na Figura 3.5.

O primeiro passo para o cálculo da variável manipulada desta malha de controle, o ângulo das rodas $\delta(t)$, consiste em comparar a posição atual do veículo ao caminho desejado. O segmento do caminho mais próximo à posição atual do veículo deve ser encontrado. O caminho é representado neste sistema por um conjunto de posições {C_x , C_y} que pode conter até 5000 pontos, dependendo da extensão do mesmo. Utilizar métodos convencionais para encontrar o segmento mais próximo da posição atual(x, y) seria computacionalmente caro. Foi utilizado então um método para busca de mínimo de uma função chamado Método da Seção Áurea. Este método é detalhado no Apêndice B, e está referido na Figura 3.4 pelo seu nome original em inglês, *Golden Section Search*.

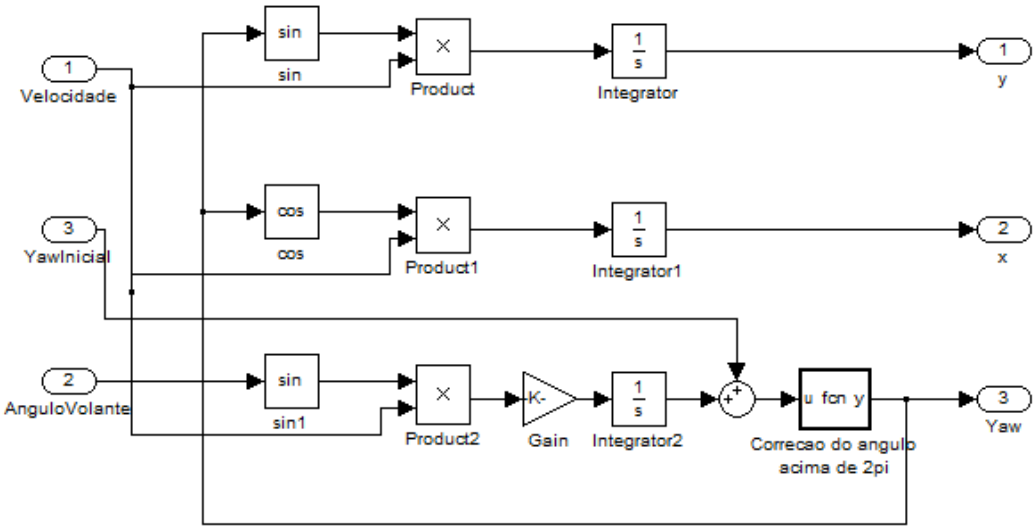

Figura 3.5: Bloco de simulação do modelo cinemático de Ackerman para um automóvel.

O intervalo da busca pelo segmento mais próximo ao longo do caminho $\{C_x , C_y\}$ é dinâmico. A cada iteração da malha de controle, este segmento é procurado em uma faixa do caminho, sendo a mesma determinada pelo segmento encontrado na iteração anterior. Isso permite que o caminho passe pela mesma posição no espaço duas ou mais vezes, isto é, partes do caminho podem se cruzar sem o risco de o veículo escolher o caminho errado no momento do cruzamento. Além disso, o método utilizado para a busca do segmento mais próximo requer que a função à ser minimizada seja monotônica, sendo portanto necessário haver somente um mínimo no intervalo. Pelo fato de a velocidade do veículo ser limitada e a frequência das iterações ser suficientemente alta, pode-se afirmar que o veículo nunca passará para um segmento do caminho distante daquele calculado na iteração anterior.

Na primeira simulação realizada, utilizou-se como parâmetros do controlador $k_1 = 1$ e $k_2 = 3$. O resultado está representado na Figura 3.6. O veículo parte da origem(0,0) com inclinação de -95°. Observou-se nas curvas um erro de distância transversal e de até 2 metros, como mostrado na Figura 3.7. Ao retornar à linha reta, foram necessários cerca de 4 segundos até o veículo se reaproximar do caminho.

Em todas as simulações apresentadas nesta seção, o caminho desejado é representado pela curva tracejada, e o comportamento do veículo é representado pela linha cheia.

No início da simulação mostrada na Figura 3.6, percebe-se um atraso entre o início da curva e a resposta do controlador, resultando no distanciamento do veículo.

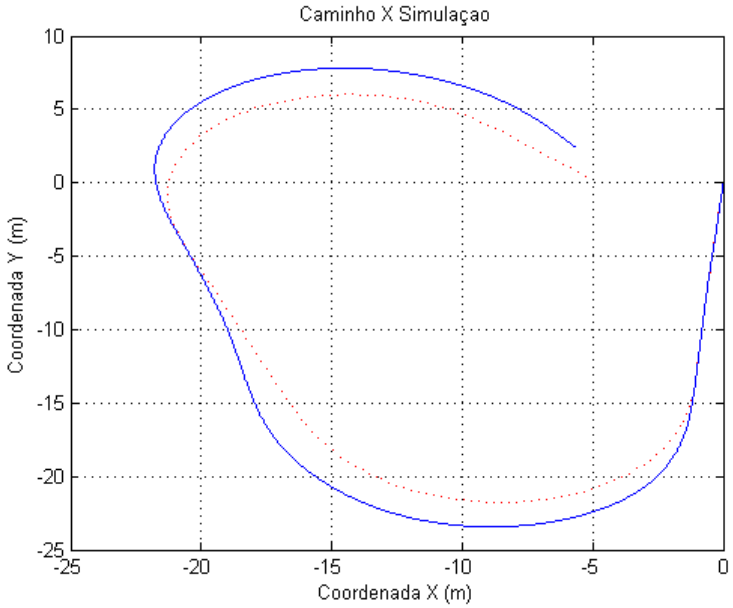

Figura 3.6: Simulação do movimento do veículo para o controlador com parâmetros $k_1 = 1$ e $k_2 = 3$. A curva tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo.

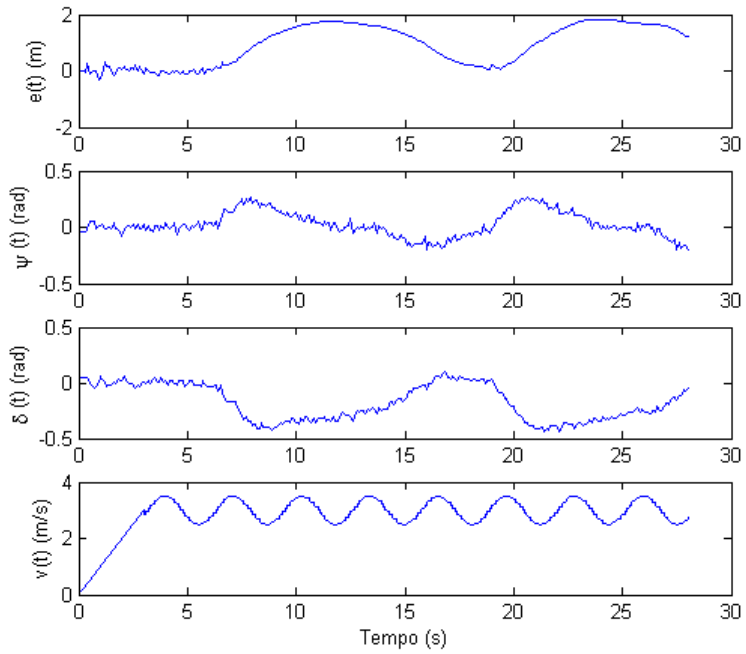

Figura 3.7: Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com parâmetros $k_1 = 1$ e $k_2 = 3$.

Acredita-se que a causa deste atraso seja devido a baixa frequência de execução do ciclo de controle, sendo a mesma aproximadamente 10Hz. Devido a limitações computacionais do equipamento utilizado nos experimentos (computador portátil EEE-PC, da fabricante Asus), foi necessário trabalhar na frequência mencionada. Foi proposta então uma mudança no algoritmo para tentar reduzir o atraso provocado pela referida condição.

O cálculo da variável manipulada (o ângulo das rodas) é feito a partir de duas componentes, uma proporcional à tangente do segmento mais próximo do caminho e outra proporcional (não-linearmente) à distância do veículo até o caminho (Equação 3.7). O atraso provocado pelo baixa frequência de execução das iterações, sendo o mesmo de aproximadamente 100ms, afeta a segunda componente da referida equação. Com o carro em movimento, ao ser calculado a tangente do segmento mais próximo do caminho, o veículo já se encontra em outra posição cuja tangente aponta para outra direção, caso o veículo esteja em uma curva. Sendo assim, o cálculo da componente proporcional à tangente do caminho foi modificado, de forma a considerar não mais o segmento mais próximo, mas sim alguns segmentos à frente. Em outras palavras, foi feita uma previsão do caminho para o cálculo da segunda componente da Equação de Controle 3.7.

Baseado na distância entre os pontos do caminho e no atraso do ciclo de controle, foi estimado uma previsão de 20 posições no índice do segmento. Isto significa um adiantamento espacial de aproximadamente 0,60 metros, visto que os pontos do caminho estão espaçados de aproximadamente 0,05m. Uma nova simulação foi feita, resultando em erros menores, como mostrado na Figura 3.8. A distância máxima neste caso foi de 0,57 metros (Figura 3.9) e a distância média 0,25m (RMS[2]).

Aplicando-se previsões maiores ao cálculo da tangente do caminho, observou-se um adiantamento do veículo em relação à curva, como mostrado na Figura 3.10. Neste caso foram utilizados novamente os ganhos $k_1 = 1$ e $k_2 = 3$ e uma previsão de 40 posições, equivalente a aproximadamente 1,2 metros.

Estabelecido uma previsão ótima de 20 posições, foram variados os outros parâmetros do controlador. Conforme explicado anteriormente, um dos papéis do parâmetro k_2 , que é adicionado à velocidade $v(t)$ no denominador da função arco-tangente, é prevenir que o argumento da função se torne muito grande. Como os experimentos foram realizados em baixa velocidade, de 0 à 3m/s, optou-se por manter o ganho k_2 com valor 3.

[2]Do inglês, Root Mean Square.

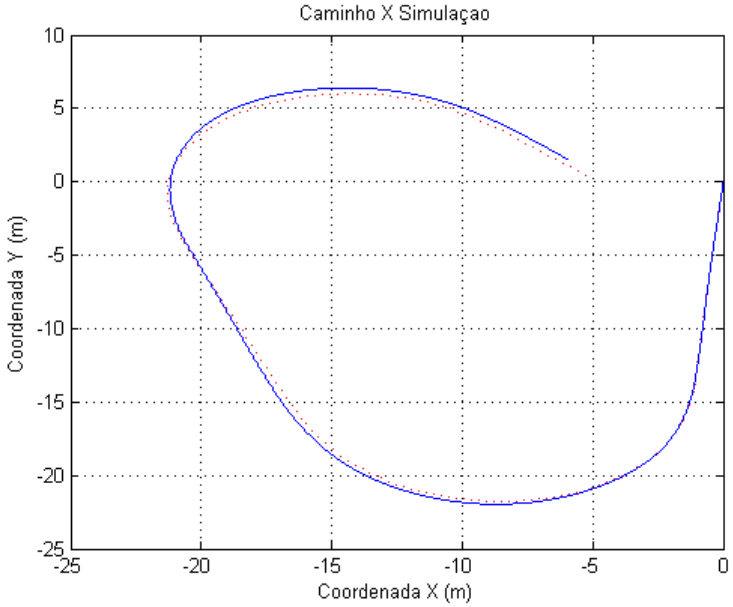

Figura 3.8: Simulação do movimento do veículo para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$.

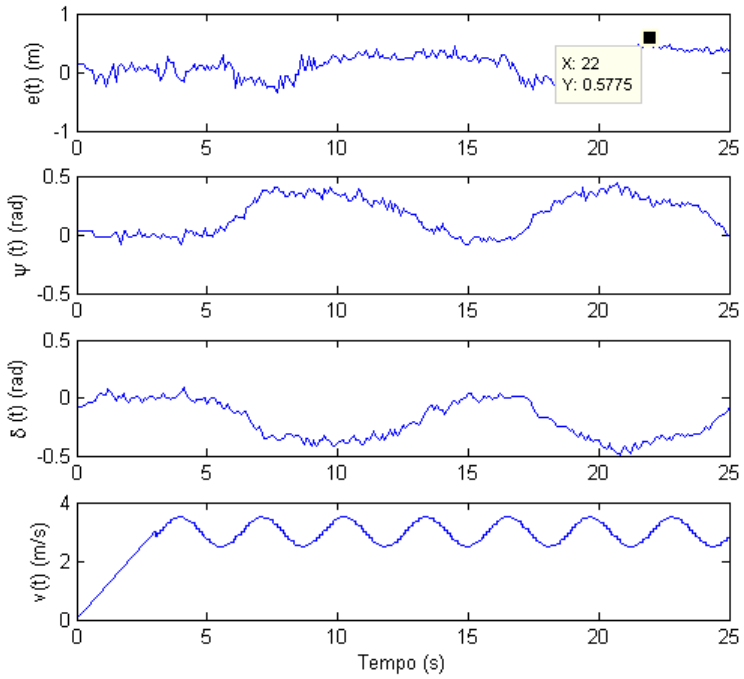

Figura 3.9: Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$.

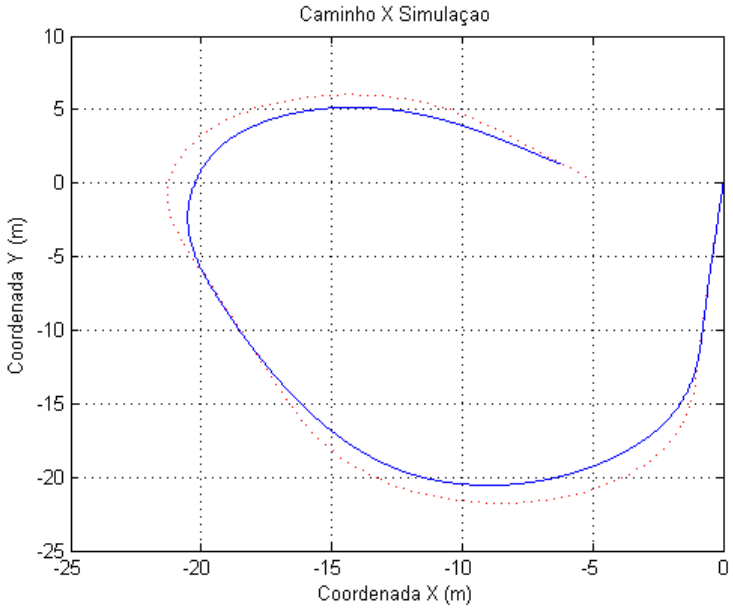

Figura 3.10: Simulação do movimento do veículo para o controlador com previsão de 40 posições no cálculo da tangente do caminho e parâmetros $k_1 = 1$ e $k_2 = 3$. Observa-se um adiantamento do veículo em relação ao caminho.

Foi simulado então o controlador com o ganho $k_1 = 5$, cujo resultado é mostrado na Figura 3.11. Nesta simulação percebe-se uma forte oscilação do veículo em relação ao caminho desejado, com um afastamento de até 3 metros do mesmo, conforme previsto por Hoffmann em [Hoffmann et al., 2007].

Outras configurações do controlador foram simuladas. Foi observado o melhor desempenho utilizando-se $k_1 = 2,1$, $k_2 = 3$ e uma previsão no cálculo da tangente do caminho de 20 posições. Percebe-se com esta configuração um distanciamento médio de 0,18m (RMS) e máximo de 0,39 metros do caminho, conforme mostrado nas Figuras 3.12 e 3.13.

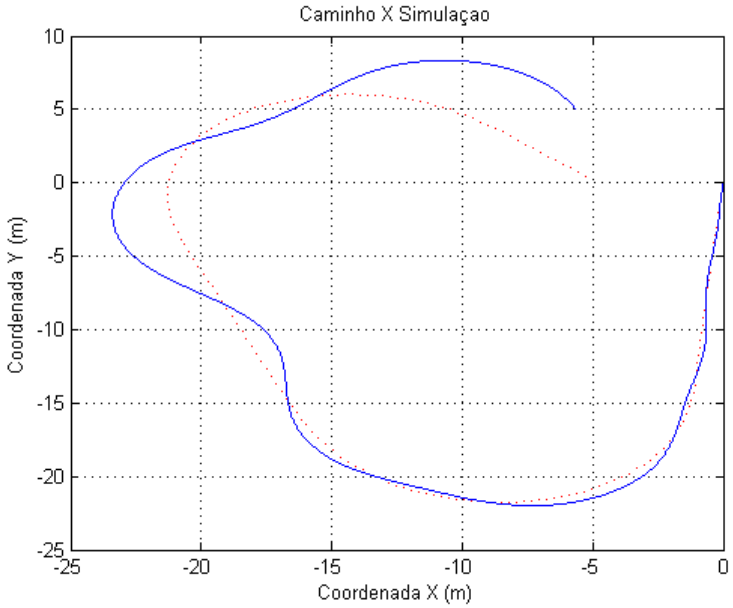

Figura 3.11: Simulação do movimento do veículo para o controlador com previsão de 20 posições no cálculo da tangente do caminho e parâmetros $k_1 = 5$ e $k_2 = 3$.

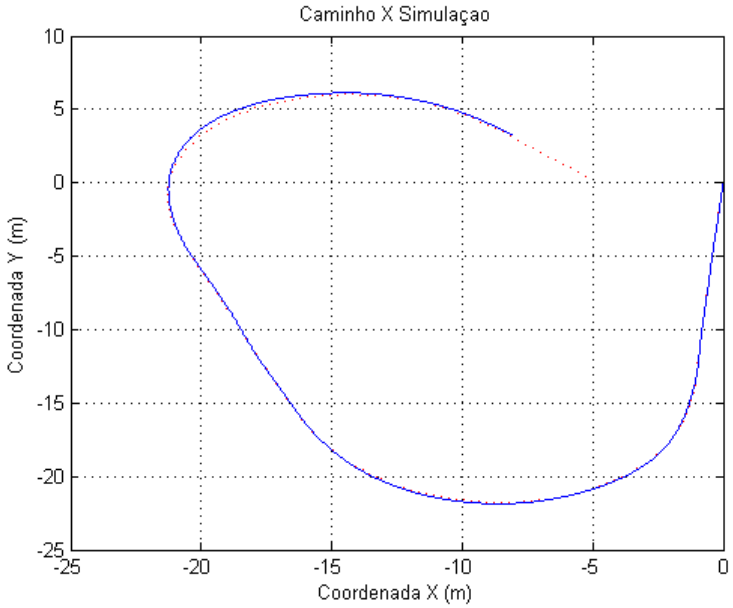

Figura 3.12: Simulação do movimento do veículo para o controlador deslocamento de 20 posições na busca pela tangente do caminho e parâmetros $k_1 = 2,1$ e $k_2 = 3$.

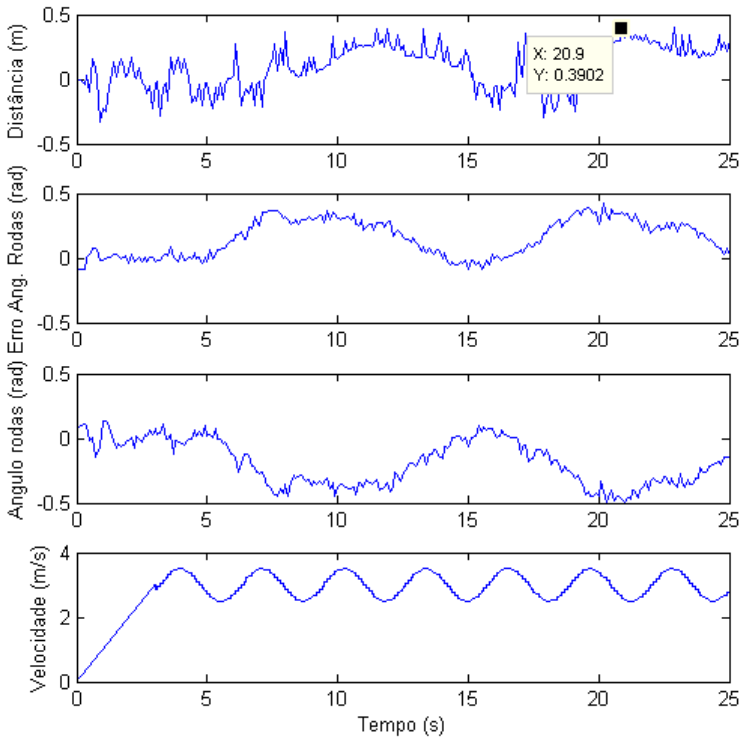

Figura 3.13: Simulação da distância de afastamento do caminho ($e(t)$), ângulo das rodas dianteiras ($\delta(t)$), erro do mesmo ângulo em relação ao caminho ($\psi(t)$) e velocidade do veículo ($v(t)$) para o controlador com previsão de 40 posições no cálculo da tangente do caminho e parâmetros $k_1 = 2,1$ e $k_2 = 3$.

CAPÍTULO 4

Implementação e Validação do Sistema de Controle

Para a integração dos sistemas atuadores e de localização, foi necessária a criação de um software que englobasse interfaces de comunicação para a operação dos sistemas atuadores e de localização e o módulo de controle da direção.

Neste capítulo, a arquitetura do software de operação do veículo é descrita, bem como as classes necessárias a seu adequado funcionamento. Em seguida, são apresentados os experimentos realizados para a validação do módulo de controle concebido, acompanhados de uma análise de desempenho.

4.1. *Software* de Operação

O *software* de operação foi desenvolvido utilizando-se a IDE Visual C++ 2008 Express Edition, da fabricante Microsoft. O software desenvolvido possui uma interface gráfica com botões para as diversas operações possíveis, como mostrado na Figura A.1. Cada um dos botões é conectado internamente a uma das classes responsáveis pela operação dos atuadores.

Estão presentes na parte central superior da interface gráfica mostradores das variáveis mais importantes envolvidas no processo. As variáveis de posição $x(t)$ e $y(t)$, orientação $\psi_T(t)$ e velocidade $v(t)$ são fornecidas pelo sistema de localização. A partir da combinação de informações de localização e o do caminho desejado, são calculadas

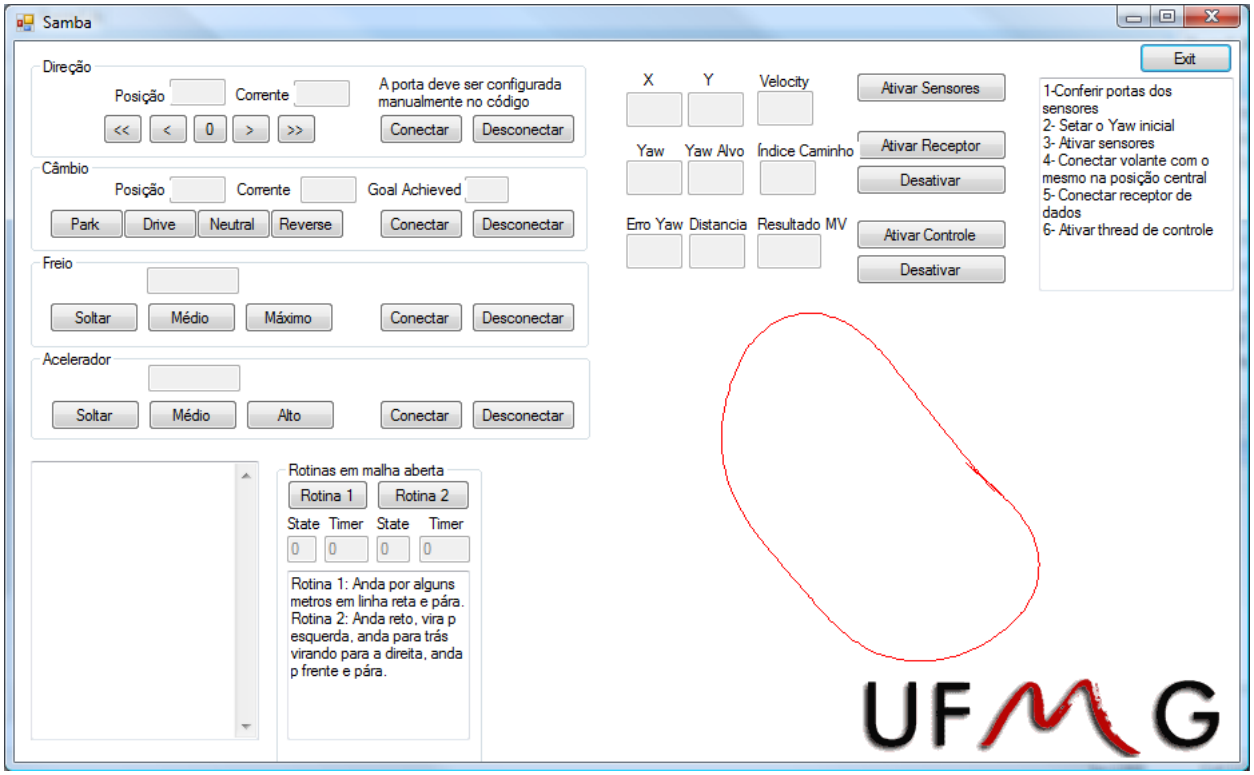

Figura 4.1: Interface do software do operação do veículo.

as outras variáveis mostradas pela interface gráfica: tangente do segmento do caminho mais próximo à posição atual do veículo, denominado Yaw Alvo, o índice do segmento do caminho em que o veículo se encontra, denominado Índice Caminho, o erro do ângulo de guinada do veículo em relação ao caminho, denominado Erro Yaw, e por fim a variável manipulada calculada pelo controlador, denominada Resultado MV[1] na Figura A.1.

É mostrado também na interface gráfica o caminho a ser seguido pelo veículo. O mesmo é inserido no programa a partir de um arquivo chamado *path.txt*. Este arquivo contém as coordenadas C_x e C_y em metros pelas quais o caminho é definido, considerando que o veículo parte sempre da posição (0;0) ao ser iniciado o sistema de localização.

No canto esquerdo inferior da interface, uma caixa de texto mostra informações sobre os sistemas atuadores e o status das conexões com os mesmos. No canto superior, uma segunda caixa de texto mostra instruções para a correta ativação dos sistemas atuadores e de localização.

[1]Do inglês, Manipulated Variable.

4.1.1. Classes

O software desenvolvido para a operação do veículo é baseado em uma biblioteca de classes, cada uma responsável por realizar a comunicação entre o software e os sistemas de atuação ou de localização.

Como mostrado na Figura 4.2, as classes denominadas Freio, Acelerador e Câmbio possuem como base a classe Módulo. Visto que nas três classes citadas são implementados protocolos de comunicação semelhantes com microcontroladores PIC do mesmo modelo, foi criado para as três uma classe base em comum. Nela estão implementados os métodos básicos para a comunicação com os três sistemas de atuação mencionados: Conectar(), Desconectar() e EstaConectado(). A classe base Módulo possui um objeto chamado InterfaceSerial, pertencente a classe CSerial. Está classe possui métodos para comunicação serial com dispositivos periféricos conectados ao computador principal via RS232[2].

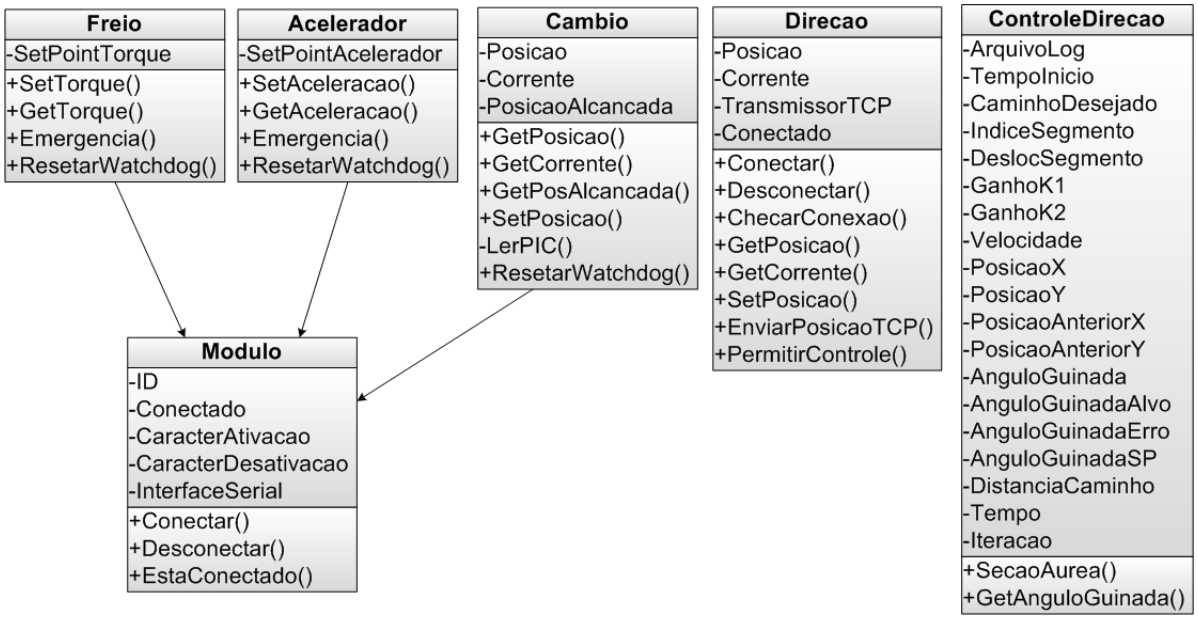

Figura 4.2: Biblioteca criada para a operação dos sistemas atuadores.

Além dos métodos implementados pela classe base, as classes Freio, Acelerador e Câmbio possuem métodos para o controle e leitura de suas respectivas variáveis manipuladas. Além disso, possuem métodos para emergência e reinicialização do temporizador *watchdog*[3].

[2]RS-232 é um padrão para troca serial de dados binários entre um DTE (terminal de dados, de Data Terminal equipment) e um DCE (comunicador de dados, de Data Communication equipment). É comumente usado nas portas seriais dos PCs.

[3]Watchdog é um temporizador que deve ser reinicializado a uma frequência determinada, comumente utilizado para identificar falha na comunicação entre dispositivos.

O acionamento da direção é feito a partir de um controlador EPOS. Para a comunicação com este dispositivo, um conjunto de funções foi fornecida em uma biblioteca DLL[4] pelo fabricante. Foi criada então uma classe, onde estão implementadas as funções básicas de operação deste atuador, listadas na Figura 4.2.

Para a interface TCP com o sistema de localização do veículo, foi criada a classe InterfaceLocalização, que possui os métodos básicos de comunicação, listados na Figura 4.3. Ao início do ciclo de controle, a função ReceberMensagemTCP() é executada, recebendo do *software* responsável pela fusão sensorial uma mensagem contendo os valores de $x(t)$, $y(t)$, $\psi_T(t)$ e $v(t)$.

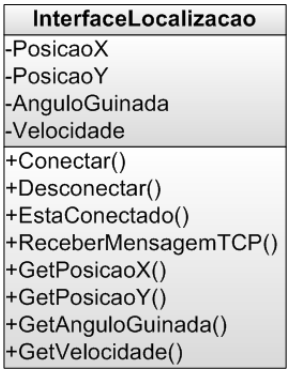

Figura 4.3: Classe responsável pela interface com o sistema de localização do veículo.

4.2. Validação do Controlador de Posição e Atitude

Para a validação do sistema de controle implementado, é avaliado a qualidade com a qual o veículo seguiu o caminho pré-definido. Os experimentos mostrados nesta seção foram realizados em um estacionamento localizado no campus da Universidade Federal de Minas Gerais, área mostrada na Figura 4.4.

Primeiramente foi realizado um experimento semelhante à primeira simulação descrita na Seção 3.4. A linha cheia na Figura 4.5 mostra o comportamento do veículo com o controlador com ganhos $k_1 = 1$ e $k_2 = 3$, sem previsão no cálculo da tangente do caminho. Percebe-se um atraso na resposta do veículo em relação ao início da curva, como foi previsto na simulação. O erro de distância do caminho foi ainda maior do que na simulação. Acredita-se que isso se deve a um atraso por parte do sistema de localização ainda maior do que aquele implementado na simulação.

[4]Do inglês, Dinamic-link Library, é a implementação feita pela Microsoft para o compartilhamento de bibliotecas

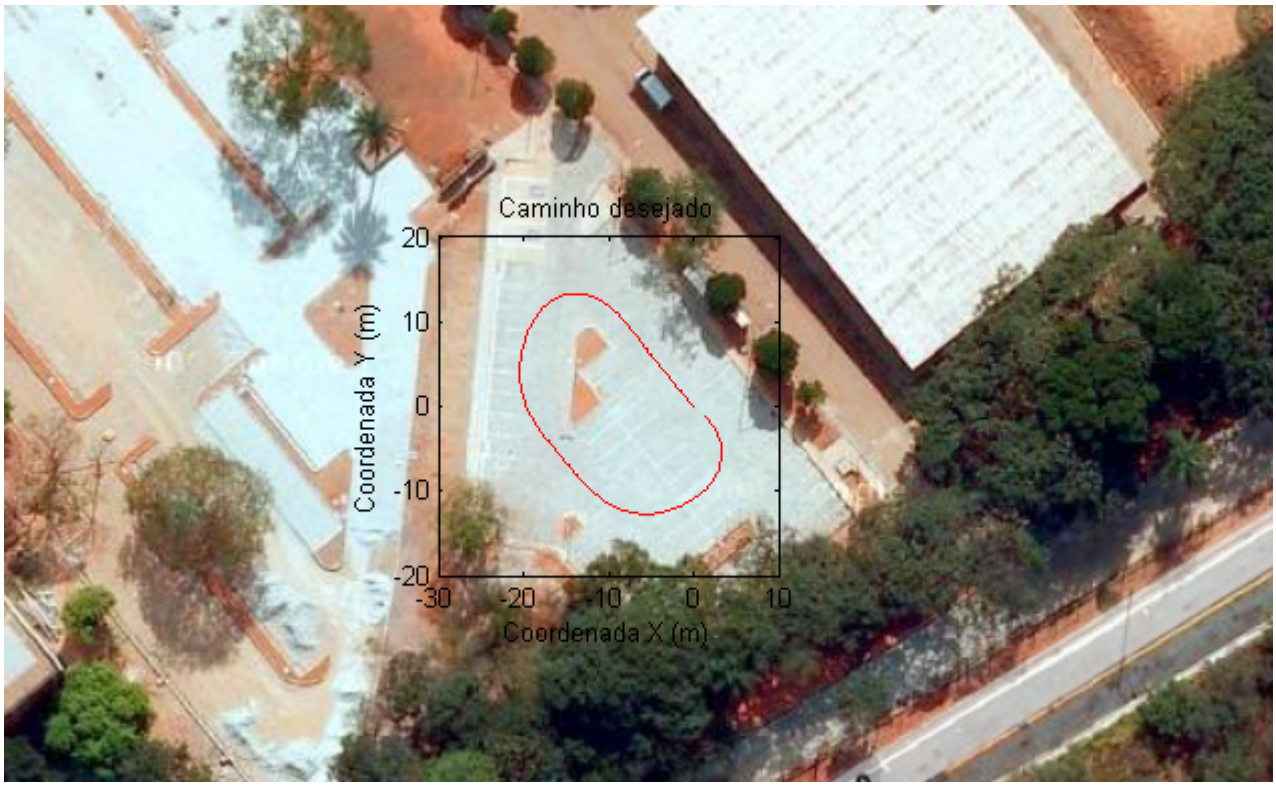

Figura 4.4: Área da UFMG em que foi realizado o experimento.

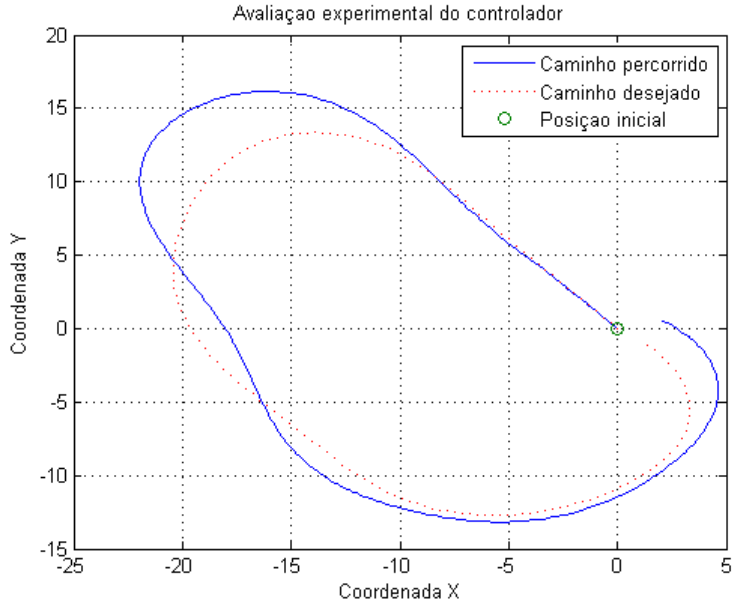

Figura 4.5: Avaliação experimental do controlador com parâmetros $k_1 = 1$ e $k_2 = 3$, sem previsão no cálculo da tangente do caminho. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo.

Uma solução encontrada para minimizar o efeito negativo provocado pelo atraso supracitado foi a previsão do caminho no cálculo da tangente do mesmo, conforme explicado na Seção 3.4. Foi aplicado então a previsão de 20 posições no cálculo da tangente do caminho. Conforme mostrado na Figura 4.6, observou-se um melhor desempenho em relação ao experimento sem a previsão. Mediu-se um erro RMS de 0,73m e um erro máximo de 1,20m.

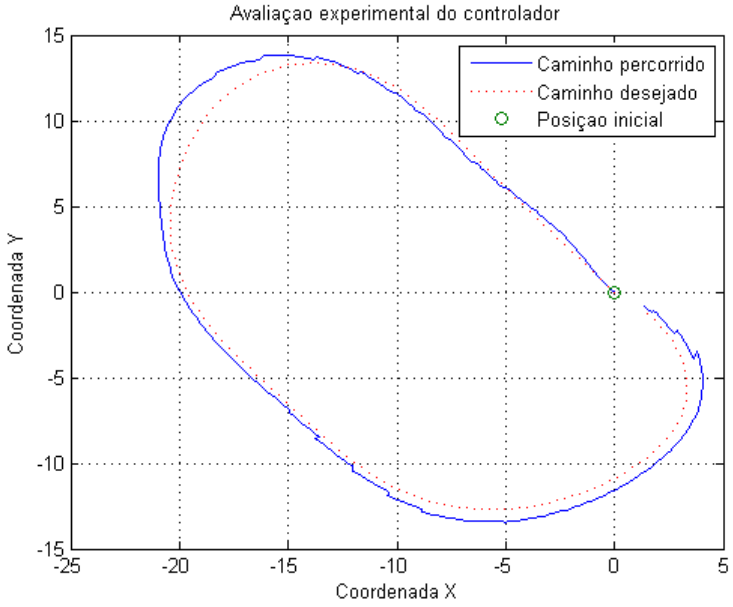

Figura 4.6: Teste prático do controlador com parâmetros $k_1 = 2,1$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo.

Percebe-se também na Figura 4.6 uma pequena oscilação na parte inicial da trajetória, tendo a variável manipulado se mostrado agressiva em alguns momentos, conforma mostrado na Figura 4.7. Foi experimentado então um ganho k_1 menor, aplicando-se num próximo experimento os ganhos $k_1 = 0,72$ e $k_2 = 3$ e novamente a previsão de 20 posições. O resultado está mostrado na Figura 4.8. Foi obtido nesse caso um erro RMS de 0,67m e máximo de 1,17m, havendo assim uma pequena melhora no desempenho do controlador.

Considerando-se que a componente mais importante pelo controle da posição e orientação do veículo é a primeira parcela da Equação (3.7), sendo a segunda parcela responsável pela correção de perturbações e pelo controle fino do caminho, optou-se por manter o controlador com um ganho k_1 menor, resultando numa correção mais lenta do erro, porém com uma variável manipulada menos agressiva.

Repetiu-se por fim o experimento com os ganhos $k_1 = 0,72$ e $k_2 = 3$ e a previsão de

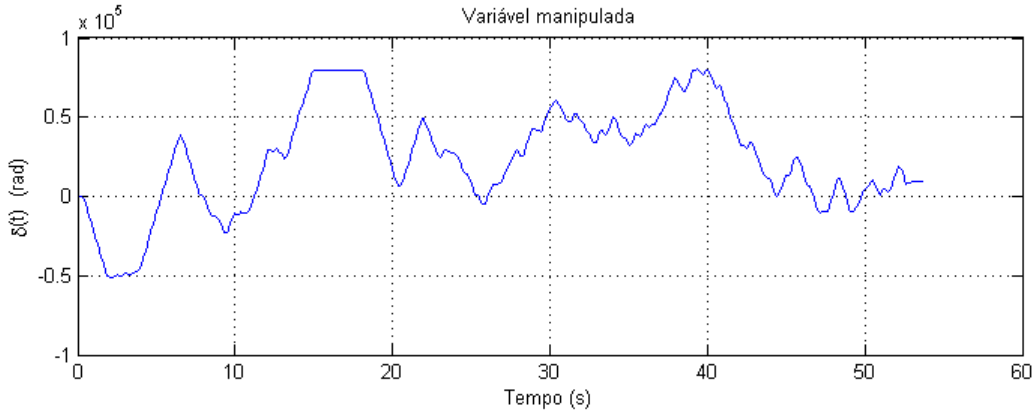

Figura 4.7: Comportamento da variável manipulada ($\delta(t)$) em um experimento com parâmetros do controlador $k_1 = 2,1$ e $k_2 = 3$.

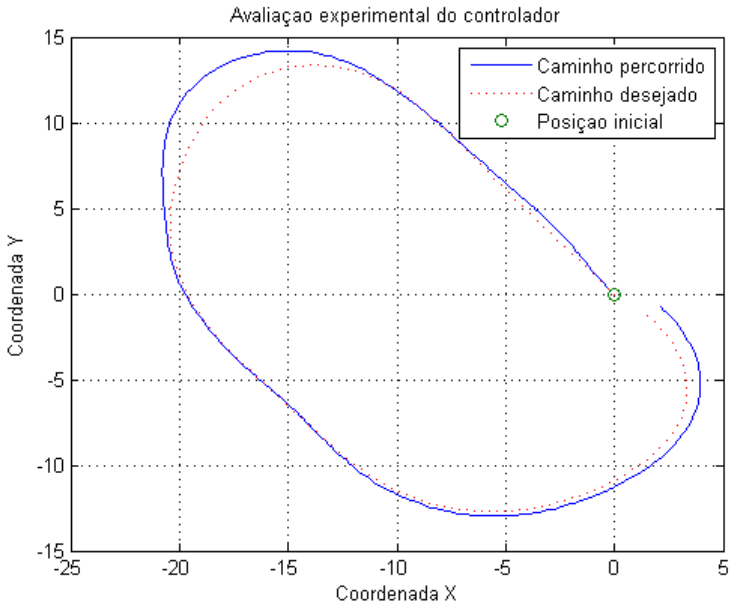

Figura 4.8: Avaliação experimental do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia representa o comportamento do veículo.

20 posições. Obteve-se um resultado semelhante ao anterior (Figura 4.9), com um erro RMS de 0,59m e máximo de 1,18m (Figura 4.10).

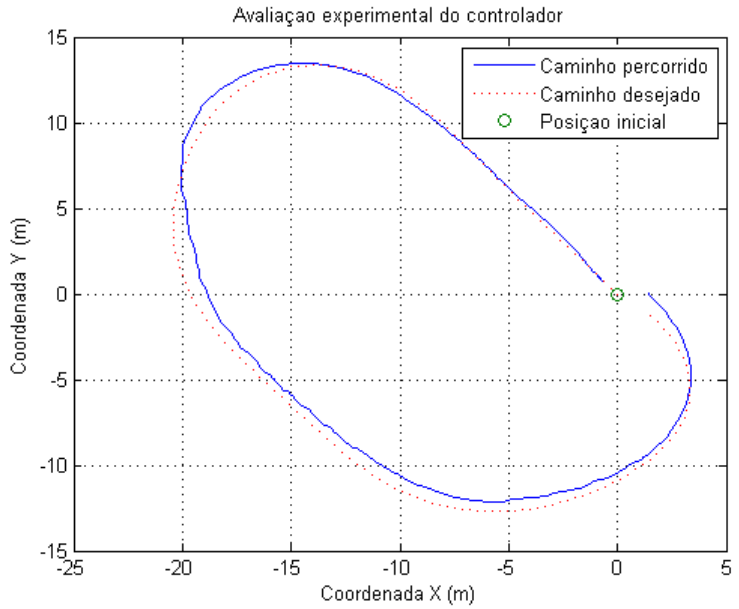

Figura 4.9: Teste prático do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$. A linha tracejada representa o caminho desejado e a linha cheia o comportamento do veículo.

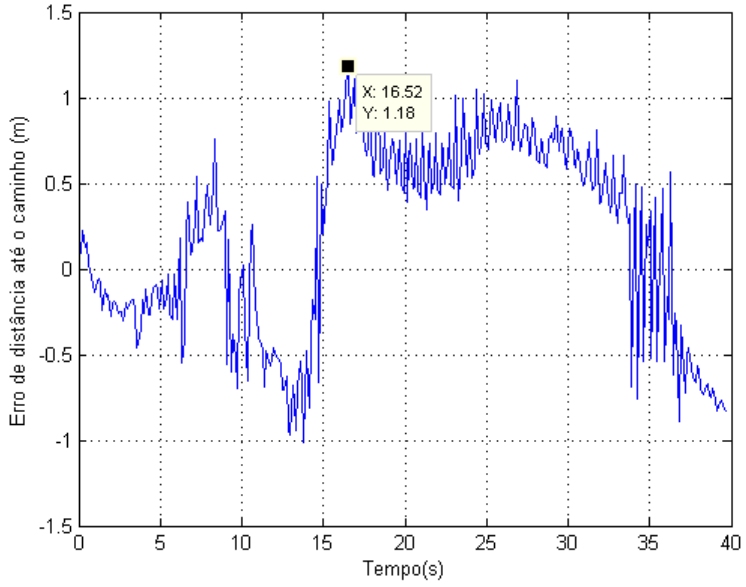

Figura 4.10: Erro de distância obtido no teste prático do controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$.

Por fim, realizou-se um experimento em um local com uma área maior que os demais experimentos apresentados, onde o Cadu seguiu um caminho de 240m de extensão,

retornando a um ponto próximo ao seu ponto de origem. Esta área é coberta por terra e grama, estando o carro mais propício ao escorregamento lateral, fator não levado em conta no modelo utilizado neste trabalho. Um vídeo deste experimento pode ser visto em [CORO, 2009]. O caminho percorrido pelo veículo está mostrado na Figura 4.11. Percebe-se uma anomalia em um certo trecho da curva, onde a mesma apresenta descontinuidades. Esta anomalia foi resultado do sistema de localização do veículo e não representa a verdadeiro comportamento do mesmo, mas uma aproximação. O veículo de forma completamente autônoma o caminho previamente definido do início ao fim com um erro médio (RMS) de 0,51m e um erro máximo de 1,65m (no trecho onde ocorreu a anomalia relacionada ao sistema de localização), conforme mostrado na Figura 4.12.

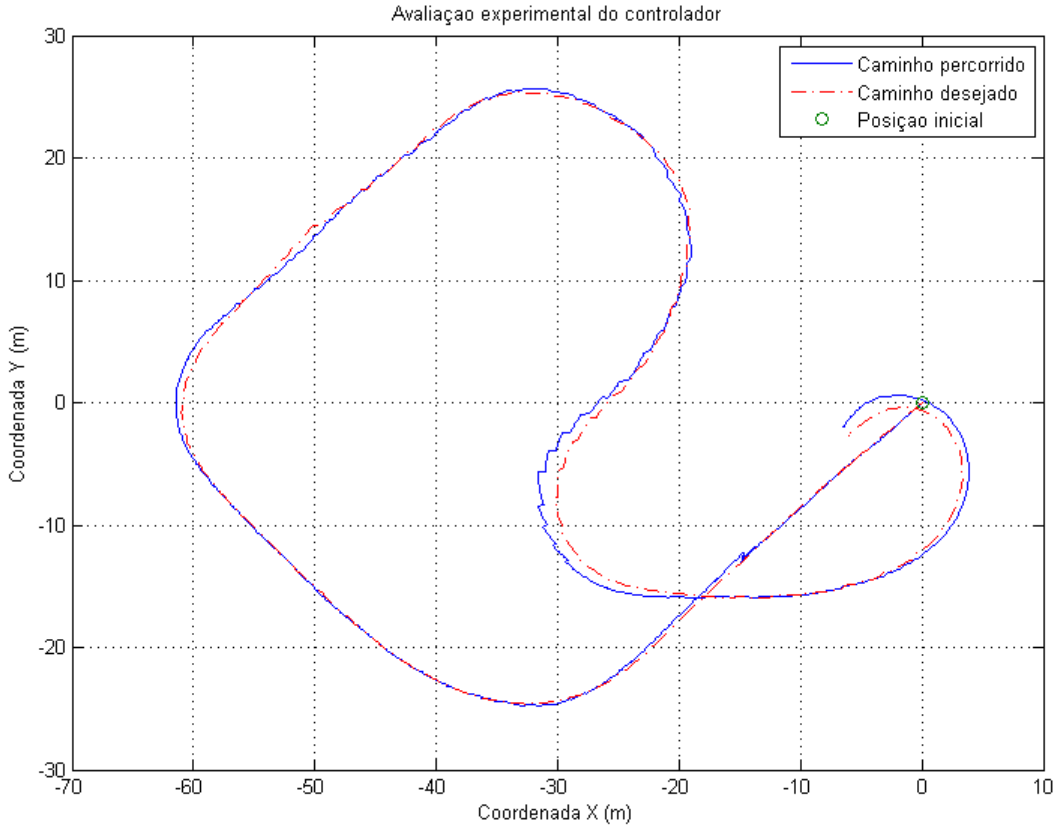

Figura 4.11: Comportamento do veículo em um experimento realizado em um caminho de 240m de extensão. Controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$.

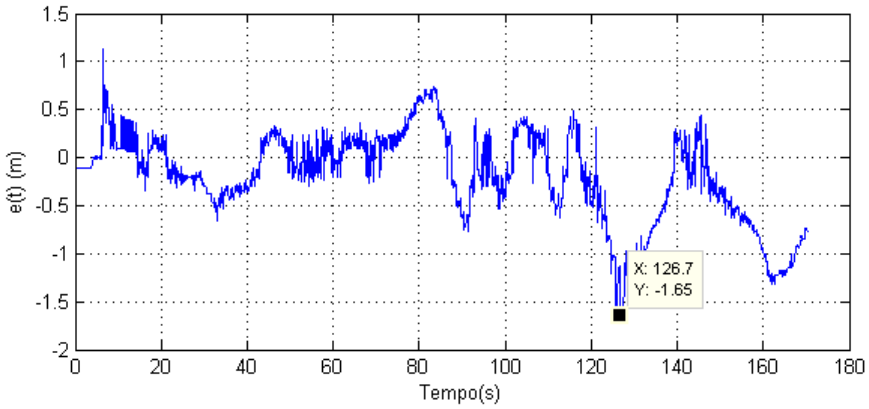

Figura 4.12: Erro de distância obtido em um caminho de 240m de extensão coberto por terra e grama. Controlador com parâmetros $k_1 = 0,72$ e $k_2 = 3$.

CAPÍTULO 5

Conclusões Finais

Como considerações finais, avalia-se o trabalho realizado quanto aos resultados obtidos e esperados. Em seguida, como propostas de continuidade, são apresentadas sugestões para o aperfeiçoamento do sistema desenvolvido e possíveis soluções para os problemas encontrados.

5.1. Resultados

Os sistema de atuação do freio, aceleração, direção e câmbio foram integrados, centralizando as interfaces de operação em um software de controle geral. O funcionamento dos sistemas atuadores foi comprovado por diversos experimentos.

Foi implementada uma interface de comunicação TCP[1] entre o sistema sensorial e o software de controle, permitindo o controle por realimentação de estados. A comunicação entre os *softwares* funcionou de forma robusta e não apresentou falhas durante os experimentos.

Um módulo de controle de direção foi projetado, simulado e implementado no software de operação. O veículo Cadu operou autonomamente, a partir do controle da orientação de suas rodas dianteiras. Resultados experimentais demonstraram a habilidade do veículo de seguir um caminho com extensão de 50m em uma área coberta por paralelepípedos e pedras com um erro médio de 0,59m. Em uma área coberta por terra e grama, o veículo seguiu de forma completamente autônoma um caminho de 240m de

[1]Do inglês, Transmission Control Protocol, é um conjunto de protocolos de comunicação entre computadores em rede.

extensão com um erro médio de 0,51m.

Ao aluno, em especial, a realização deste trabalho, no que concerne aos objetivos de um Projeto Final de Curso, permitiu o amadurecimento de conceitos e aprofundamento em diversos conhecimentos adquiridos ao longo do curso de Engenharia de Controle e Automação.

5.2. Propostas de continuidade

A realização deste trabalho envolveu todos os sistemas de acionamento e sensoriais presentes no veículo, permitindo assim a detecção de pontos passíveis de melhoras no projeto.

Quanto ao sistema de localização do Cadu, sugere-se primeiramente a retirada da IMU do sistema de fusão sensorial. A mesma não trás uma contribuição significativa, uma vez que a única medida aproveitada é a aceleração no eixo X do veículo, podendo esta medida ser por aquela do sensor de velocidade das rodas, que apresenta menos ruído. Além disso, o sistema é vulnerável à descalibragem ou mal-funcionamento dos sensores, podendo qualquer um dos quatro módulos sensoriais comprometer todo o sistema de localização. Contra isso, propõe-se a detecção automática de descalibragem e mal-funcionamento, visto que grande parte das medidas fornecidas são redundantes. Neste contexto, sugere-se também a implementação de um módulo de recalibragem automática do sensor de velocidade das rodas, a partir de informações fornecidas pelo receptor de GPS. Adicionalmente, sugere-se uma revisão do modelo cinemático presente no algoritmo de fusão sensorial.

Entretanto, é também conveniente considerarmos a substituição de todo o sistema de localização por outro com maior precisão e robustez. Neste contexto destaca-se o GPS diferencial, tecnologia capaz de prover informação de localização do veículo com precisão de 20cm ou melhor. A utilização do mesmo dispensaria a necessidade de todos os outros sistemas sensoriais para a localização, o que diminuiria significativamente a vulnerabilidade do sistema. Em contrapartida, esta solução possui um custo elevado e sua utilização desfavorece o objetivo didático do projeto de pesquisa, uma vez que os alunos envolvidos trabalhariam apenas com um sistema sensorial.

A comunicação entre os diversos sistemas sensoriais e atuadores do veículo e o computador central é feita a partir de portas USB. Este meio de comunicação se mostrou impróprio para a aplicação em questão, por não apresentar robustez e praticidade ne-

cessárias. Além disso, o mesmo apresentou falhas ao ser utilizado com um determinado sistema operacional. Propõe-se então a utilização de um barramento de comunicação, à citar CAN[2], LIN[3] e Modbus[4].

Para aumentar o nível de autonomia do veículo Cadu, propõe-se primeiramente o desenvolvimento de um módulo de controle de velocidade, baseado em um sistema de acionamento que permita um controle da frenagem com boa precisão. Em seguida, propõe-se um módulo de planejamento de caminho assistido por um sistema de detecção de obstáculos. Para o módulo citado, sugere-se a implementação de um dos vários algoritmos de planejamento de caminho conhecidos, à citar *Wave-Front*, Decomposição Aleatória, A-Estrela e *Brushfire*. Para a detecção de obstáculos sugere-se um sensor de distância baseado em laser, permitindo a detecção desde objetos grandes como um carro ou uma pessoa até obstáculos pequenos como meios-fios e canteiros. Uma segunda opção seria a detecção de obstáculos por meio de visão estéreo, a partir do cálculo do mapa de disparidade das imagens obtidas, que possui um custo menor e maior facilidade de implementação, porém possui uma menor precisão. Este meio seria pouco confiável para a detecção de obstáculos pequenos como os supracitados. Com a implementação dos sistemas citados neste parágrafo, o veículo se tornaria menos dependente de um sistema de localização de grande precisão (e consequentemente de alto custo), uma vez que o sistema de detecção de obstáculos seria o responsável pelo ajuste fino do controle de direção, mantendo o veículo no centro da pista.

[2]Do inglês, Controller Area Network.
[3]Do inglês, Local Interconnect Network.
[4]Protocolo de comunicação de dados utilizado em sistemas de automação industrial.

APÊNDICE

A

Manual de Operação do Sistema

Este apêndice descreve o procedimento de inicialização do sistema de controle de direção do veículo autônomo Cadu. Este sistema é baseado em módulos sensoriais e de atuação. A sua execução depende do funcionamento dos seguintes módulos: acionamentos do freio, direção e câmbio, receptor de GPS, Sistema de Medições Inerciais (IMU) e o sensor de velocidade das rodas dianteiras.

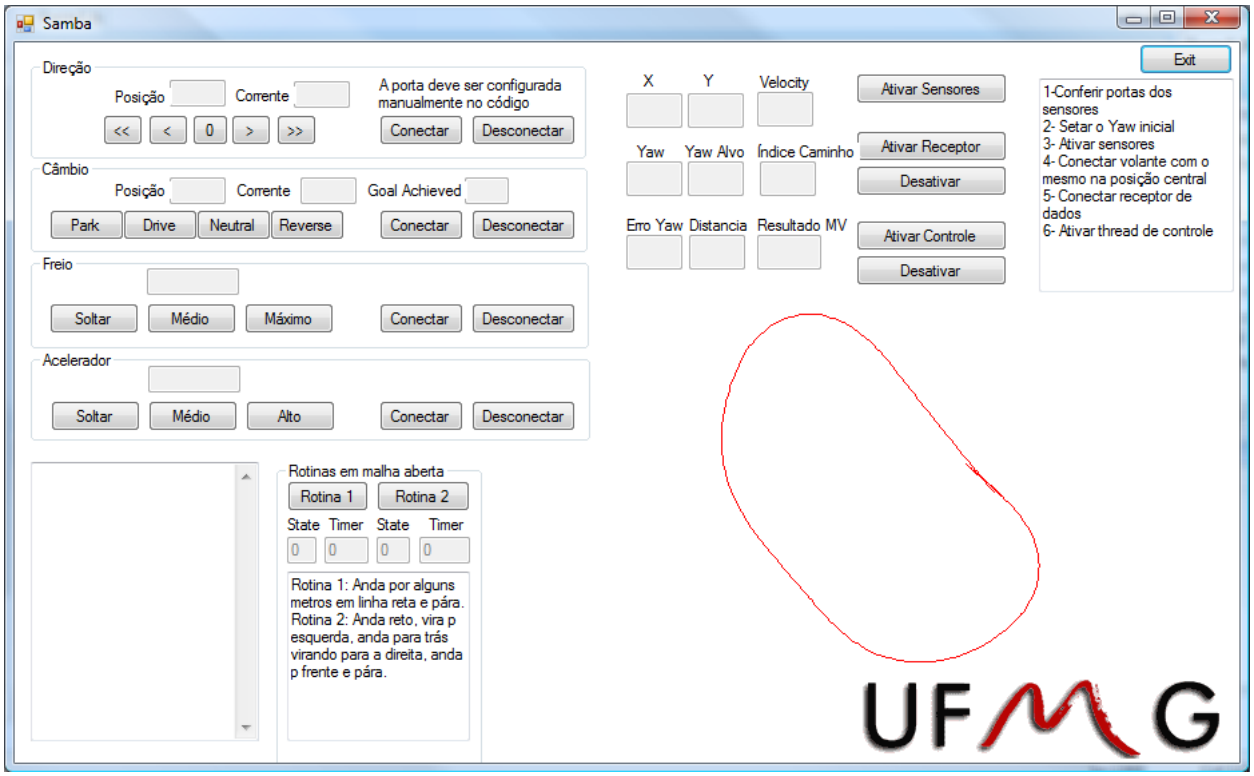

Figura A.1: Interface do Samba, programa principal de operação do veículo.

O procedimento descrito é válido para a execução do software no sistema operacional *Windows XP*. O *software* principal de operação, responsável pelo controle da direção e demais atuadores, foi denominado Samba e sua interface gráfica está mostrada na Figura A.1.

O experimento deve ser realizado em baixas velocidades. A aceleração natural do veículo já o torna excessivamente rápido. Recomenda-se então deixar o freio-de-mão em um nível intermediário (terceiro nível de baixo para cima), para que o carro se movimente em velocidades abaixo de 3,0m/s.

A inicialização do sistema de controle de direção envolve, portanto, os seguintes passos:

1. **Cabos USB**: Plugar cabo do *Hub-USB* ao computador que contém o *software* de operação (Samba). Desplugar todos os cabos USB conectados às entradas do Hub.;

2. **Configuração das portas dos sensores**: Os sensores GPS, IMU e de velocidade das rodas se conectam serialmente ao computador. Utilizando o Gerenciador de Dispositivos do *Windows* (Meu Computador -> Gerenciar), é possível identificar qual a porta serial designada para um periférico (no caso os sensores) no momento em que o mesmo é plugado ao computador, observando o item Portas na lista de dispositivos. Identificadas as portas, deve-se abrir o projeto do *software* na IDE Visual Studio. Abrindo-se os respectivos arquivos ::cppprincipais dos três projetos, a porta é definida nas primeiras linhas do código-fonte. Modifica-se portanto as portas para os valores identificados e compila-se os três projetos. Os arquivos executáveis serão criados na pasta correta, portanto não é necessário fazer modificações adicionais;

3. **Teste dos módulos**: Estando os seis dispositivos citados conectados ao *Hub USB* e este conectado ao computador, cada módulo deve ser testado separadamente a partir da tela de operação principal. Para os atuadores (freio, direção e câmbio), basta clicar em Conectar, enviar um comando manual e verificar se o comando foi executado. Para o acionamento da direção, deve-se clicar adicionalmente em Ativar Controle, para que o acionamento seja ativado e o comando manual seja executado. Para os sensores, deve-se clicar em Ativar Sensores, aguardar que os quatro programas (GPS, IMU, velocidade das Rodas e Fusão Sensorial) sejam executados. Deve-se verificar em cada um dos programas, se estão sendo enviadas as medidas (novas medidas aparecerão na tela à cada segundo continuamente). Caso um deles não envie as medidas, deve-se desenergizar todos eles (por exemplo desligando a chave geral, identificada no painel do carro) e desplugar o cabo

USB (vindo do *Hub*) do computador, para que os microcontroladores internos dos sensores sejam reiniciados. Energiza-se novamente os sensores, pluga-se o cabo USB ao computador e executa-se este procedimento novamente;

4. **Inicialização do Controle de Direção**: O caminho que deseja-se seguir deve estar gravado no arquivo "path1.txt", localizado na mesma pasta que o arquivo executável do *software* Samba. Ativa-se então primeiramente os sensores, clicando em Ativar Sensores. Verificado o correto funcionamento do sensores (conforme explicado no item anterior), ativa-se o acionamento da direção (Conectar), o acionamento do câmbio (Conectar) e o acionamento do freio (Conectar). Verifica-se novamente o funcionamento de todos os módulos. Por fim, deve-se clicar em Ativar Receptor (conecta o Samba ao programa de fusão sensorial) e, por fim, Ativar Controle (a direção, que até o momento estava solta, passa a ter sua posição comandada pelo Samba). Após cerca de 10 segundos (tempo necessário para a inicialização do sistema de localização), o controle do veículo é iniciado;

5. **Execução do Controle**: Neste item, é descrito somente o comportamento do veículo, não será necessária nenhuma intervenção humana. Todas as medidas feitas por todos os sensores e todos os resultados do sistema de localização são continuamente gravados em arquivos localizados na pasta "Logs"do projeto. Primeiramente o freio será pressionado, o câmbio irá de *Park* para *Drive*, o freio será liberado e o carro se movimentará. O Cadu irá então seguir autonomamente o caminho desejado. Ao atingir o fim do caminho, o freio será pressionado, o carro irá parar, o câmbio irá de *Drive* para *Park* e o freio será novamente liberado. Este é o fim da rotina de controle;

6. **Finalização do Programa**: Fecha-se então os cinco programas (Samba, GPS, IMU, velocidade das Rodas e Fusão Sensorial) que continuam em execução, para que a gravação do dados coletados se interrompa. Desliga-se a chave geral, para a desativação dos atuadores. Neste momento o freio será pressionado por alguns segundos, portanto mantenha os pés longe do freio.

À qualquer momento durante a execução do controle, com o carro em movimento, caso o veículo apresente qualquer comportamento diferente do esperado, o sistema de emergência deve ser ativado, pressionando os botões presentes ao seu redor, no interior do veículo ou por controle remoto.

APÊNDICE

B

Método da Seção Áurea

O Método da Seção Áurea desempenha um importante papel no controle da direção do veículo e é aplicável a um problema de minimização onde a solução se encontra em um intervalo conhecido $[a, b]$, sendo neste trabalho o conjunto de posições que formam o caminho desejado, e a função $f(x)$ (a distância da posição atual até o caminho) é unimodal dentro do intervalo. Isto é, o sinal da derivada $\dot{f}(x)$ muda no máximo uma vez em $[a, b]$, de forma que $f(x)$ aumenta ou diminui monotonicamente entre $[a, x^\circ]$ e $[x^\circ, b]$, onde x° é a solução procurada.

O procedimento de busca é feito em 3 passos:

- Passo 1- São selecionados dois pontos, $c = a + (1 - r).h$ e $d = a + r.h$ dentro do intervalo $[a, b]$, onde $r = (\sqrt{5} - 1)/2$ e $h = b - a$.
- Passo 2- Se os valores de $f(x)$ nos dois pontos são muito próximos (i.e, $f(a) \approx f(b)$) e a largura do intervalo e suficientemente pequena (i.e., $h \approx 0$), deve-se sair do loop(?) e declarar $x^\circ = c$ ou $x^\circ = d$ dependendo se $f(c) < f(d)$ ou não. Caso contrário, deve-se passar para o passo 3.
- Passo 3- Se $f(c) < f(d)$, deve-se declarar o novo limite inferior do intervalo $b \leftarrow d$. Caso contrário, deve-se declarar o novo limite superior do intervalo $a \leftarrow c$. Em seguida deve-se retornar ao passo 1.

A cada iteração, a nova largura do intervalo é

$$b - c = b - (a + (1 - r)(b - a)) = r \cdot h; \tag{B.1}$$

ou

$$d - a = a + r.h - a = r \cdot h, \tag{B.2}$$

de forma que o mesmo se torne r vezes a largura do intervalo anterior.

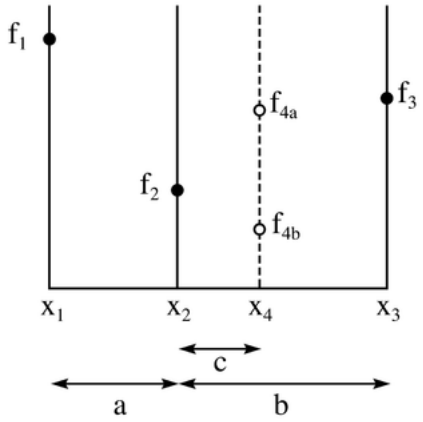

Figura B.1: Método da Seção Áurea

A razão Áurea r é fixa, de forma que um ponto $c_1 = b_1 - r \cdot h_1 = b - r^2.h$ no novo intervalo $[c, b]$ conforme com $d = a + r \cdot h = b - (1 - r).h$, ou seja,

$$r^2 = 1 - r, \tag{B.3}$$

$$r^2 + r - 1 = 0, \tag{B.4}$$

$$r = \frac{-1 + \sqrt{1+4}}{2} = \frac{-1 + \sqrt{5}}{2}. \tag{B.5}$$

Referências

[Choset et al., 2005] Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L. E., and Thrun, S. (2005). *Principles of Robot Motion*. The MIT Press.

[CORO, 2009] CORO (2009). Laboratório de Sistemas de Computação e Robótica. http://coro.cpdee.ufmg.br/. Acessado em 20 de abril de 2009.

[Garmin, 2005] Garmin, I. I. (2005). *GPS 18 Technical Specifications*. Accessed on November 9th, 2009.

[Gingichashvili, 2007] Gingichashvili, S. (2007). Darpa's urban challenge 2007. *TheFutureOfThings.com*. Acessado em 04 de novembro de 2009.

[Hoffmann et al., 2007] Hoffmann, G., Tomlin, C., Montemerlo, M., and Thrun, S. (2007). Autonomous automobile trajectory tracking for off-road driving: Controller design, experimental validation and racing. *Proceedings of the 26th American Control Conference*.

[McBride et al., 2008] McBride, J. R., Ivan, J. C., Rhode, D. S., Rupp, J. D., Rupp, M. Y., Higgins, J. D., Turner, D. D., and Eustice, R. M. (2008). A perspective on emerging automotive safety applications, derived from lessons learned through participation in the darpa grand challenges. *Journal of Field Robotics*, 25(10):808–840. Special Issue on the 2007 DARPA Urban Challenge.

[MicroStrain, 2006] MicroStrain (2006). *3DM-GX1® Product Datasheet*. MicroStrain Inc.

[Nice, 2000] Nice, K. (2000). How automatic transmissions work. *HowStuffWorks.com*. Accessed on November 5th, 2009.

[Online, 2007] Online, F. (2007). Número de mortes em acidentes de trânsito cresce 9% em três anos. *Folha Online*. http://www1.folha.uol.com.br/folha/cotidiano/ult95u134616.shtml.

[Rojo et al., 2007] Rojo, J., Rojas, R., Gunnarsson, K., Simon, M., Wiesel, F., Ruff, F., Wolter, L., Zilly, F., Santrac, N., Ganjineh, T., Sarkohi, A., Ulbrich, F., Latotzky, D., Jankovic, B., and Hohl, G. (2007). Spirit of berlin: An autonomous car for the darpa urban challenge - hardware and software architecture.

[Santos, 2009] Santos, M. M. (2009). Desenvolvimento de um sistema de localização para um veículo terrestre. Dissertação de Mestrado. Belo Horizonte, Brasil.

[Santos et al., 2008] Santos, M. M., Freitas, E. J. R., Vinti, M. N. W., Iscold, P., Tôrres, L. A. B., and Pereira, G. A. S. (2008). Automation and localization of a robotic car. *Proceedings of the 3rd Applied Robotics and Collaborative Systems Enginnering (Robocontrol'08)*, pages 1–7. In: Workshop Robocontrol, 2008, Bauru.

[Silva, 2009] Silva, T. M. (2009). Controle automático do mecanismo de frenagem de um veículo não tripulado. Projeto Final de Curso. Belo Horizonte, Brasil.

[Thrun, 2006] Thrun, S. (2006). Winning the darpa grand challenge. *Google TechTalks*.

[Thrun et al., 2006] Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A., and Mahoney, P. (2006). Stanley: The robot that won the darpa grand challenge. *Journal of Field Robotics*, 23(9):661–692. Special Issue on the DARPA Grand Challenge.

[Xu, 2007] Xu, G. (2007). *GPS: Theory, Algorithms and Applications*. Number ISBN: 3540727140. Springer. 2a edição.