

JÉSSICA BARBOSA HELUANY
WILLIAM RICARDO KISHIMOTO DINIZ

**MODELO SIMPLIFICADO EM ESCALA 1:16 DE CARRO
AUTOGUIADO UTILIZANDO UMA CÂMERA COMO
SENSOR EM AMBIENTE CONTROLADO**

São Paulo
2012



**JÉSSICA BARBOSA HELUANY
WILLIAM RICARDO KISHIMOTO DINIZ**

**MODELO SIMPLIFICADO EM ESCALA 1:16 DE CARRO
AUTOGUIADO UTILIZANDO UMA CÂMERA COMO SENSOR EM
AMBIENTE CONTROLADO**

Dissertação de projeto de formatura
apresentada à Escola Politécnica da
Universidade de São Paulo

Área de concentração: Engenharia
Elétrica

Orientadores:
Prof. Dr. Fuad Kassab Junior
Prof. Dr. Ricardo Paulino Marques

**São Paulo
2012**

Às nossas pais

Dionéia & Sérgio Meluany

Cecília & Milton Diniz

*que sempre batalharam para que
pudéssemos chegar onde estamos hoje.*

AGRADECIMENTOS

- Aos Professores Dr(s). Fuad Kassab Junior e Ricardo Paulino Marques pela orientação, discussões e acompanhamento constantes;
- Ao aluno de graduação Wesley Wein Chen pela ajuda e companhia;
- Aos colegas e amigos da EPUSP, pelas colaborações e apoio moral prestadas em algum momento deste trabalho;
- Às nossas famílias pelo apoio e compreensão;

*"Scientists study the world as it is;
engineers create the world that has never been."*

Theodore von Kármán

RESUMO

Nos últimos anos o carro autônomo tem sido pesquisado por diversas razões, desde aplicações militares até o âmbito social, para ajudar na melhoria da qualidade de vida das pessoas. Estas pesquisas têm sido realizadas em diversos países sendo que a DARPA (órgão de defesa dos EUA) realiza o maior evento neste ramo. O Brasil ainda possui poucas pesquisas na área e a intenção deste projeto é incentivar estas pesquisas. Para isto será desenvolvido um modelo simplificado em escala 1:16 de carro autônomo e este deverá funcionar em um ambiente controlado. Contudo, apesar destas restrições simplificadoras, o projeto deve tomar o cuidado de desenvolver sistemas que possam ser adaptados a veículos reais. Assim sendo, o protótipo deverá utilizar uma câmera para detectar a trajetória e deverá ser completamente embarcado e independente de conexões externas.

Palavras chave: Carro autônomo. Sistema embarcado. Telemetria. Controle de estabilidade. Controle de posição. Controle de velocidade. Aquisição de imagens. Filtragem de imagens.

ABSTRACT

In recent years, the autonomous vehicle has been researched for several reasons, from military applications to the social sphere, to improve the life quality of people. These studies have been conducted in several countries, and DARPA (the U.S. defense agency) has the biggest event in this field. Brazil still has few researchs in the field and the intent of this project is to encourage such research. For this objective, a simplified model will be developed in a 1:16 scale autonomous vehicle that should operate in a controlled environment. However, despite these simplifying constraints, the project should take care to develop systems that can be adapted to real vehicles. Thus, the prototype will use a camera to detect the trajectory and should be fully embedded and independent of external connections.

Keywords: Autonomous vehicle. Embedded systems. Telemetry. Estability control. Position control. Speed control. Image acquisition. Image filtering.

LISTA DE FIGURAS

Figura 2-1 Foto do chassi.	3
Figura 2-2 Exemplo do carrinho seguindo pista.	4
Figura 3-1 Vista de cima do chassi e seus componentes mecânicos principais.	6
Figura 3-2 Esquemático da estrutura final do carrinho.	7
Figura 3-3 Suporte de apoio da placa da câmera e a placa, respectivamente.	8
Figura 3-4 Haste e suporte da câmera.	8
Figura 3-5 Placa dos sensores de velocidade.	9
Figura 3-6. Placas de alimentação e da ponte-H dos motores, respectivamente.	10
Figura 3-7. Apoio das placas de alimentação e dos motores.	10
Figura 3-8. Apoio da placa da Freescale (à esquerda) e a própria placa (à direita).	11
Figura 3-9 - Esquema da montagem das rodas com as lãs de Neodímio.	11
Figura 4-1 - Esquemática da Placa de Alimentação.	13
Figura 4-2 - Desenho do Circuito Impresso da Placa de Alimentação.	14
Figura 4-3 - Esquemática da Placa de Controle dos Motores.	15
Figura 4-4 - Desenho do Circuito Impresso de Controle dos Motores.	16
Figura 4-5 - Esquemática da Placa de Controle dos Motores.	17
Figura 4-6 - Desenho do Circuito Impresso de Controle dos Motores.	17
Figura 4-7 Desenho do Circuito Impresso da Placa de Filtragem da Câmera.	18
Figura 4-8 Esquemático da Placa de Filtragem da Câmera.	19
Figura 5-1 - Sinal PWM.	20
Figura 5-2 - Exemplo Câmera Matricial Monocromática.	22
Figura 5-3 - Sinal Analógico de Sincronização Horizontal de uma Câmera Matricial.	23
Figura 5-4 - Exemplo Câmera Matricial Colorida.	24
Figura 5-5 - Exemplo de Sinal da Câmera Linear (Freescale Application Note Linescan Camera Mode).	25
Figura 5-6 - Exemplo da Aplicação do filtro de Gauss variando σ [9].	26
Figura 5-7 - Os seis graus de liberdade de um objeto. Adaptada de [10].	28
Figura 5-8 - Efeito da Subviragem ("Understeering").	29
Figura 5-9 - Correção da Subviragem ("Understeering") pelo controle de Estabilidade (roda em vermelho freada).	29
Figura 5-10 - Efeito da Sobreviragem ("Oversteering").	30
Figura 5-11 - Correção da sobreviragem ("Oversteering") pelo controle de estabilidade (roda em vermelho freada).	30
Figura 6-1. Diferença do erro no centro de massa e de um ponto à frente.	31
Figura 6-2. Diagrama do Modelo (Referência [8]).	32
Figura 7-1. Conversões e protocolos de comunicação do projeto.	37
Figura 7-2. Fluxograma do Software de Telemetria.	39
Figura 7-3. Software de Telemetria (conectando o software).	40
Figura 7-4. Software de Telemetria (Conectado).	41
Figura 7-5. Software de Telemetria (Gráficos).	41
Figura 7-6. Software de Telemetria (Gráficos) - Botão Pausar/Continuar.	42
Figura 7-7. Arquivo CSV aberto no Excel e no Bloco de Notas.	42
Figura 9-1 - Exemplo de bom (esquerda) e mal (direita) ajuste do foco da câmera.	49
Figura 9-2 - Sistema para Calibração do Giroscópio.	51
Figura 9-3 - Resultado da Calibração do Giroscópio.	54
Figura 10-1 - Sinal da Câmera em Relação à posição do Carrinho.	57
Figura 10-2 - Exemplo de sinal de saída da câmera.	57
Figura 10-3 - Filtragem da imagem por Hardware (comparador).	58
Figura 10-4 - Filtragem da imagem por Hardware (comparador) com Ruído.	58
Figura 10-5 - Rotina do Microcontrolador.	60
Figura 10-6 Trecho de imagem obtida pela câmera linear.	64
Figura 10-7 - Exemplo de Pista com variação na iluminação durante o trajeto.	65
Figura 11-1. Diagrama de blocos do controle de posição.	67
Figura 11-2. Lugar geométrico das raízes para a função de transferência do modelo do carrinho.	69

Figura 11-3. Interpolação da função $\arctg()$ por polinômio de grau 2	70
Figura 12-1. Gráfico das velocidades de regime	72
Figura 12-2. Gráfico do PWM em 60%	73
Figura 13-1 - Diagrama de velocidades nas rodas traseiras	77
Figura 14-1. Reta montada para a realização dos testes	80
Figura 14-2. Gráfico do erro de posição na reta	81
Figura 14-3. Gráfico do controle de velocidade de 2m/s.	81
Figura 14-4 - Pista de Testes	82
Figura 14-5 - Resultado do Controle de Estabilidade	85
Figura 15-1 - Esquemático do Mapeamento de Pontos	86
Figura 15-2 - Translação de Sistemas Cartesianos.....	87
Figura 15-3 - Rotação de Sistemas Cartesianos	88
Figura 15-4 - Aproximação da função Cosseno.....	90
Figura 15-5 - Aproximação da função Cosseno entre -15° e 15°	90
Figura 17-1 - Exemplo de Sinalização.....	102

LISTA DE TABELAS

Tabela 1. Orçamento preliminar	5
Tabela 2. Siglas do software de telemetria	38
Tabela 3. Velocidades para seus respectivos PWMs.....	47
Tabela 4. Relação de PWM com o raio de curva e o ângulo do servo	48
Tabela 5 - Valores da Calibração do Giroscópio para rotação de 273,421º/s.....	52
Tabela 6 - Valores da Calibração do Giroscópio para rotação de 203,008º/s.....	52
Tabela 7 - Valores da Calibração do Giroscópio para rotação de 183,052º/s.....	52
Tabela 8 - Valores da Calibração do Giroscópio para rotação de 175,897º/s.....	52
Tabela 9 - Valores da Calibração do Giroscópio para rotação de -243,795º/s.....	53
Tabela 10 - Valores da Calibração do Giroscópio para rotação de -214,499º/s.....	53
Tabela 11 - Valores da Calibração do Giroscópio para rotação de -185,249º/s.....	53
Tabela 12 - Valores da Calibração do Giroscópio para rotação de -176,183º/s.....	53
Tabela 13 - Pontos para Calibração do Giroscópio	54
Tabela 14 - Variação de Velocidade de Reto Segundo o Erro de Posição Medido.....	96

SUMÁRIO

1. Introdução.....	1
1.1 Título do Projeto.....	1
1.2 Motivação.....	1
1.3 Objetivos Gerais do Projeto	1
1.4 Objetivos Específicos do Projeto	2
2. Análise Preliminar do Projeto	3
2.1 Componentes Principais do Projeto	3
2.2 Orçamento preliminar.....	5
2.3 Viabilidade/Riscos.....	5
3. Estrutura mecânica	6
3.1 Chassi	6
3.2 Suportes Mecânicos.....	7
3.2.1 Suporte da placa da câmera.....	8
3.2.2 Haste da Câmera.....	8
3.2.3 Placa dos Sensores de Velocidade.....	9
3.2.4 Suporte da placa de alimentação e da placa dos motores.....	9
3.2.5 Suporte da Placa da Freescale	10
3.3 Rodas.....	11
4. Circuitos Impressos.....	13
4.1 Placa de Alimentação	13
4.2 Controle dos Motores	14
4.3 Placa dos Sensores Hall.....	16
4.4 Placa da Filtragem da Câmera.....	17
5. Conceitos Básicos.....	20
5.1 Sinal PWM	20
5.2 Conversão A/D	20
5.3 Câmeras para a obtenção da pista.....	21
5.4 Sinal da câmera	21
5.5 Câmera Matricial X Câmera Linear	22
5.5.1 Câmera Matricial.....	22
5.5.2 Câmera Linear	24
5.6 Filtro Gaussiano	26
5.7 Algoritmo de Canny	27
5.8 Giroscópio	27

5.9 Controle de Estabilidade - Conceito.....	28
6. Modelos.....	31
6.1 Modelo Cinemático do Carrinho.....	31
6.2 Modelo Cinemático dos Motores.....	35
7. Software de Telemetria.....	37
8. Sensores, Atuadores e Medida de Qualidade.....	43
8.1 Atuadores.....	43
8.1.2 Servo Motor.....	43
8.1.3 Motores.....	43
8.2 Sensores.....	44
8.2.1 Câmera.....	44
8.2.2 Sensores Hall.....	44
8.2.3 Giroscópio.....	45
8.3 Medida de Qualidade do Projeto.....	45
9. Obtenção de dados para Calibração de Sensores e Atuadores.....	47
9.1 Atuadores.....	47
9.1.1 Motor.....	47
9.1.2 Servo.....	47
9.2 Sensores.....	49
9.2.1 Câmera.....	49
9.2.2 Giroscópio.....	50
10. Obtenção e Filtragem da Imagem.....	56
10.1 Obtenção da Imagem.....	56
10.1.1 Por Largura de Pulso (IPWM).....	56
10.1.2 Por Conversão A/D.....	59
10.2 Filtragem da Imagem.....	60
10.2.1 Detecção de Bordas.....	60
10.2.2 Filtro Diferencial Isolado.....	61
10.2.3 Algoritmo de Canny.....	63
11. Sistema de Controle de Posição.....	67
12. Sistema de Controle de Velocidade.....	72
13. Sistema de Controle de Estabilidade.....	76
14. Resultados Obtidos com Sistemas de Controle.....	80
14.1 Sistema de Controle de Posição.....	80
14.2 Sistema de Controle de Velocidade.....	81
14.3 Ganhos de Desempenho em Velocidade Constante.....	82
14.4 Obtenção de Imagem por Largura de Pulso (referência).....	83

14.5	Obtenção de Imagem por Conversão A/D com Filtro Diferencial Isolado	83
14.6	Obtenção de Imagem por Conversão A/D com Algoritmo de Canny	83
14.7	Sistema de Controle de Estabilidade	84
15.	Armazenagem e Correção de Pontos (Mapeamento Local)	86
15.1	Determinação de estado: Reta ou Curva	91
15.1.1	Pelo Raio de Curvatura	91
15.1.2	Pelo Coeficiente Angular	93
15.1.3	Pelo Erro Acumulado	93
15.1.4	O Método Escolhido	94
16.	Otimizações com Velocidades Variáveis	95
16.1	O Ganho Integral em Curvas	95
16.2	Decisão de Velocidade	95
17.	Conclusões	98
17.1	Análise do Projeto	98
17.2	Propostas de Continuidade	101
17.3	Aumento da Escala	101
17.4	Microcontrolador para Telemetria	101
17.5	Interpretação de Sinais na Pista	102
17.6	Identificação de Obstruções da Pista	103
18.	Bibliografia principal para o trabalho	104
19.	Resumo do conteúdo da bibliografia principal	105
20.	Referências	106
21.	ANEXOS	108

1. Introdução

1.1 Título do Projeto

O projeto recebeu o nome de Poli – Gryphon em referência ao animal mitológico que possui corpo de leão e, cabeça, asas e garras superiores de águia. Isto porque desejamos que o carrinho apresente a agressividade e a força de um leão ao mesmo tempo em que tenha também a agilidade, velocidade e visão de uma águia.

1.2 Motivação

Em 1970 foram produzidos 30 milhões de carros no mundo e, em 2005, este número já havia aumentado para 65 milhões. Este crescimento faz com que seja necessária uma utilização mais efetiva das ruas, estradas e combustível [1].

O carro autônomo tem sido pesquisado nos últimos anos como uma possível solução para este problema. Há vários projetos ao redor do mundo sobre esse assunto, entre os quais se destaca o projeto DARPA (*Defense Advance Projects Agency*), cujo objetivo é estimular os princípios científicos relativos ao movimento, projeto, e uso de veículos e robôs autônomos [2]. Um dos exemplos de carro desenvolvido neste projeto é o 'Little Ben', que foi desenvolvido por membros de três faculdades americanas e conseguiu cumprir o objetivo da competição (efetuar manobras efetivamente no trânsito), através da modificação de um Toyota Prius [3]. E este projeto de formatura procura incentivar justamente a pesquisa deste tipo de veículo no país.

Apesar de ser feito em escala (1:16), os sistemas, modelos e softwares desenvolvidos no decorrer do projeto poderão ser adaptados para veículos em escala natural, dando um passo a mais no desenvolvimento desta tecnologia no Brasil.

1.3 Objetivos Gerais do Projeto

O objetivo principal do projeto é o estudo de sistemas necessários para o desenvolvimento e para a otimização de carros autônomos.

Para este projeto foi considerado um ambiente controlado onde não há objetos que possam obstruir a pista. Os obstáculos considerados durante o decorrer do projeto foram curvas (para a direita e para a esquerda) com raio mínimo de 50 centímetros e uma rampa com inclinação máxima de 15°.

A pista utilizada possui cor branca, 60 centímetros de largura e tem seu centro demarcado por uma faixa preta que poderá variar de 1,7 a 3,0 centímetros de comprimento.

Para testar estes sistemas foi desenvolvido um sistema embarcado capaz de fazer com que um carrinho (escala 1:16) fosse capaz de percorrer a pista descrita acima de maneira autônoma, ou seja, sem qualquer comunicação ou auxílio externo ao sistema.

O carrinho utiliza a linha preta para identificar sua posição em relação à pista, mas não é obrigado a se manter em cima da linha, contudo ele deve se manter com pelo menos duas rodas dentro da delimitação da pista.

Como objetivo para a sua performance, foi definida uma velocidade de reta mínima de 2,0 m/s.

1.4 Objetivos Específicos do Projeto

Para cumprir o objetivo principal, alguns resultados parciais foram visados. Para que o protótipo seguisse a pista, a primeira tarefa feita foi detectá-la. Isto foi feito através de um sensor óptico devido à fácil adaptação do mesmo em um carro em escala natural (considerando, para esta comparação, que o carro em escala natural se manteria no ambiente controlado descrito acima). Após a obtenção da imagem pelo sensor óptico a mesma deve ser tratada para que o protótipo identifique a trajetória a ser seguida, ou seja, foram necessários algoritmos capazes de interpretar a imagem, detectando as curvas.

Além da interpretação das curvas, foi importante a existência de uma malha de controle de velocidade que permitisse que o protótipo se mantivesse em uma velocidade constante independentemente de estar em uma reta ou uma curva. Este controle também é importante para que o carrinho possa variar sua velocidade dependendo de seu estado atual. Em uma reta ele pode andar mais rápido que em uma curva, aumentando sua eficiência. Para este controle foi estabelecido um tempo de subida de 0,5 segundo.

Com as curvas detectadas e o controle de velocidade, foi possível trabalharmos no controle de posição. Ou seja, a malha de controle responsável por manter o carrinho na pista.

Com estes objetivos específicos cumpridos, o protótipo já era capaz de andar sozinho em baixas velocidades.

2. Análise Preliminar do Projeto

2.1 Componentes Principais do Projeto

Os componentes principais do projeto são três: Chassi, Pista e Microcontrolador. A descrição detalhada de cada um deles se encontra abaixo.

As medidas da pista foram feitas pensando-se em uma pista real de 10 metros de largura ($10\text{m}/16 = 62,5\text{cm}$).

O chassi e o microcontrolador foram reaproveitados de projetos anteriores devido ao fato de já estarem disponíveis (não foi necessário esperar/pagar frete de importação). Além disso, já havíamos pagado por eles (custo de projeto reduzido) e, como foram utilizados em projetos anteriores, o conhecimento prévio da mecânica do chassi e do funcionamento do microcontrolador e sua placa de validação já estavam avançados.

Por fim, o sensor óptico foi escolhido com auxílio de uma matriz AHP que se encontra no Anexo II.

- 1) **Chassi:** o chassi utilizado (Figura 2-1) é o vendido pela Freescale Semiconductor e é composto principalmente por quatro rodas, dois motores – um em cada roda traseira, e um servo (Futaba S3003) para controlar a direção das rodas dianteiras de maneira que se possa modificar o ângulo das mesmas.



Figura 2-1 Foto do chassi.

- 2) **Pista:** a pista, como descrita anteriormente, será branca com aproximadamente 60 cm de largura. Para que o carrinho se localize, uma linha preta com largura entre 1,7 cm a 3,0 cm é desenhada em seu centro. Contudo, esta linha serve apenas como guia para que ele identifique a trajetória. O carrinho deve se manter

na pista branca, mas não é necessário que este siga a linha preta. Por exemplo, ele poderá realizar uma curva mais fechada do que a linha preta sugere (como mostra a Figura 2-2 abaixo).

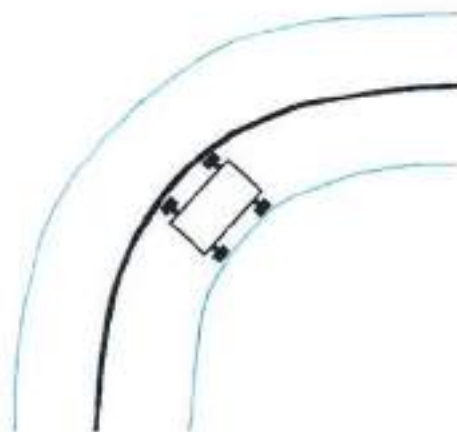


Figura 2-2 Exemplo do carrinho seguindo pista.

O tamanho de 60cm da pista representa, na escala escolhida (1:16), uma pista de aproximadamente 10 metros que acreditamos ser um valor razoável, já considerando uma margem de segurança, para que um veículo em escala natural consiga se manter.

Para efeito de comparação, um circuito de Fórmula 1 recente (Yas Marina em Abu Dhabi, inaugurado em outubro de 2009) possui uma largura de pista que varia entre 12 metros (largura mínima) e 16 metros (largura máxima) [4].

- 3) **Microcontrolador:** O carrinho utiliza o microcontrolador MPC5604B da Freescale Semiconductor. Apenas um microcontrolador é utilizado e sua escolha se deve ao fato de ser um microcontrolador de 32 bits que faz parte de uma família de microcontroladores especializada em aplicações automotivas [5], assim, ao realizarmos o projeto nele, estaremos aumentando a chance de compatibilidade entre nosso sistema e carros atuais. Além disso, já possuíamos este microcontrolador devido a projetos anteriores.
- 4) **Sensor Óptico:** A detecção de trajetória é realizada com o auxílio de uma câmera (sensor CCD – *Charge Coupled Device*).

A escolha da câmera para a detecção de trajetória foi motivada pelo fato de que ela seria mais útil para um carro em escala natural que outros sensores (como sensores de refletância ou sensores Hall). Isto porque não precisaria modificar as pistas já existentes e ainda poderia servir para outras utilidades (como detecção de semáforos e obstruções da pista) que não serão abordadas neste projeto.

2.2 Orçamento preliminar

Tabela 1. Orçamento preliminar.

Item	Preço (R\$)
Câmera CCD	62
Placas de cobre para circuitos impressos	30
Cabos de conexão	50
Conectores	15
Peças para suporte mecânico	50
Acelerômetro e giroscópio	120
Total	357

2.3 Viabilidade/Riscos

Como os integrantes do grupo são de departamentos diferentes e nas áreas mais necessárias para o desenvolvimento do projeto (sistemas eletrônicos e automação e controle), acabou tornando-se viável a realização deste projeto tão interdisciplinar. Isto porque cada aluno atuou onde possui mais conhecimento, ao mesmo tempo em que houve uma constante discussão da integração das partes.

A integração dos sistemas foi a parte de maior risco para o projeto, pois ele é composto de vários sistemas que devem ser interligados corretamente entre si não só no hardware, mas também no software associado.

Nota-se, também, que uma grande dificuldade em projetos que envolvem muita eletrônica é a necessidade de importação de dispositivos, o que pode atrasar o cronograma devido ao frete e atrasos na taxaço.

3. Estrutura mecânica

3.1 Chassi

Para a confecção do carrinho foi utilizado um kit da Freescale Semiconductor composto pelas seguintes partes:

- Chassi
- Servo motor
- Placa de interface do microcontrolador Bolero
- Bateria de 7,2V

Sendo o chassi equipado de quatro rodas emborrachadas e dois motores DC, cada um acoplado a uma das rodas traseiras por meio de um conjunto de engrenagens para redução.

O servo motor possuía duas possibilidades de fixação no chassi: uma na parte mais frontal, e a outra um pouco recuada. Optou-se pela posição frontal para que sobrasse mais espaço para a construção dos apoios das outras placas de circuito impresso e da haste de fixação da câmera. A seguir pode-se ver uma foto (Figura 3-1) do chassi apenas com os componentes do kit (exceto a placa do microcontrolador e a bateria):

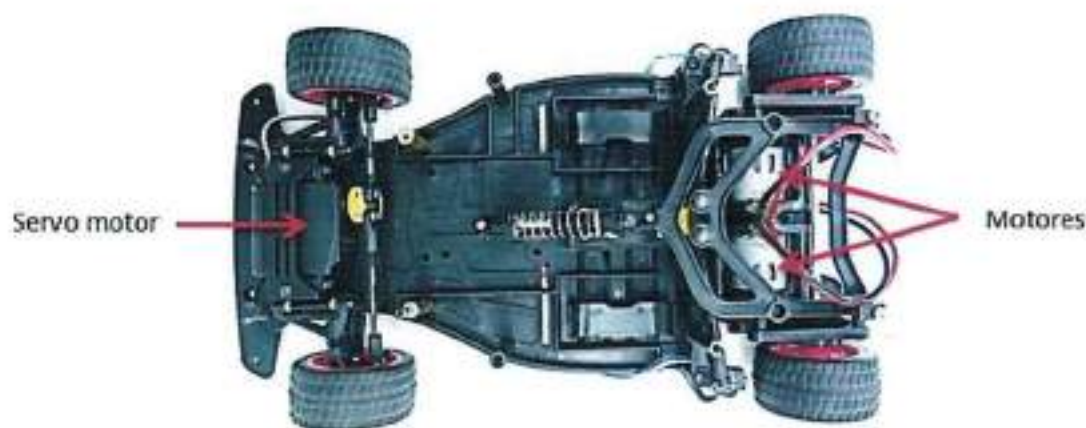


Figura 3-1 Vista de cima do chassi e seus componentes mecânicos principais.

As dimensões do chassi são de, aproximadamente, 16,2 x 27,5cm. Nota-se que a parte de trás, onde estão os motores, possui uma espécie de suspensão, dando mais mobilidade e flexibilidade ao chassi quando este está em movimento. Esta mobilidade é ajustável por um parafuso preso a uma cabeça redonda que permite movimento em todas as direções, e pode ser um parâmetro importante quando forem considerados obstáculos.

3.2 Suportes Mecânicos

Para a fixação das placas de circuito impresso e da câmera, foram confeccionados suportes feitos com MDF e recobertos com fibra de vidro, sendo posteriormente lixados e pintados de preto para que combinassem com as cores do chassi. Como o material do chassi é plástico, isso facilitou a abertura de furos onde foi necessário para a colocação de peças. No total, foram manufaturadas para a parte mecânica três bases para fixação das placas, e uma haste e um suporte para a câmera. Quanto à parte eletrônica, foram feitos quatro circuitos: um para a câmera, um de alimentação, um para os sensores de velocidade e um para a ponte-H dos motores, os quais serão descritos no capítulo 4.

Durante o projeto mecânico tomou-se muito cuidado com o posicionamento de tudo para que o carrinho ficasse o mais baixo possível, visando manter o centro de massa baixo e, portanto, proporcionando mais estabilidade ao carrinho. Além disso, o peso concentrado próximo ao eixo traseiro aumenta a força normal neste ponto, consequentemente aumentando a tração das rodas traseiras com a pista.

A seguir, tem-se uma descrição de todas as peças do carrinho, incluindo os apoios e as placas de circuito começando da parte frontal do chassi até cobrir toda a sua extensão. Abaixo se tem um esquemático (Figura 3-2) da estrutura final do carrinho:

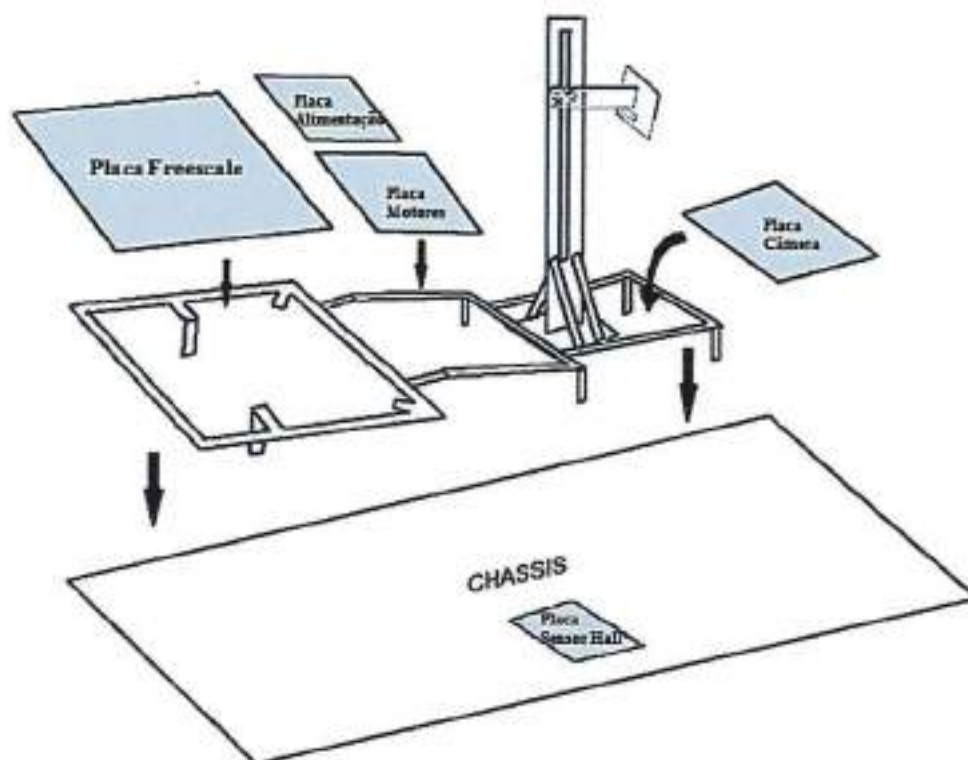


Figura 3-2 Esquemático da estrutura final do carrinho.

3.2.1 Suporte da placa da câmera

Para posicionar a placa da câmera na frente do chassi era preciso elevá-la acima da altura do servomotor. Isso foi feito com o uso de pequenas peças metálicas de fixação, parafusos e o apoio feito em MDF (Figura 3-3). O tamanho do apoio é bem próximo ao da placa, garantindo menos peso e um melhor design ao conjunto.

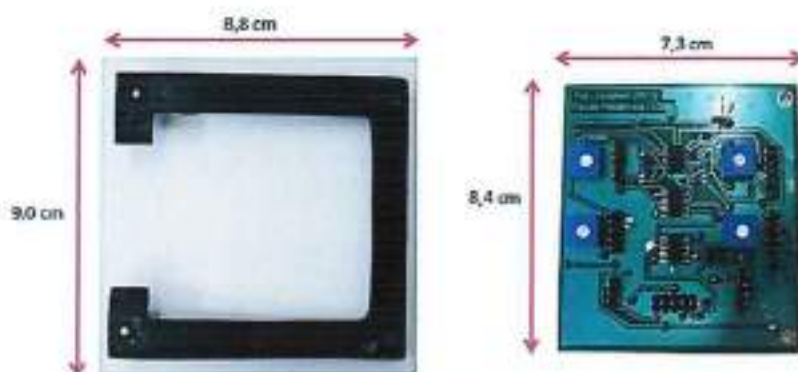


Figura 3-3 Suporte de apoio da placa da câmera e a placa, respectivamente.

3.2.2 Haste da Câmera

Dado que a câmera é responsável pelo 'visão' do carrinho, a haste que a suporta deveria ser o mais rígida possível, evitando assim ruídos na imagem devido à vibração da mesma em relação ao chassi. Além disso, para permitir um estudo da melhor posição da câmera tanto em ângulo quanto em altura, a haste, juntamente com o suporte onde a câmera fica fixa, deveria apresentar flexibilidade de posicionamento. Dessa forma, a seguinte estrutura foi feita para atender às essas necessidades (Figura 3-4):



Figura 3-4 Haste e suporte da câmera.

O friso no meio da haste possibilita o ajuste do suporte em qualquer altura do intervalo, e o parafuso borboleta que prende o suporte no friso permite variação angular. Além disso, há quatro apoios que descem da haste até a base, distribuindo melhor as forças atuantes na estrutura durante o movimento do carrinho. Testes anteriores sem esses apoios em forma de treliça evidenciaram um movimento vertical da haste muito maior do que com os apoios.

A haste foi fixada no chassi por meio da abertura de quatro novos furos para colocação de parafusos e travamento com porcas, apresentando um resultado muito satisfatório em relação às características desejadas.

3.2.3 Placa dos Sensores de Velocidade

Devido ao tamanho reduzido da placa de sensores de velocidade (vide Figura 3-5 abaixo), e também à necessidade de se conectar fios das quatro rodas nela, esta placa foi fixada ao chão do chassi por meio de fita dupla face espessa. Tal posicionamento facilitou a instalação das conexões, além de ajudar no design, visto que evita mais fios expostos no carrinho.

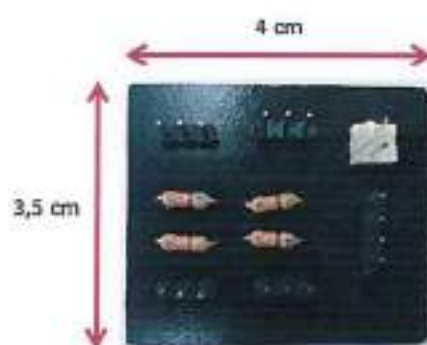


Figura 3-5 Placa dos sensores de velocidade.

3.2.4 Suporte da placa de alimentação e da placa dos motores

Assim como no apoio da câmera, este suporte (Figura 3-7) também deveria estar acima de um determinado nível: a altura da bateria. Novas peças foram utilizadas para fornecer a altura desejada na parte próxima à haste da câmera. Para o lado oposto foram feitos furos bem maiores para que o apoio pudesse se aproveitar de hastes já existentes no chassi. Nela são apoiadas duas placas: a de alimentação e a da ponte-H dos motores (Figura 3-6), tomando-se o devido cuidado para que não interfiram na estrutura de suspensão traseira.

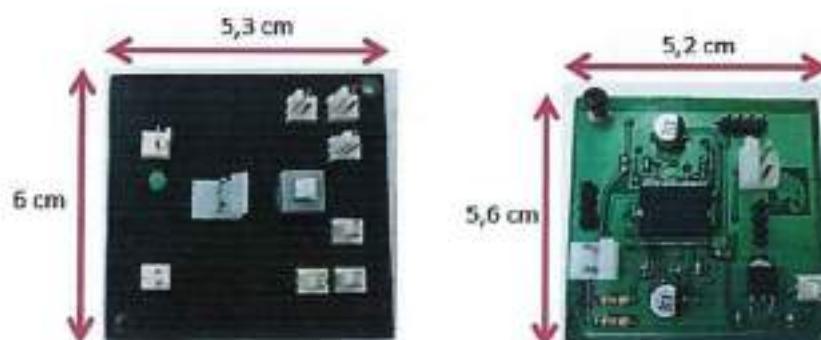


Figura 3-6. Placas de alimentação e da ponte-H dos motores, respectivamente.

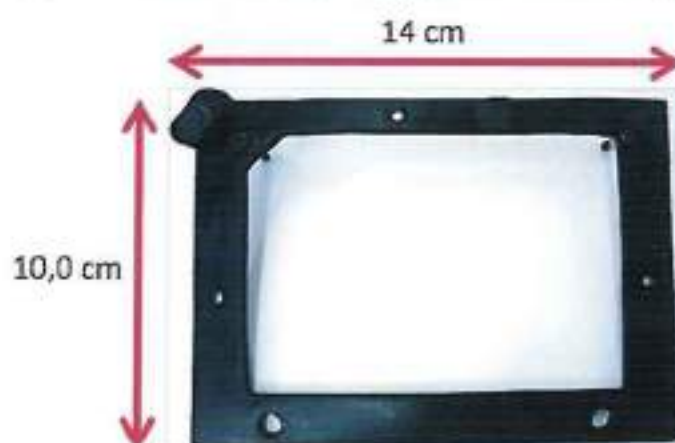


Figura 3-7. Apoio das placas de alimentação e dos motores.

3.2.5 Suporte da Placa da Freescale

Para a construção desse apoio houve uma dificuldade adicional, que foi a falta de estruturas do próprio chassi na altura que a placa poderia ficar utilizando-se as mesmas hastes em que foi encaixado o apoio das placas de alimentação e dos motores. Além disso, estas hastes estão na parte fixa do chassi, sendo que o apoio para a placa se estenderia para a parte da suspensão. Para resolver esse problema foram adicionadas duas peças em L, juntamente com duas espumas para que o conjunto não perdesse a mobilidade fornecida pela suspensão. Essas espumas estão presas nas pequenas estruturas internas ao apoio, conforme pode ser visto a seguir na Figura 3-8:

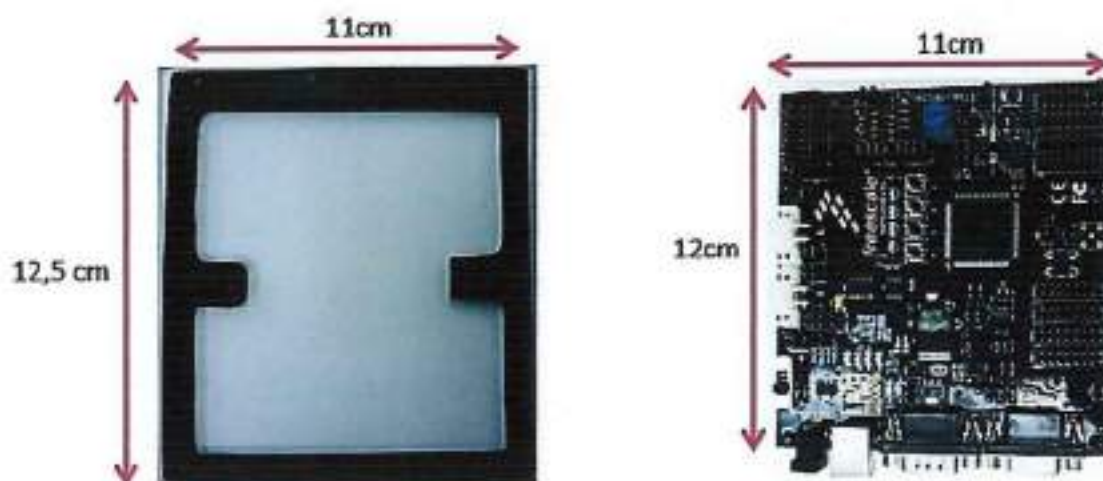


Figura 3-8. Apoio da placa da Freescale (à esquerda) e a própria placa (à direita).

3.3 Rodas

Como será explicado no Capítulo 8, para o método adotado para a medição da velocidade do carrinho é necessário que haja ímãs em suas rodas. O campo magnético destes ímãs é detectado pelos Sensores Hall, que ficam posicionados próximos às rodas. Estes dados são enviados ao microcontrolador para que o mesmo possa efetuar os controles necessários.

Para minimizar a influência que os ímãs teriam no movimento do carrinho, foram utilizados ímãs de neodímio. Estes ímãs, mesmo em pequenas dimensões, possuem campo magnético suficiente para ser detectado através da espuma e da borracha dos pneus. Isto permitiu montar os ímãs no interior da roda, como mostra a Figura 3-9 abaixo.

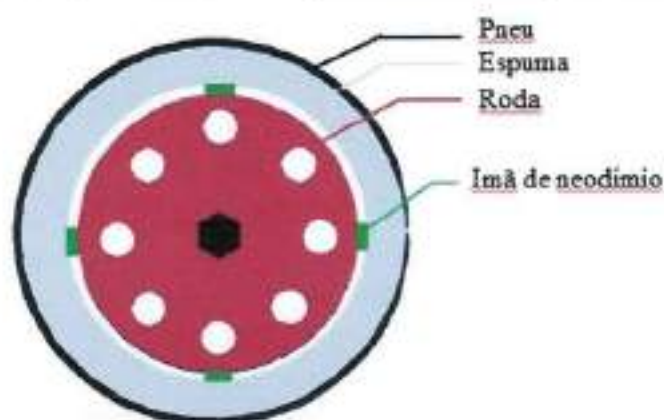


Figura 3-9 - Esquema da montagem das rodas com os Ímãs de Neodímio

Como a camada de espuma preenche a maior parte da roda, os ímãs ficam bem alojados e não influenciam no movimento das rodas do carrinho.

Utilizando os sensores Hall UGN3132 conseguimos uma precisão elevada na medida de velocidade já que os mesmos passam de alto para baixo na presença de um dos polos do ímã e só mudam de baixo para alto na presença do outro polo, evitando ruídos de transição. Também por este motivo, os quatro ímãs foram posicionados com faces externas na ordem Norte-Sul-Norte-Sul, criando uma onda retangular (quadrada em velocidade constante) onde o tempo entre duas bordas era o tempo em que o carrinho percorreu um quarto da circunferência da roda. Como esta distância é constante tem-se uma relação simples para obter a velocidade a partir do tempo medido entre as bordas.

4. Circuitos Impressos

Neste capítulo serão descritos os circuitos elétricos que foram desenvolvidos para este projeto. São eles: alimentação, controle dos motores, filtragem da câmera e a placa dos sensores Hall. O circuito impresso do microcontrolador foi adquirido pronto e sua documentação pode ser encontrada no site da Freescale Semiconductor com o nome de TRK5604B.

4.1 Placa de Alimentação

O circuito de alimentação tem como objetivo receber a tensão da bateria (7,2V) e redistribuir esta tensão de maneira apropriada. Assim, ela possui 2 reguladores de tensão de 5,0V onde as cargas de 5V (filtragem da câmera, sensores Hall, *Bluetooth*) devem ser distribuídos. Também há duas saídas de 7,2V para a placa do microcontrolador e para a placa de controle dos motores. O esquemático da mesma e o desenho de seu circuito impresso podem ser vistos abaixo nas Figura 4-1 e 4-2, respectivamente:

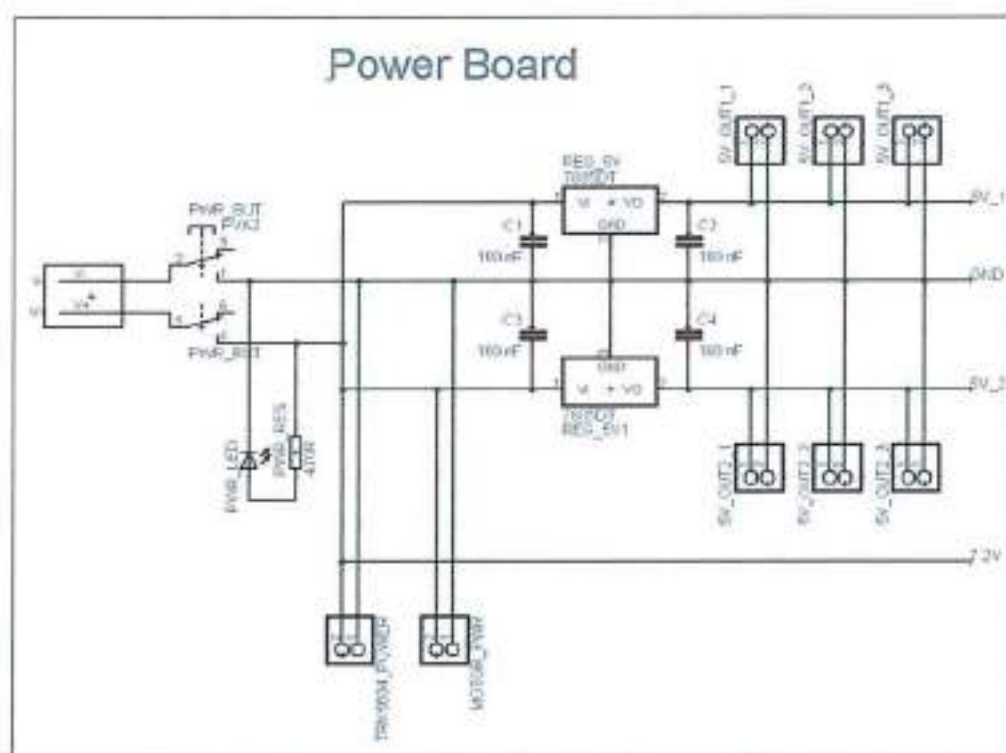


Figura 4-1 - Esquemático da Placa de Alimentação

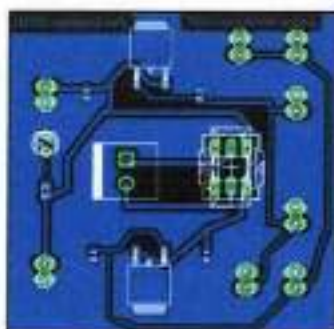


Figura 4-2 - Desenho do Circuito Impresso da Placa de Alimentação

Este circuito é composto pelos seguintes componentes:

- 4 capacitores de 100nF
- 1 resistor de 470Ω
- 2 reguladores de voltagem de 5V LM7805;
- 1 Botão do tipo Liga/Desliga com 4 terminais
- 9 Conectores de alimentação 1x2
- 1 LED verde 3mm

4.2 Controle dos Motores

A maior parte dos circuitos elétricos do carrinho utilizam uma tensão de 5V. Contudo, motores exigem uma tensão maior (7,2V) para possuir maior potência e também consomem muita corrente (cerca de 1,3A) quando travado (carga máxima). Devido a esta maior potência, controla-los com os circuitos digitais utilizados não é possível sendo necessária a utilização de circuitos conhecidos como Pontes-H.

Há diversos modelos de Ponte-H no mercado. Para este projeto foi escolhido modelo MC33932, também da Freescale Semiconductor. Os motivos para escolha são a existência de duas Pontes-H no mesmo chip, o fato do fabricante fornecer amostras destes chips e também facilidades técnicas. Entre estas facilidades estão a possibilidade de desativar as duas saídas para os motores (tristate) ou toda a Ponte-H, o controle da direção de rotação dos motores, os freios dinâmicos através da imposição da mesma tensão nas duas saídas para os motores e pinos que indicam a corrente de *feedback* do motor. Esta última é interessante para saber quanta potência o motor está consumindo e, assim, identificar falhas ou travamento nos mesmo. Além de tudo isso seu empacotamento é no estilo SMD, facilitando seu posicionamento na placa e reduzindo espaço e peso no circuito.

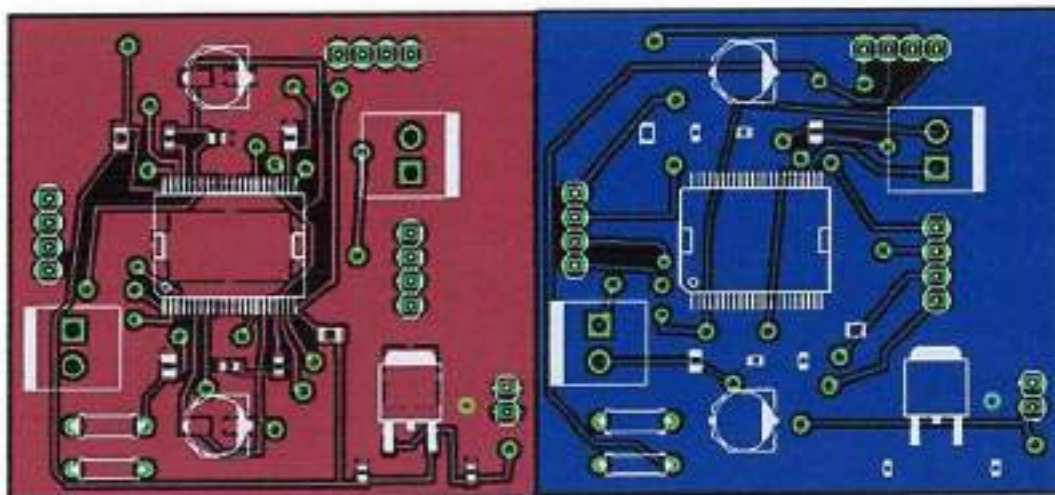


Figura 4-4 - Desenho do Circuito Impresso de Controle dos Motores

Este circuito é composto pelos seguintes componentes:

- 4 capacitores de 100nF
- 2 capacitores de 33nF
- 2 capacitores de 1 μ F
- 2 capacitores de 100mF
- 2 resistores de 270 Ω
- 2 resistores de 10k Ω
- 1 regulador de voltagem de 5V LM7805;
- 1 Ponte-H MC33932
- 3 barras de pinos 1x4
- 3 barras de pinos 1x2

4.3 Placa dos Sensores Hall

Para organizar os sinais, os demais cabos e alojar os transistores de *Pull-Up* necessários para o funcionamento dos sensores Hall, foi elaborada uma placa de unificação dos quatro sensores Hall.

Esta placa é simples e consiste apenas de resistores e pinos (os sensores Hall estão fixados próximos às rodas e se conectam a esta placa através de cabos de 3 vias). O esquemático da placa e o desenho de seu circuito impresso podem ser vistos a seguir (Figuras 4-5 e 4-6):

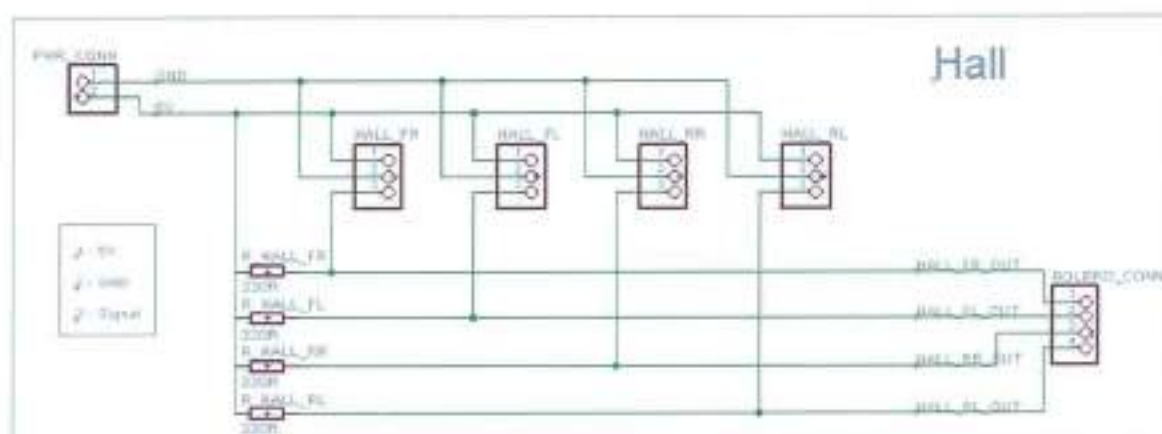


Figura 4-5 - Esquemático da Placa de Controle dos Motores

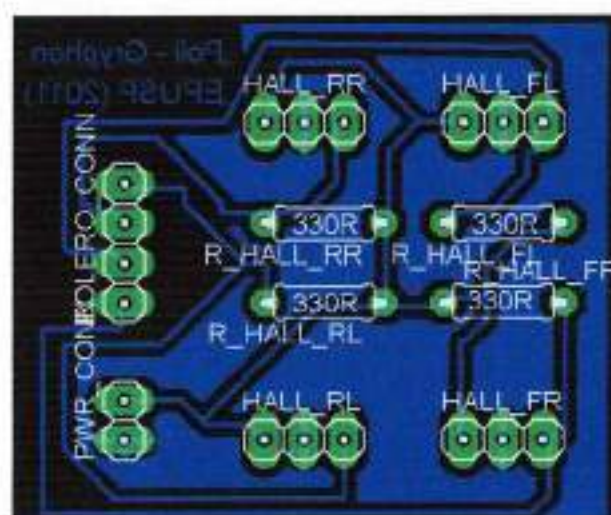


Figura 4-6 - Desenho do Circuito Impresso de Controle dos Motores

Este circuito é composto pelos seguintes componentes:

- 1 barra de pinos 1x4
- 4 barras de pinos 1x3
- 1 barra de pinos 1x2
- 4 resistores de 330Ω

4.4 Placa da Filtragem da Câmera

Esta placa deve ser utilizada caso se deseje utilizar o método de obtenção de imagem por largura de pulso descrito no capítulo 10. Caso contrário, a câmera e o servo podem ser ligados diretamente na placa de alimentação e na placa do microcontrolador.

Esta placa recebe a saída analógica da câmera e passa por um comparador que também está conectado a um potenciômetro. Desta maneira pode-se configurar a partir de qual tensão o sinal é considerado baixo e a partir de qual tensão o sinal é considerado alto.

A saída deste comparador passa então por um *Flip-Flop* tipo D que recebe o mesmo sinal de *clock* enviado para as câmeras. Isto evita que o sinal entre dois pixels, onde há mais ruído, seja repassado para o microcontrolador.

Além destes há também pinos para a conexão de até 4 câmeras e também do servo utilizado para virar as rodas da frente. O esquemático da placa e o desenho de seu circuito impresso podem ser vistos a seguir nas Figuras 4-8 e 4-7, respectivamente.

Este circuito é composto pelos seguintes componentes:

- 5 barras de pinos 1x5
- 1 pino 1x1
- 1 barra de pinos 1x2
- 1 barra de pinos 1x3
- 1 barra de pinos 2x5
- 1 resistor 470 Ω
- 8 resistores de 330 Ω
- 4 resistores de 10k Ω
- 4 potenciômetros de 1k Ω
- 1 capacitor de 100nF
- 1 LED verde de 3mm
- 1 circuito LM339
- 2 circuitos 74HC74

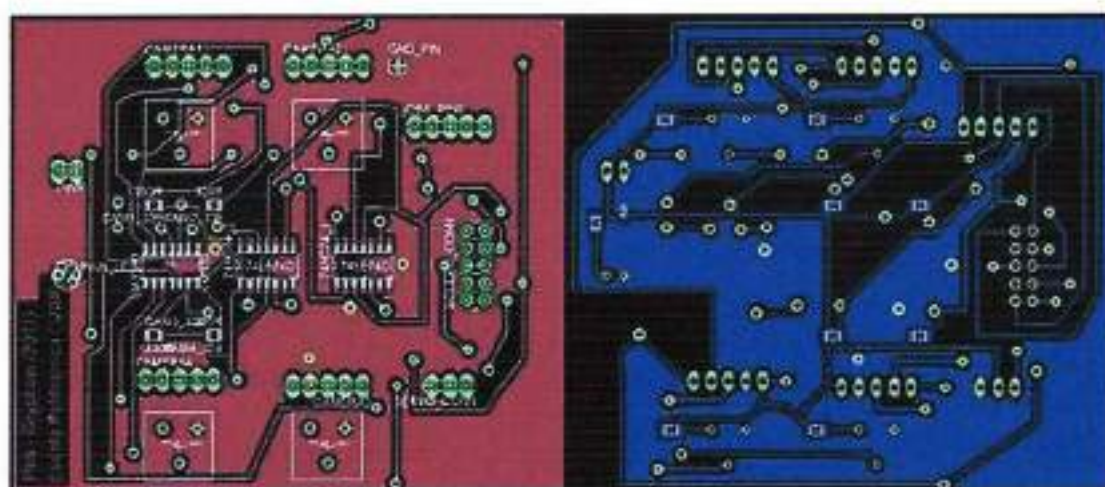


Figura 4-7 Desenho do Circuito Impresso da Placa de Filtragem da Câmera

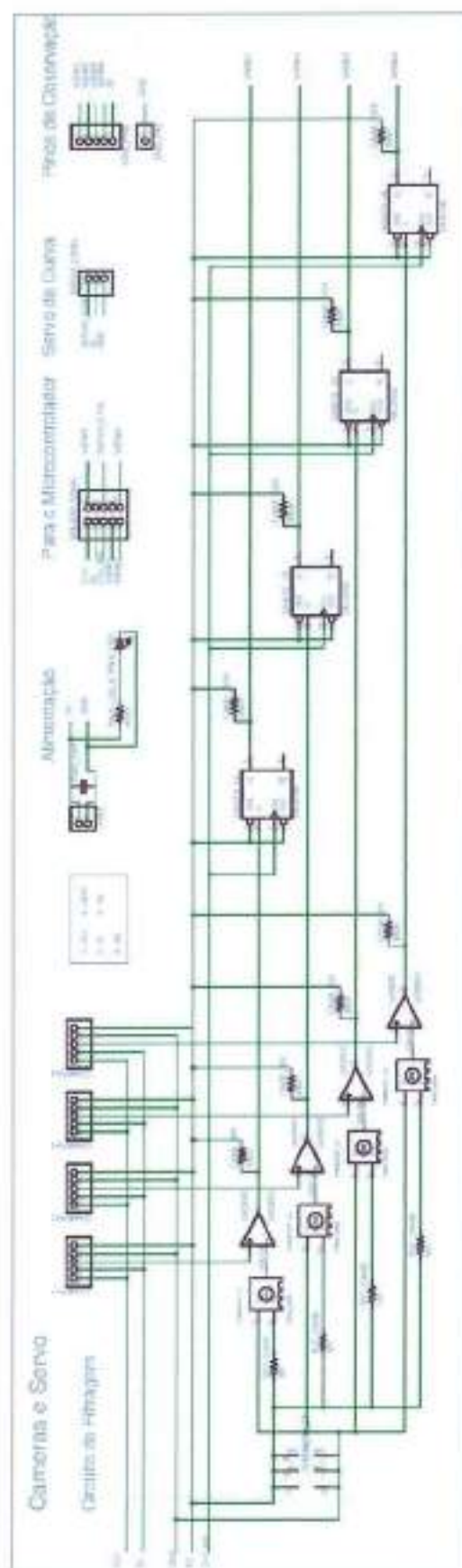


Figura 4-8 Esquemático da Placa de Filtragem da Câmera.

5. Conceitos Básicos

5.1 Sinal PWM

O sinal PWM (*Pulse Width Modulation*) ou Modulação por Largura de Pulso é um sinal muito utilizado no controle dos atuadores deste projeto. Assim sendo, é interessante dedicar um breve subcapítulo para explicar seu funcionamento.

O sinal PWM possui dois atributos importantes: período e *duty cycle*. O período T , como esperado, é o período em que o sinal se repetirá. Já seu *duty cycle* (d) é a porcentagem do período T em que o sinal fica no nível alto. Ou seja, o sinal ficará alto durante $d \cdot T$ e baixo durante $(1-d) \cdot T$. A Figura 5-1 abaixo demonstra seu funcionamento para *duty cycle* de 25%, 50% e 75%:

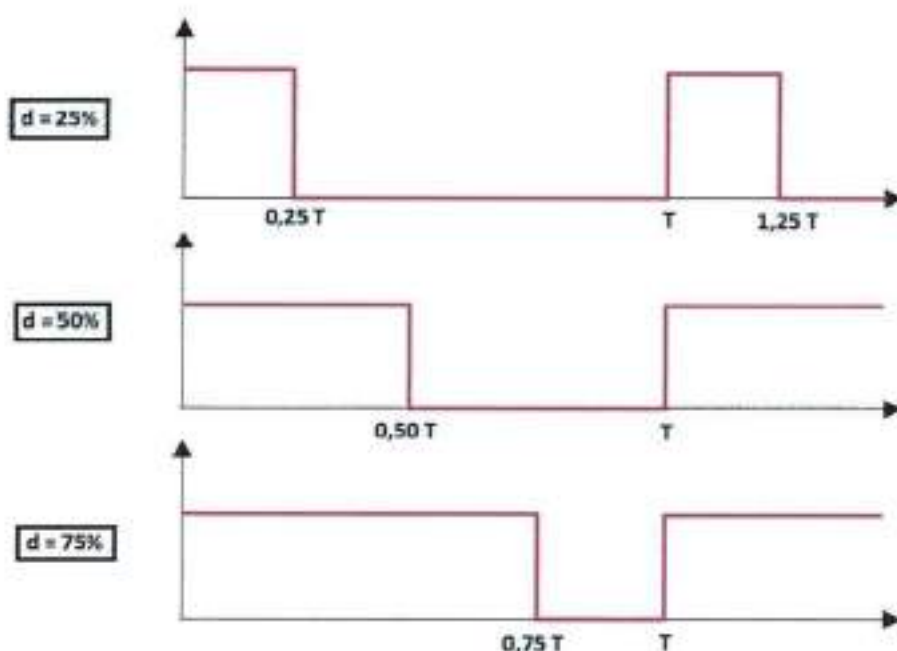


Figura 5-1 - Sinal PWM

5.2 Conversão A/D

Os microcontroladores são sistemas digitais e, portanto atuam na maior parte do tempo com entradas binárias. Contudo, muitas vezes os microcontroladores devem controlar (ou medir) grandezas analógicas. Para isto o funcionamento normal de suas portas não irá servir e o método da conversão analógica/digital deve ser utilizado.

Uma conversão analógica/digital (doravante denominada como A/D) é a maneira como o microcontrolador interpreta sinais analógicos. No caso do MPC5604B, seu conversor

possui uma precisão de 10 bits, ou seja, o resultado da conversão é um número entre 0 e 1023 ($2^{10} - 1$). No caso desse projeto a tensão de referência é de 4,93 volts (medidos na placa), ou seja, caso a tensão de entrada seja de 4,93 volts, o conversor irá resultar no valor 1023. Caso ela seja de 0 volts, o conversor irá resultar em 0. Os demais valores entre 0V e 4,93V resultarão em um valor interpolado linearmente entre estes dois. De maneira geral o resultado N é um valor tal que o teto (ou o chão, dependendo da tecnologia utilizada) de $N \cdot (4,93/1024)$ seja o mais próximo da tensão de entrada. Existem diversas tecnologias utilizadas para se fazer isso, a mais simples é um contador conectado a um conversor digital-analógico (pode ser implementado através de chaves (bits) e divisores de tensão adequadamente regulados para cada bit) que é comparado com a tensão de entrada. Quando a conversão digital-analógica do contador ultrapassa a tensão de entrada o valor contido nele é o resultado da conversão [6].

Sabendo disso, vê-se que a conversão A/D necessita de um tempo para cada conversão (para que o contador atinja o valor certo). Este tempo é proporcional à velocidade com que este contador é incrementado e seu resultado comparado.

Quando a conversão A/D é utilizada o tempo necessário para a conversão deve ser considerado no projeto já que a necessidade de muitas conversões pode deixar a medição das variáveis de controle muito lenta.

5.3 Câmeras para a obtenção da pista

São duas as possibilidades de câmeras para a detecção da pista: câmera linear ou matricial. A linear tem entre as suas vantagens, a velocidade, visto que a quantidade de informação captada é reduzida, e maior facilidade/velocidade de processamento no microcontrolador. Já a câmera matricial, apesar de ter um maior requerimento de processamento, fornece uma visão muito mais ampla do ambiente.

5.4 Sinal da câmera

O princípio de funcionamento de câmeras CCD (*Charge Coupled Device*) é muito simples: a luz ambiente entra através da lente e incide na matriz (ou linha) de sensores, que são uma espécie de capacitores. Estes são carregados de acordo com a intensidade da luz, todos ao mesmo tempo. Por fim, o sinal de saída da câmera se constitui do descarregamento

desses capacitores, um de cada vez, gerando uma saída serial. Dessa forma, a câmera oferece também a facilidade de operação, pois é composta por apenas três sinais: VDD, GND e Saída da imagem. Mais detalhes técnicos das câmeras encontram-se no Anexo I.

5.5 Câmera Matricial X Câmera Linear

5.5.1 Câmera Matricial

A câmera matricial obtém uma imagem como seu nome diz: matricial. Ou seja, ela possuirá duas dimensões de pixels. Cada elemento da matriz de pixels armazenará o valor da intensidade luminosa daquele ponto do sensor. Por exemplo, uma câmera 3x3 que fotografasse uma linha vermelha em um fundo branco poderia apresentar a seguinte matriz (Figura 5-2) de intensidade luminosa (8 bits):

255	72	255
255	72	255
255	72	255



Figura 5-2 - Exemplo Câmera Matricial Monocromática

No exemplo acima vemos que cada pixel que contém apenas a cor branca possui o valor 255. Fisicamente cada pixel do sensor pode ser entendido como um capacitor que se carrega quando exposto à luz [7]. Com isso, os pixels que estão focados em cores brancas (que refletem toda a luz) acabam se carregando mais do que os pixels focados na cor vermelha (que refletem apenas o vermelho) e, por isso, os pixels de cor vermelha acabam possuindo um valor menor na matriz.

Outra constante que afeta os valores de cada pixel é o tempo de exposição. Se expusermos o pixel durante uma grande quantidade de tempo eles serão carregados até saturarem. Os pixels com foco no vermelho irão carregar mais lentamente devido a menor intensidade luminosa, mas se expostos por tempo suficiente também atingirão a tensão de saturação fazendo com que a matriz seja constante e igual a 255.

Já se o tempo de exposição for muito pequeno os pixels não terão tempo de serem carregados adequadamente, resultando em diferenças de valores muito baixas entre o branco e a linha vermelha, (branco = 12 e vermelho = 8, por exemplo). Isto seria muito ruim, pois qualquer ruído no sinal da imagem já faria confundir detalhes dela.

Com isso em mente, tem-se que o tempo de exposição do sensor para cada foto deve ser o suficiente para que a imagem apresente uma boa nitidez, mas sem que os pixels acabem saturando.

Em geral as câmeras matriciais já vêm com este tempo configurado de fábrica. No modelo pensado para este projeto (CM-26N/P) o tempo total para obter e transmitir uma imagem é de cerca de 16ms.

É importante lembrar que, por se tratar de um sistema embarcado em movimento, durante o tempo decorrido entre uma imagem e outra (período de atualização da imagem) o carrinho estará se movendo. Ou seja, no instante seguinte à obtenção da imagem da câmera, esta imagem passa a estar desatualizada com relação à posição atual do carrinho. Este erro é inerente a todas as câmeras (pois todas necessitam, no mínimo, do tempo de integração), mas quanto menor o tempo de integração (maior taxa de atualização) menor será o erro. Como este tipo de câmera normalmente vem de fábrica com uma taxa de atualização fixa de 60Hz (16ms entre as imagens), ela possui uma desvantagem com relação à câmera linear (que será abordada mais adiante).

Outro ponto negativo da câmera matricial é com relação à dificuldade de sincronização quando comparada à câmera linear. A câmera matricial necessita de diferentes sincronizações ocorrendo simultaneamente. Isto porque sua saída é analógica, ou seja, é necessário sabermos quando se começa um pixel, quando se começa uma linha, quando se começa uma matriz de cor e quando se começa uma imagem. Para isto a câmera emite diferentes formas de sinal e temos de detectar cada uma delas, por exemplo, o sinal de sincronização horizontal da Figura 5-3:



Figura 5-3 - Sinal Analógico de Sincronização Horizontal de uma Câmera Matricial

O sinal acima é um exemplo do sinal de sincronização horizontal. A detecção deste sinal é uma das fases necessárias no tratamento de uma imagem matricial, mas deixa de ser necessário quando trabalhamos com a câmera linear.

Um último detalhe da câmera matricial é com relação ao fato de ser colorida. Até o momento levamos em consideração a intensidade luminosa, mas algumas câmeras captam as cores da imagem. Para isso ela possui pixels especializados para capturar, por exemplo, as cores: VERMELHO, VERDE e AZUL (padrão RGB – *Red Green Blue*).

O funcionamento de uma câmera colorida é idêntico ao de uma monocromática (explicado acima) com a diferença de que ao invés de uma única matriz de intensidade luminosa são passadas três matrizes, uma com relação à intensidade do vermelho, outra com relação à intensidade do verde e uma última com relação à intensidade do azul. As três matrizes para a linha vermelha acima seriam, por exemplo (Figura 5-4):

Vermelho			Verde			Azul					
255	255	255	255	10	255	255	3	255			
255	255	255	255	10	255	255	3	255			
255	255	255	255	10	255	255	3	255			

Figura 5-4 - Exemplo Câmera Matricial Colorida

Pelo fato da pista controlada ser branca e preta, a detecção de cores não é um fator relevante na escolha da câmera e este projeto não abordará o tratamento de imagens coloridas. Caso projetos futuros utilizem câmeras matriciais coloridas, mas desejem apenas a intensidade luminosa da imagem sugere-se utilizar a ideia encontrada em [8] onde apenas uma das matrizes de cor é utilizada e as outras descartadas. Como o branco possui todas as cores e o preto nenhuma, qualquer uma das matrizes deverá servir, diferente do exemplo acima onde a matriz vermelha não é capaz de detectar uma linha vermelha.

Assim vemos que apesar de apresentar um maior campo de visão a câmera matricial possui um tempo maior entre as imagens (aproximadamente 16ms por imagem), uma maior complicação em sua obtenção devido ao sincronismo horizontal e vertical e também uma perda de tempo devido ao fato de obter e transmitir as 3 matrizes de cores.

5.5.2 Câmera Linear

A câmera linear é, basicamente, uma câmera matricial com uma única linha. Ou seja, seu sensor é capaz de obter uma única linha da imagem o que resulta em um campo de visão muito reduzido quando comparada às câmeras matriciais que chegam a ter centenas de linhas.

Contudo, a simplicidade de sua obtenção neste projeto e a possibilidade de configurar sua taxa de atualização são vantagens interessantes para este projeto. Além disso, a desvantagem do campo de visão pode ser atenuada como será visto a seguir. Como o Anexo II mostra, através de uma matriz AHP, que a câmera linear é mais adequada do que a câmera matricial para este projeto.

A câmera escolhida para este projeto (Parallax TSL1401) é monocromática, possui 128 pixels de resolução, e um exemplo de como sua imagem é lida pode ser encontrado na Figura 5-5 abaixo:

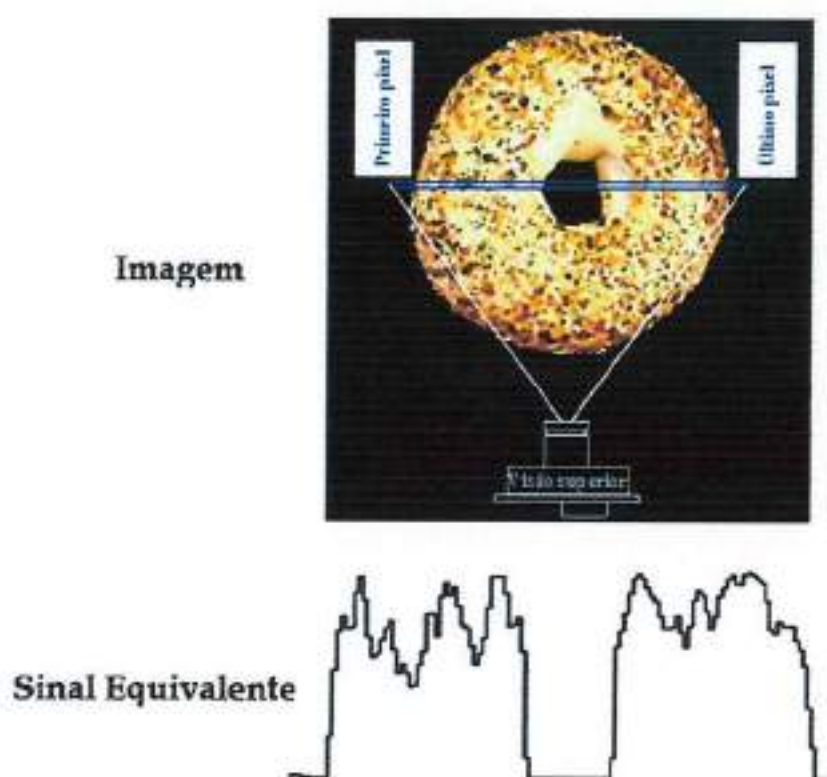


Figura 5-5 - Exemplo de Sinal da Câmera Linear (Freescale Application Note Linescan Camera Mode).

Como se vê na imagem acima, seu sinal é, como na câmera matricial, proporcional à quantidade luz, mas apenas em uma linha. Vê-se que o sinal detecta a primeira parte da rosquinha, vai para zero enquanto está no buraco e detecta a segunda parte da rosquinha caindo a zero no final (pois o buraco e o final estão pretos).

5.6 Filtro Gaussiano

O filtro de Gauss é um filtro que basicamente calcula uma média ponderada entre os pixels da imagem. Recebe o nome de Gaussiano pelo fato de usar valores da curva gaussiana como pesos para a ponderação.

A Figura 5-6 a seguir exemplifica a aplicação do filtro:

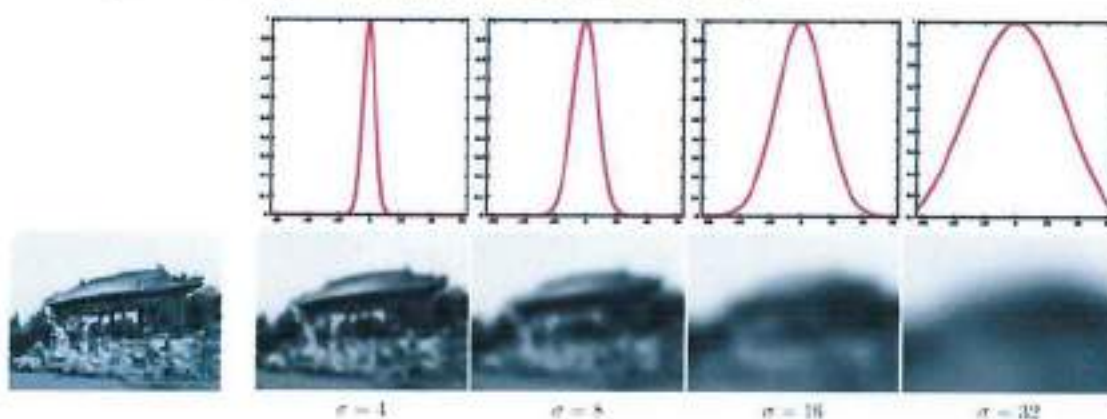


Figura 5-6 - Exemplo da Aplicação do filtro de Gauss variando σ [9].

É interessante perceber que estamos acostumados a analisar a nitidez da imagem naturalmente. Assim sendo temos a impressão de que as imagens onde o filtro foi aplicado são piores do que a original.

Contudo, trabalhando com imagens é importante ter em mente que os filtros não servem apenas para deixar uma imagem visualmente mais bonita ou mais nítida, mas para ressaltar alguma característica dela que nos interessa. No caso do filtro gaussiano ele faz com que a imagem perca a nitidez, porém faz também que os ruídos sejam consideravelmente reduzidos. O fator de configuração deste filtro está no σ da gaussiana, ou seja, seu desvio padrão. Quanto maior o σ maior será a “abertura” da curva e, portanto, maior será o número de pixels levados em consideração na média. Este é o motivo dela ficar mais borrada com o aumento do σ .

Para este projeto a função de Gauss considerada será a de apenas uma dimensão:

$$G(x) = \frac{1}{\sigma\sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}}$$

5.7 Algoritmo de Canny

O algoritmo de Canny é a utilização do filtro de Gauss com um filtro diferencial. O filtro de Gauss serve para eliminar os ruídos enquanto o diferencial irá detectar as bordas. Além deles ainda é possível fazer um refinamento da imagem através da maximização ou minimização dos pixels resultante dos 2 filtros o que causa um afinamento da linha.

A aplicação de filtros digitais em imagens se dá através de máscaras. É definida uma máscara que representa aquele filtro e, então, se faz a convolução daquela imagem com aquela máscara. Assim sendo, os 2 filtros do algoritmo de Canny podem ser calculados da seguinte maneira:

$$(I * G(x))'$$

Contudo, por ser uma convolução, a seguinte simplificação é válida:

$$G'(x) * I$$

Com esta simplificação podemos calcular $G'(x)$ e aplicar uma única máscara a imagem ao invés de duas. Assim, temos que:

$$G'(x) = \frac{-x}{\sigma^3 \sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}}$$

Sabe-se da função de Gauss que, a partir de 3σ seus valores começam a ser desprezíveis. Assim, para pixels que se distanciarem mais de 3σ do pixel central o peso considerado será 0.

5.8 Giroscópio

O giroscópio é um sensor passivo (mede a energia vinda do ambiente) e proprioceptivo (tem sensibilidade própria). É um sensor de rumo que preserva a sua orientação em relação a um quadro fixo de referência [7], sendo, portanto, capaz de fornecer a velocidade angular em relação a um determinado eixo do objeto ao qual está acoplado. Desta forma, um giroscópio com três eixos é capaz de fornecer o quanto o objeto levanta ("pitch"), o quanto se desvia da rota ("yaw"), e o quanto rola ("roll"). Tais movimentos são ilustrados na Figura 5-7 abaixo:



Figura 5-7 - Os seis graus de liberdade de um objeto. Adaptada de [10]

Dada a natureza deste projeto, o movimento mais relevante a ser trabalhado é o quanto o carrinho desvia-se da rota, ou seja, a velocidade angular em relação ao seu eixo z (que devido ao modelamento adotado até então, aponta de cima para baixo do carrinho).

5.9 Controle de Estabilidade - Conceito

O controle de estabilidade é um sistema que ainda não é comum nos carros populares do Brasil, mas a Bosch o fabrica (com o nome de ESP®) desde 1995 e o mesmo será item de fábrica obrigatório nos carros vendidos na União Europeia a partir de 2014 [11].

Esta obrigatoriedade se deve ao fato do acréscimo de segurança que este item dá ao veículo, já que com ele o carro passa a obedecer aos movimentos do motorista da forma mais fiel possível, sem que haja problemas de derrapagens em curvas bruscas.

Quando um motorista está em alta velocidade e se depara com uma curva fechada ou um obstáculo, seu reflexo o faz virar o volante rapidamente. Neste instante, devido à velocidade e à inércia do carro, o fenômeno conhecido como subviragem (*"understeering"*) passa a acontecer: apesar das rodas estarem viradas o suficiente para fazer a curva, a inércia do veículo faz com que força centrípeta causada pelo atrito das rodas com o chão não seja o suficiente para fazer a curva com o raio desejado, consequentemente aumentando o raio da curva realizada (Figura 5-8).

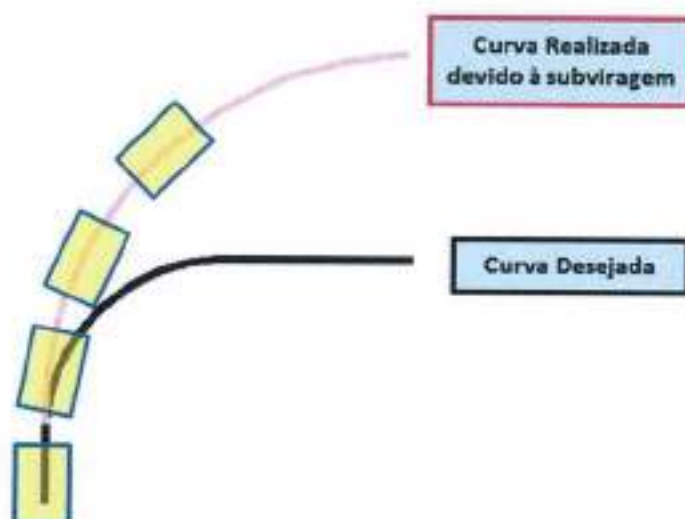


Figura 5-8 - Efeito da Subviragem ("Understeering").

Para corrigir este efeito, obviamente indesejável, o controle de estabilidade atua freando a roda traseira direita (no caso de curvas a direita) e esquerda (no caso de curvas a esquerda). Esta frenagem em apenas uma das rodas gera um torque adicional no veículo auxiliando, com a força de atrito gerada, que as rodas da frente sigam a trajetória imposta pelo motorista. A Figura 5-9 ilustra o torque gerado para a direita (primeiro quadro) e para a esquerda (segundo quadro) com o controle de estabilidade:

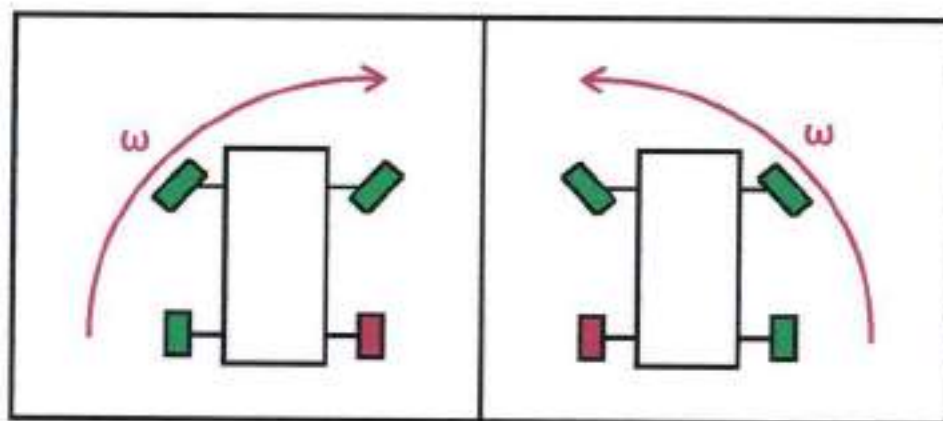


Figura 5-9 - Correção da Subviragem ("Understeering") pelo controle de Estabilidade (roda em vermelho freada).

Quando o veículo realiza a virada mais brusca um segundo efeito passa a ocorrer, este conhecido como sobreviragem ("Oversteering"), popularmente conhecido como "sair de traseira". Este acaba sendo um efeito colateral da primeira atuação do controle de estabilidade já que o mesmo adiciona um torque ao movimento do veículo no início da curva. O efeito da subviragem pode ser visto abaixo, na Figura 5-10:

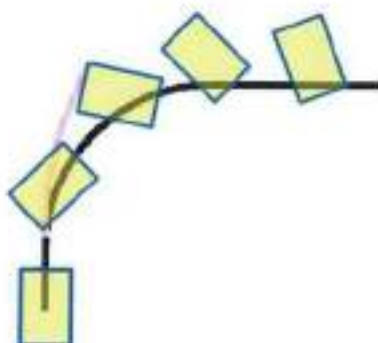


Figura 5-10 - Efeito da Sobreviragem ("Oversteering")

Para corrigir este efeito uma técnica semelhante é aplicada. A roda da frente oposta à curva passa a ser freada, gerando um torque contrário ao gerado inicialmente e, portanto, diminuindo a velocidade angular excessiva. A Figura 5-11 abaixo ilustra o funcionamento:

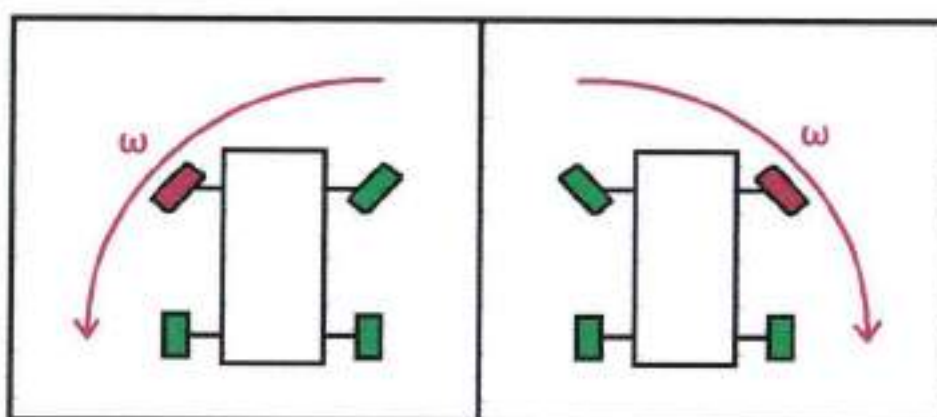


Figura 5-11 - Correção da sobreviragem ("Oversteering") pelo controle de estabilidade (roda em vermelho freada).

Assim, após essas duas fases de atuação (correção da subviragem seguida da correção da sobreviragem), o controle de estabilidade permite que o veículo reaja melhor aos comandos do motorista, mantendo uma trajetória mais estável e segura.

Para detectar se a condição é normal, de subviragem ou de sobreviragem, o controle de estabilidade faz o controle da velocidade angular do veículo, utilizando como referência a velocidade angular que o veículo deveria ter para determinada posição das rodas frontais. Por exemplo, se elas estiverem alinhadas, a velocidade angular de referência será zero já que nesta posição o motorista deseja andar em linha reta.

6. Modelos

6.1 Modelo Cinemático do Carrinho

A referência [8], escrita por Rodney A. Busquim e Silva, possui um modelo muito interessante para o nosso projeto. Isto porque, como será visto adiante, ele é um modelo SISO (*Single Input Single Output*) e com polos nulos. Ou seja, um simples controle proporcional já é capaz de anular este erro.

Além disso, diferente de outros modelos encontrados, a variável de controle não é a distância do centro de massa do carrinho até a pista, mas de um ponto à frente do carrinho até a pista naquele ponto (Figura 6-2).

Isso significa que, ao se anular o erro ξ , se faz com que o carrinho passe pelo ponto C da trajetória. Isto é possível porque o ponto C está à frente e é um ponto "futuro". Se o ponto C estivesse na região do centro de massa (como na maioria dos modelos) as rodas deveriam girar de 90° para que o carrinho anulasse efetivamente o erro. A Figura 6-1 abaixo representa as duas situações, sendo as setas vermelhas o erro a ser anulado em cada caso, e as setas verdes a direção que o carrinho deveria se mover para anular este erro.

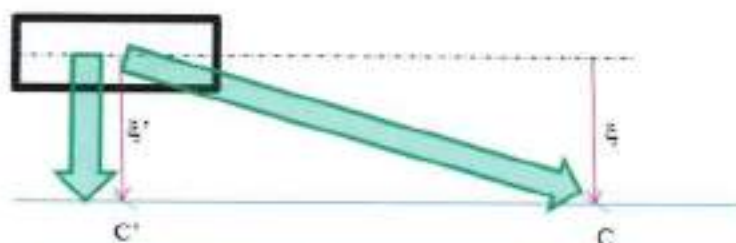


Figura 6-1. Diferença do erro no centro de massa e de um ponto à frente.

Olhando a figura acima fica claro que o carrinho não conseguirá anular o erro ξ' devido ao fato de que suas rodas viram, no máximo, 30° . Já o erro ξ parece ser perfeitamente possível de ser anulado com a escolha correta de velocidade e ângulo das rodas frontais.

Esclarecendo as vantagens do modelo, pode-se partir para a sua dedução (que também pode ser encontrada na referência [8]). As variáveis utilizadas seguirão o seguinte esquema da Figura 6-2.

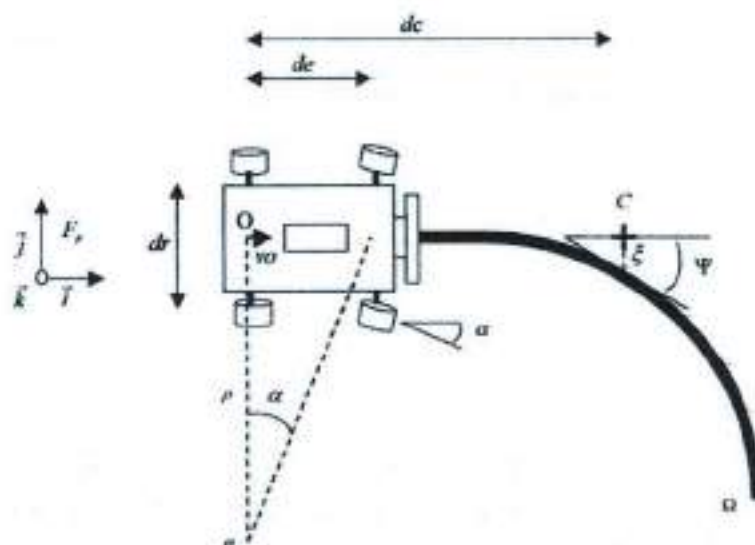


Figura 6-2. Diagrama do Modelo (Referência [8])

Sendo v_o a velocidade do ponto O (centro do eixo traseiro), ρ o raio do centro instantâneo de rotação (doravante denominado CIR), α o ângulo que as rodas da frente fazem e ω a velocidade angular com que carrinho rotaciona em torno do CIR, as seguintes relações podem ser deduzidas:

$$V_o = \omega * \rho$$

$$\tan \alpha = \frac{de}{\rho} \rightarrow \rho = \frac{de}{\tan \alpha}$$

$$V_o = \omega * \left(\frac{de}{\tan \alpha} \right)$$

$$\therefore \boxed{\omega = \left(\frac{V_o}{de} \right) * \tan \alpha}$$

Lembrando que o atuador de posição utilizado (servomotor conectado às rodas frontais) atua no ângulo α , se v_o for considerado constante, tem-se uma função $w(\alpha)$ que pode ser considerada como entrada do sistema (saída do controlador).

A segunda parte da dedução diz respeito à vantagem explicitada acima. É preciso encontrar o valor de ξ em função de parâmetros do nosso sistema. Considerando um referencial não inercial F_f fixo no carrinho, pode-se escrever (V_c , V_o e ω escritos no referencial F_f , mas com valores em relação o solo):

$$\vec{V}_C = \vec{V}_O + \vec{\omega} \times (C - O)$$

$$\vec{V}_C = V_O \hat{i} + \left(V_O * \frac{dc}{de} * \tan \alpha \right) \hat{j}$$

Para escrever uma relação para ξ deve-se perceber que esta é uma distancia relativa ao ponto C e, portanto, o conceito de velocidade relativa (referência [8]) deve ser utilizado, resultando em:

$$\dot{\xi} = \dot{\xi}_{vc} + \dot{\xi}_{vo}$$

$$\dot{\xi}_{vc} = V_O * \frac{dc}{de} * \tan \alpha$$

$$\dot{\xi}_{vo} = V_O * \tan \psi$$

$\dot{\xi}_{vc}$, relativa, é simplesmente a projeção da velocidade do ponto C na direção j

$\dot{\xi}_{vo}$, arrasto, pode ser obtido com a relação trigonométrica $\tan \psi = \frac{\dot{\xi}_{vo}}{V_O}$

Somando e substituindo $u = \omega = \frac{V_O}{de} * \tan \alpha$, fica-se com:

$$\dot{\xi} = dc * u + V_O * \tan \psi$$

Linearizando para pequenos ângulos ψ , fica-se com:

$$\dot{\xi} = dc * u + V_O * \psi$$

Derivando novamente e considerando ($\dot{\psi} = \omega = u$, válido caso não haja escorregamento das rodas):

$$\ddot{\xi} = dc * \dot{u} + V_O * u$$

Aplicando a transformada de Laplace e isolando os termos ficamos, finalmente, com:

$$\boxed{\frac{\xi(s)}{U(s)} = \frac{V_O + dc * s}{s^2}}$$

Como foi dito anteriormente, pode-se ver que o modelo encontrado possui um polo duplo em zero. Isto significa que um controle proporcional deve ser o suficiente para levar o

erro para zero. Viu-se também, durante a dedução, que o modelo considera um ξ à frente do carrinho, o que também é bom, como explicado anteriormente.

Contudo, três limitações deste modelo puderam ser vistas durante a dedução.

1) O modelo faz uso da aproximação para pequenos ângulos de ψ

Esta limitação é uma preocupação válida já que o ângulo ψ não é limitado. Contudo, mesmo que esta aproximação cause um erro, o controlador projetado é capaz de corrigi-lo.

2) Sem escorregamento

O escorregamento é ruim não apenas devido às aproximações impostas no modelo, mas também devido à perda de eficiência e desempenho do carrinho. Para minimizar o escorregamento basta manter a velocidade dentro de um limite controlado. Isto pode ser feito através de um controle de velocidade e de um sistema capaz de selecionar a velocidade adequada para cada obstáculo.

$$3) \ u = \omega = \dot{\psi} = \frac{V_o}{de} * tg \alpha$$

A entrada deste modelo, considerando a limitação 2, é a velocidade angular do carrinho. O problema com esta definição é que o atuador de posição atua no ângulo α somente, e ω é função de α e de V_o .

Há duas alternativas neste momento. Poderia-se abordar um modelo multivariável, utilizando V_o como segunda entrada. Contudo, isto nos faria perder a simplicidade de análise e projeto de um sistema SISO. Assim sendo, a solução escolhida pelo grupo foi considerar V_o constante por trechos. Ou seja, no instante de cálculo da entrada, consideramos que V_o é o valor medido naquele instante e desacoplado de ξ .

A segunda alternativa evita o sistema multivariável e faz com que ω seja função de α somente (os demais dados passam a ser parâmetros do sistema).

Tendo $\omega(\alpha)$ pode-se calcular sua inversa, obtendo $\alpha(\omega)$. Com isso, dado uma entrada u (calculada pelo controlador), pode-se calcular $\alpha(u)$ e obter o ângulo que as rodas frontais devem fazer. A função $\alpha(u)$ pode ser vista abaixo:

$$\alpha = \arctan\left(\frac{de}{Vo} * u\right)$$

6.2 Modelo Cinemático dos Motores

O modelo utilizado para os motores foi retirado da Apostila de Laboratório de Controle 2011A da Escola Politécnica da Universidade de São Paulo, escrita pelos professores:

- Fuad Kassab Junior
- José Jaime da Cruz
- Julio Cesar Aya
- Ricardo Paulino Marques

No capítulo 4 da apostila referida, o modelo encontrado é o seguinte:

$$\frac{\Omega(s)}{U(s)} = \frac{K * n^2}{1 + s * T}$$

No carrinho seria interessante ter a função de transferência com a velocidade linear (V) ao invés da velocidade angular (Ω) e o *duty cycle* do PWM do motor ao invés da tensão U . Contudo estas grandezas possuem, aproximadamente, uma relação diretamente proporcional (o PWM controlando a tensão média do motor e $V = \omega * R$ sem considerar o escorregamento). Considerando-se:

$$\Omega = K_1 * V$$

$$U = K_2 * PWM$$

E uma constante $K_3 = K * \frac{K_2}{K_1}$, pode-se obter a seguinte função de transferência:

$$\frac{V(s)}{PWM(s)} = \frac{K_3 * n^2}{1 + s * T}$$

Nesta nova função temos K_3 que é uma constante que relaciona os parâmetros elétricos da Ponte-H, o raio da roda e outras constantes e também temos n^2 que representa a redução do sistema. Como os motores foram fornecidos pela Freescale sem suas especificações técnicas, a redução n^2 será englobada em uma variável $K_4 = K_3 * n^2$. Com isso obtém-se finalmente:

$$\frac{V(s)}{PWM(s)} = \frac{K_4}{1 + s * T}$$

É importante notar que este modelo serve apenas para o caso de motores com comportamento *linear*.

7. Software de Telemetria

O software de telemetria, apesar de não ser essencial para o funcionamento do carrinho, é essencial para o projeto. Isto porque é preciso obter os dados do carrinho com a menor interferência possível. Assim sendo, este software foi desenvolvido de maneira a obter os dados por comunicação sem fio.

A comunicação sem fio permite obter os dados com o carrinho andando na pista sem que se tenha que conectar um osciloscópio ou, pior, levá-lo para a bancada. Os dados podem ser colhidos e analisados “em tempo real”.

O protocolo de comunicação sem fio escolhido foi o *Bluetooth*. A razão para esta escolha foi a baixa susceptibilidade a ruídos, a facilidade dos notebooks já virem com módulos *Bluetooth* de fábrica e ao fato de já possuímos um módulo que faz a conversão RS232 => *Bluetooth*.

Como a saída do microcontrolador utiliza protocolo UART (Transmissão e Recepção Assíncrona Universal) uma pequena placa de circuito impresso foi feita para converter de UART para RS232. O diagrama abaixo (Figura 7-1) resume as conversões e conexões:

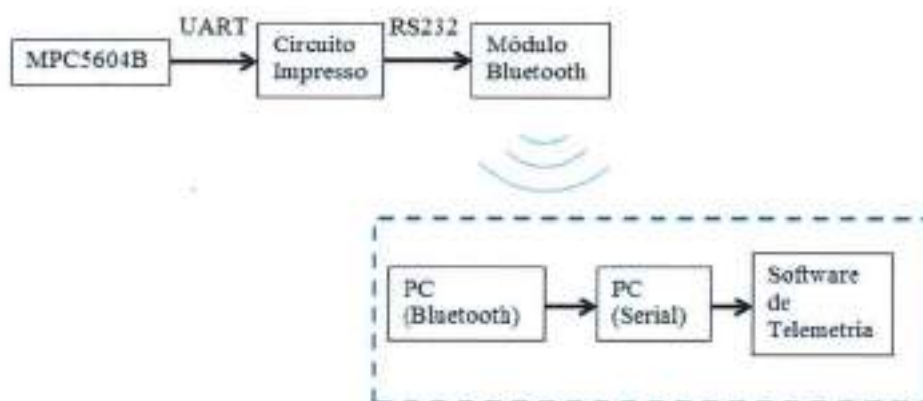


Figura 7-1. Conversões e protocolos de comunicação do projeto.

Antes da descrição do funcionamento do software de telemetria é importante explicar que ele exibe os dados obtidos de 3 maneiras distintas:

- 1) Através de um desenho esquemático
- 2) Através de gráficos
- 3) Através de um arquivo com padrão CSV

Os 3 modos serão detalhados mais a frente, mas antes do fluxograma ser explorado, é importante apontar que as maneiras (1) e (2) exibem os dados conforme os mesmos são obtidos. Já o método (3) gera um arquivo para análise posterior no Matlab, Excel ou outro programa compatível com o padrão CSV (*Comma Separated Values*), em que os valores de cada célula são separados pelo caractere ponto-vírgula.

É esperado que o microcontrolador envie os dados em sequência, assim cada linha do arquivo CSV representaria um ciclo do programa principal do microcontrolador.

Por exemplo, se os dados esperados são vef (Velocidade da roda Esquerda da Frente), vet (Velocidade da roda Esquerda de Trás) e vel (média das velocidades das quatro rodas), após o microcontrolador enviar o valor de α para o servomotor ele enviaria as seguintes mensagens:

Mensagem 1: "vef"
 Mensagem 2: "1502"
 Mensagem 3: "vet"
 Mensagem 4: "1498"
 Mensagem 5: "vel"
 Mensagem 6: "1500"

Sendo 1502, 1498 e 1500 os valores de vef, vet e vel respectivamente. Ao receber o valor de vel, o software de telemetria sabe que terminou um ciclo do laço infinito do programa principal do microcontrolador, portanto ele escreverá uma linha no arquivo CSV.

Um último detalhe sobre o arquivo CSV é com relação ao seu nome. Para auxiliar na organização dos arquivos, o arquivo gerado recebe o nome referente à data e à hora em que foi gerado. Por exemplo, o arquivo 21-5-2012_4-22-28.csv foi gerado no dia 21 de Maio de 2012 às 4h22min28s.

A Tabela 2 abaixo define as siglas interpretadas pelo Software e o que cada uma significa:

Tabela 2. Siglas do software de telemetria.

vet	Velocidade da roda Esquerda de Trás
vef	Velocidade da roda Esquerda da Frente
vdv	Velocidade da roda Direita da Frente
vdv	Velocidade da roda Direita de Trás
vel	Média das velocidades das 4 rodas
qsi	Valor do erro de posição atual
alp	Ângulo Alpha das rodas frontais

Como especificado nos objetivos e na motivação do projeto, o software gera gráficos e arquivos de qualquer sistema que siga suas regras de envio (uma *string* cadastrada seguida do valor da variável representada pela *string*). Ou seja, ele poderá não apenas ser utilizado para um carro em escala natural como também, com poucas adaptações (cadastramento de novas *strings*) ser utilizado para obter e salvar dados de qualquer sistema embarcado.

O software de telemetria possui um funcionamento simples que pode ser descrito pelo seguinte fluxograma:

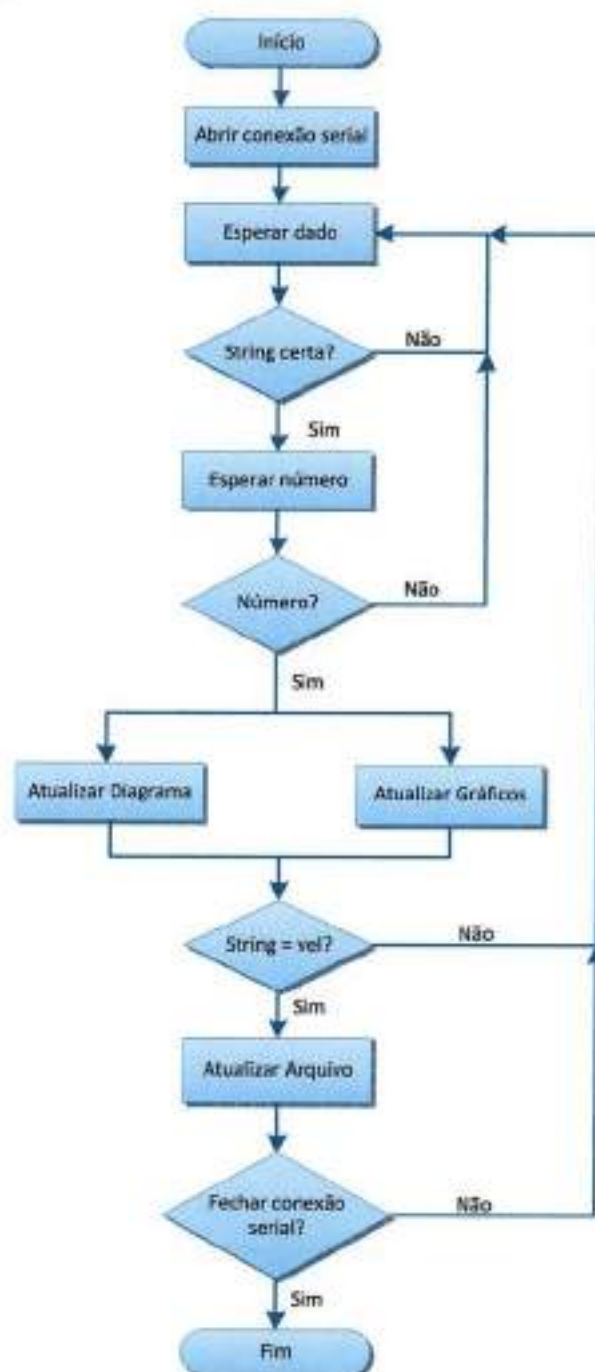


Figura 7-2. Fluxograma do Software de Telemetria

O fluxograma é de fácil entendimento. O programa abre uma conexão serial e aguarda o recebimento de dados. Ao receber os dados ele irá confirmar se a *string* recebida é alguma das que estão cadastradas (tabela acima). Caso não seja, ele irá retornar ao ponto de esperar uma nova *string*. Caso seja, irá seguir adiante e passará a esperar o valor daquela variável, digamos *vet* (Velocidade da roda Esquerda de Trás).

O esperado é que o próximo dado enviado seja o valor dessa variável. Caso não seja (por falha na transmissão ou erro de programação do microcontrolador) o software irá anular aquela leitura e voltará a esperar uma nova *string* cadastrada. Voltar ao estado de espera de *string* cadastrada permite que haja uma resincronização de envios (*string* seguida de valor do dado correspondente à *string*) entre o software de telemetria e o microcontrolador.

Caso o valor recebido seja um valor numérico, ele irá atualizar o diagrama esquemático (será mostrado mais abaixo) e os gráficos (também serão mostrados mais abaixo).

Por fim, o arquivo será atualizado apenas após o último dado ser enviado pelo microcontrolador, no caso *vel*. Se o usuário decidiu por fechar a conexão o programa fechará a conexão e se encerrará após a atualização do arquivo.

Janela Principal



Figura 7-3. Software de Telemetria (conectando o software).

Ao abrir o software a janela acima (Figura 7-3) aparecerá. No canto superior esquerdo há um menu *drop-down* para escolher a porta serial que está ligada ao *Bluetooth*.

Ao selecionar a porta uma nova conexão será estabelecida e o programa passará a receber os dados e atualizar o esquemático (desenho do lado esquerdo da Figura 7-4) e o arquivo. O botão “Fechar Porta” passará a ficar disponível e os valores passarão a ser modificados continuamente.

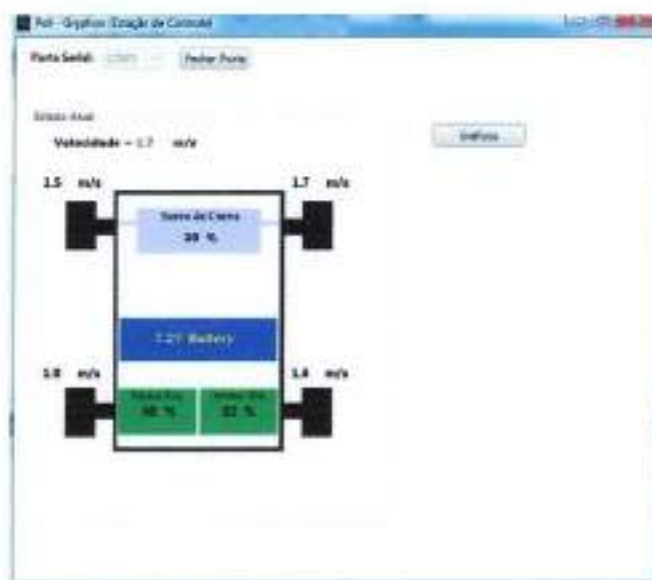


Figura 7-4. Software de Telemetria (Conectado)

Janela de Gráficos

Ao clicar no botão “Gráficos”, o software abrirá uma segunda janela (Figura 7-5):

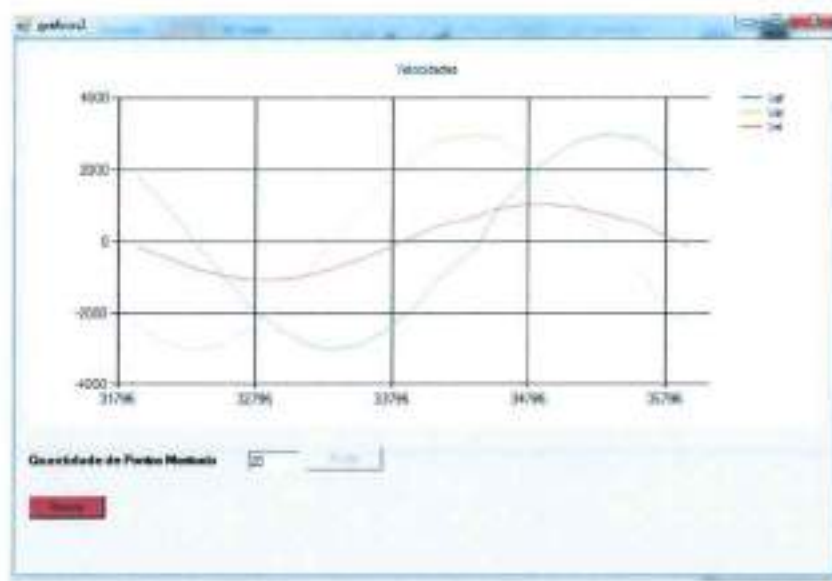


Figura 7-5. Software de Telemetria (Gráficos).

Nesta janela tem-se o traçado dos gráficos das variáveis (online). É possível aumentar ou diminuir o número de pontos mostrados no gráfico (20 nesta foto). Também é possível Pausar e Continuar os gráficos. O botão "Pausar" (vermelho) se altera para "Continuar" (verde) caso o gráfico esteja pausado (Figura 7-6 abaixo).



Figura 7-6. Software de Telemetria (Gráficos) - Botão Pausar/Continuar.

Arquivo

Após fechar a conexão o arquivo CSV é salvo. Seu formato pode ser averiguado através da Figura 7-7:



Figura 7-7. Arquivo CSV aberto no Excel e no Bloco de Notas.

8. Sensores, Atuadores e Medida de Qualidade

8.1 Atuadores

8.1.2 Servo Motor

O servo motor é um dos principais atuadores do projeto. Os módulos vendidos no mercado já vêm com controladores PID embutidos para controlar sua posição sendo, portanto, muito fácil a sua utilização e adaptação ao projeto. O módulo possui três entradas: duas de alimentação (Vcc e GND) e uma de sinal. Esta última irá controlar a posição na qual o servo motor deve se manter. A entrada de sinal recebe um sinal PWM sendo que seu *duty cycle* carrega a informação sobre a posição em que o servo deve se manter.

No caso específico do Futaba S3010, o período do PWM deve ser de 20ms e o *duty cycle* pode variar entre 5% e 10%. Isto significa que, ao mandarmos um PWM com *duty cycle* de 5%, o servo irá se movimentar o máximo possível no sentido anti-horário e, se mandar um PWM de 10%, ele se movimentará o máximo possível no sentido horário. Valores intermediários terão posições intermediárias. As posições máximas (no sentido horário e sentido anti-horário) são delimitadas fisicamente, o que significa que *duty cycles* acima de 10% ou abaixo de 5% irão mover o servo para as respectivas posições máximas.

É interessante perceber que a margem máxima de *duty cycle* está limitada em 10% para que o sistema interno do servo tenha tempo suficiente de estabilizar a posição enquanto o sinal está baixo. Desta maneira, quando o próximo comando for enviado (após um período T do primeiro comando) o sistema já estará estabilizado.

É possível reduzir o período e aumentar a margem do *duty cycle*, mas o fabricante não garante o posicionamento adequado do servo. Assim sendo, preferiu-se manter o período de 20ms sugerido pelo fabricante.

8.1.3 Motores

Os motores são controlados por Pontes-H. Estes circuitos permitem que a alimentação dos motores e dos demais circuitos eletrônicos seja feita de maneira independente. Não só isso como eles também permitem que a potência de cada motor seja controlada por um sinal PWM. Isto porque o sinal PWM mantém o motor energizado durante a parte do período em que ele está alto, o que permite controlar a tensão média oferecida ao motor, controlando assim sua velocidade.

Dessa forma, pode-se ver que o sinal PWM será o principal sinal de controle dos atuadores do projeto.

8.2 Sensores

8.2.1 Câmera

A câmera escolhida, como foi evidenciado pela matriz AHP do Anexo II, foi a câmera linear. Isto porque sua única desvantagem quando comparada à câmera matricial era o fato de que a câmera matricial possui uma imagem em duas dimensões. Ou seja, com ela era possível obter mais de um ponto de referência enquanto com a câmera linear apenas um ponto pode ser obtido.

Contudo esta única desvantagem pode ser compensada através do armazenamento de ponto como será explicado no capítulo 15. Assim sendo a câmera linear da Parallax com 128 pixels de resolução foi escolhida como o sensor de posição deste projeto.

8.2.2 Sensores Hall

Os sensores Hall são sensores capazes de detectar campos magnéticos. Neste projeto eles foram utilizados para medir a velocidade de rotação de cada roda de modo que, sabendo o raio da roda, é possível obter sua velocidade.

Os sensores Hall escolhidos foram os UGN3132 pois, mesmo a uma distância de 1cm da parte externa da roda, possuem sensibilidade suficiente para detectar o campo gerado pelos ímãs de neodímio que se encontram no interior das rodas.

Além disso, estes sensores possuem saída digital que modifica seu estado de acordo com a direção do campo detectado. Caso o lado Sul do ímã esteja de frente para o sensor, sua saída se modificará para baixa (0 lógico). O sensor manterá sua saída baixa até que o lado norte do ímã seja detectado fazendo com que sua saída fique alta (1 lógico). Este comportamento é importante, pois evita os ruídos de aproximação e afastamento dos polos do ímã. Por este mesmo motivo os 4 ímãs, internos às rodas, estão colocados com seus polos direcionados para o lado externo da roda se alternando.

Desta forma, na saída do sensor Hall será formada uma onda retangular onde cada patamar (alto ou baixo) representa o tempo entre a passagem de um ímã e de outro. Como os 4 ímãs estão igualmente espaçados o tempo entre 2 passagens de ímãs pelo sensor (duração de cada patamar) é o tempo no qual a roda percorreu um quarto de sua

circunferência. Assim, a velocidade de uma roda do protótipo pode ser encontrada pela seguinte função:

$$Vel_{roda} = \frac{1/4 * 2 * \pi * Raio_{roda}}{tempo_{patamar}}$$

8.2.3 Giroscópio

Como já falado no capítulo 5, o giroscópio é um sensor inercial capaz de medir a velocidade angular à qual está submetido. Em geral sua saída é analógica, necessitando que uma conversão A/D seja realizada pelo microcontrolador.

Este sensor é importante para o projeto, pois o sistema de controle de estabilidade utiliza esta medida para checar a estabilidade do carrinho com base na velocidade angular desejada. Esta, por sua vez, é obtida do controle posição, já que é a saída do controle (entrada da planta) como visto no modelo proposto no capítulo 6.

8.3 Medida de Qualidade do Projeto

Com o controle de velocidade e de posição prontos e testados, o carrinho já era capaz de seguir uma trajetória. As etapas seguintes fazem referência ao processo de otimização do sistema e, para isto, era necessária uma medida de qualidade para que se pudesse medir a melhoria adquirida com cada modificação.

Sabe-se que ao percorrer uma pista o carrinho varia sua velocidade de acordo com o obstáculo a sua frente (uma curva mais fechada exigirá uma velocidade menor que em uma curva mais aberta). Sabe-se também que a mínima velocidade do carrinho ao percorrer esta pista será na curva mais fechada dela.

Assim sendo, ao fazer com que o carrinho percorra a pista com uma velocidade V , constante, podemos ir aumentando o valor de V de maneira a encontrar a máxima velocidade constante (V_{max}) com a qual o carrinho consegue percorrer a pista sem erros. Esta velocidade será a velocidade máxima que o carrinho terá de ter nas curvas mais fechadas, mas nas curvas mais abertas ou nas retas, sua velocidade poderá ser maior. Com isso, a velocidade V_{max} encontrada será a velocidade mínima que o carrinho terá quando percorrer a trajetória completa variando sua velocidade.

Quanto maior for a velocidade mínima do percurso, maior será sua velocidade média e, portanto, menor será o tempo para percorrer qualquer formato de pista. Por este motivo

a velocidade $V_{\max}$ (constante) com a qual o carrinho consegue percorrer uma pista sem erros foi escolhida como medida de qualidade.

9. Obtenção de dados para Calibração de Sensores e Atuadores

9.1 Atuadores

9.1.1 Motor

Para o sistema de controle de velocidade é importante que se tenha uma curva indicando a velocidade de regime do carrinho para cada potência fornecida aos motores. Como já visto anteriormente, a ponte-H fornece aos motores uma potência proporcional ao sinal PWM que ela recebe.

Assim sendo, para o levantamento das velocidades de regime, foram enviados 5 diferentes *duty cycles* de um PWM com período de 5ms. Para cada um dos *duty cycles* escolhidos foi esperado que a velocidade do carrinho atingisse um patamar aproximadamente constante e a velocidade deste patamar foi considerada a velocidade de regime para aquela dada potência.

Os dados obtidos podem ser visualizados no item 2 do Anexo III. É interessante perceber que para o *duty cycle* de 20% a variação encontrada na medida é muito grande. Isto porque esta é a potência necessária para que o carrinho entre em movimento, ou seja, pequenas variações no atrito dos pneus com a pista já faziam o carrinho andar ou parar.

A Tabela 3 abaixo mostra os pontos obtidos durante os testes:

Tabela 3. Velocidades para seus respectivos PWMs.	
<i>Duty cycle (%)</i>	Velocidade Máxima de Reta
20%	0,1 m/s
40%	1,6 m/s
60%	2,4 m/s
80%	3,0 m/s
100%	3,5 m/s

9.1.2 Servo

Para o sistema de controle de posição é importante saber qual *duty cycle* deve ser aplicado ao servo para que as rodas frontais tenham o deslocamento angular desejado.

Segundo o fabricante sabe-se que o período deste *duty cycle* deve ser de 20ms e que o comportamento do servo é linear. Também se pode identificar a angulação mínima e máxima que o servo pode realizar através do fabricante, contudo é importante perceber que

dependendo de onde o servo está localizado, é possível que existam limitações mecânicas ao movimento das rodas (por causa do chassi, por exemplo). Assim sendo, é importante verificar a amplitude do movimento experimentalmente.

Para isto, primeiramente foi aplicado um *duty cycle* de 100% ao servo (ângulo das rodas totalmente para a direita) o qual foi reduzido gradualmente até se achar o *duty cycle* máximo (onde o servo não estava sujeito à forças contrárias ao seu movimento). No ponto em que o servo executar um leve movimento para a esquerda é porque não há mais forças contrárias ao seu movimento, sendo este, portanto seu ponto de máximo *duty cycle*. Este ponto também é o máximo que as rodas do carrinho poderão virar para a direita, definindo, portanto, o raio mínimo para a direita que o carrinho consegue executar.

Para achar o ponto de *duty cycle* máximo basta repetir o procedimento, mas para *duty cycle* iniciando de 0% e aumentando. Analogamente ao ponto de máximo, o ponto de mínimo é onde a roda faz seu ângulo máximo para a esquerda e, portanto, definirá o raio mínimo de curva para a esquerda que o carrinho conseguirá executar.

Como a saída do controle de posição será o ângulo das rodas frontais, o que precisamos encontrar neste ponto é a relação entre α e o *duty cycle* aplicado. Esta relação pode ser encontrada indiretamente com o raio mínimo dos dois pontos. Esta medida é muito mais precisa do que a medida com um transferidor e para realizá-la basta prender uma caneta ou lápis no centro do eixo traseiro do carrinho e medir o raio da circunferência desejada. Os resultados obtidos podem ser vistos na Tabela 4 abaixo:

Tabela 4. Relação de PWM com o raio de curva e o ângulo do servo.

	PWM (%)	Raio (cm)	Ângulo α (°)
Mínimo	5,75%	37,75	-27,91
Máximo	9,02%	39	27,14

Conforme dados do fabricante, o funcionamento do servo é aproximadamente linear, assim, pode-se realizar uma regressão linear, obtendo a seguinte função *Duty Cycle*(α):

$$DutyCycle(\alpha) = 11,88 * \alpha + 1481,57$$

Um ponto a ser observado é que a função acima devolve a quantidade de microssegundos (do período de 20000 microssegundos) que o sinal deve ficar alto. Foi feito deste jeito para que a programação ficasse mais simples já que o microcontrolador utiliza o tempo, em microssegundos, para definir o *duty cycle* do PWM e não uma porcentagem. Da maneira como foi feito não é necessário perder tempo de processamento calculando o tempo que o sinal deve ficar alto a partir da porcentagem.

Por exemplo, com a função acima é possível obter o ponto onde as rodas ficam centralizadas ($\alpha = 0$) que é 1481,57. Em porcentagem, seria $1481,57/20000 = 7,4\%$.

9.2 Sensores

9.2.1 Câmera

A calibração da câmera deve ser realizada utilizando o auxílio de um osciloscópio. Isto porque é importante ver seu sinal de saída para atingir o ponto ideal de funcionamento.

O primeiro ajuste a ser feito é seu foco. Algumas câmeras possuem ajuste automático de foco, como não é o caso da câmera escolhida este ajuste é necessário. O foco depende da distância do ponto de visada da câmera até sua lente, ou seja, cada vez que esta distância for modificada este ajuste deve ser refeito.

Com relação à câmera escolhida (de linha), quanto mais retangular for seu sinal (quando estiver visualizando a pista) melhor estará calibrado seu foco. A Figura 9-1 abaixo, retirada de [12] ilustra um sinal mal focalizado (lado esquerdo) e bem focalizado (lado direito).

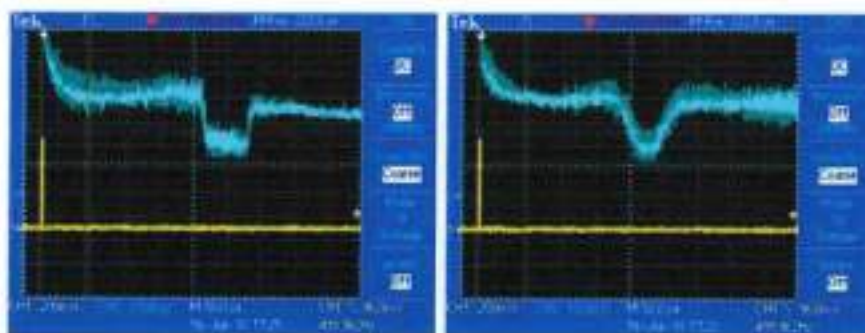


Figura 9-1 - Exemplo de bom (esquerda) e mal (direita) ajuste do foco da câmera

O segundo ajuste deve ser feito pelo software, mas ainda utilizando o osciloscópio. Para o sistema de controle de posição é importante que seja feita a transformação de pixel

para alguma unidade de comprimento como metros, milímetros ou décimos de milímetros. Para isto é necessário saber quanto cada pixel mede. Isto pode ser feito utilizando a pista e sabendo o comprimento L da faixa preta e o período do sinal de *clock* enviado para a câmera, no caso deste projeto: $4\mu\text{s}$.

Assim, para enviar os 128 pixels a câmera leva $512\mu\text{s}$. Medindo, com auxílio do osciloscópio, o comprimento do pulso negativo (faixa preta) e dividindo este valor por 4 obtém-se o número de pixel que equivale a faixa preta. Dividindo L por este valor obtém-se o comprimento de cada pixel.

9.2.2 Giroscópio

Para que seja possível utilizar um giroscópio, é necessário saber a relação entre a sua tensão de saída e a velocidade angular. Essa informação costuma ser fornecida no "*datasheet*" com unidade de milivolts por graus por segundo. Apesar de dado, muitas vezes este valor pode sofrer alterações, sendo prudente a realização de um teste de calibração antes da sua utilização.

Dos sensores e atuadores utilizados, o giroscópio é aquele que exige mais cuidado na calibração. Isto porque ele definirá o erro do sistema de estabilidade, ou seja, caso a calibração não seja adequada o sistema passará a desestabilizar o carrinho ao invés de estabilizá-lo. Além disso, ele também será utilizado na correção de pontos armazenados (como será visto no capítulo 15) e se estiver mal calibrado poderá também fazer com que o sistema de posição não funcione adequadamente.

Para calibrar um giroscópio é preciso colocá-lo em algum objeto que esteja rotacionando com velocidade conhecida, e então medir sua tensão de saída para obter pontos da curva de tensão de saída do giroscópio pela velocidade angular à qual ele está submetido.

Além disto, foi necessário um equipamento que mantivesse uma velocidade constante e em baixas rotações já que o mesmo media no máximo $300^\circ/\text{s}$. Por questões práticas, o instrumento utilizado para isto foi uma vitrola com opção de 33RPM e 45RPM e com possibilidade de ajuste em cada velocidade, o que permitiu a obtenção de mais de 2 pontos e, portanto, uma estimativa mais precisa da curva.

Foi encaixado no pino central da vitrola, um isopor para a sustentação da placa do giroscópio, o qual foi alimentado por um Arduino com a tensão de 3,3V. Ainda na plataforma

da vitrola, foi colocado o multímetro para a medida da tensão de saída, e um adesivo para a contagem do período de cada rotação. A Figura 9-2 abaixo demonstra a montagem para o levantamento da curva:

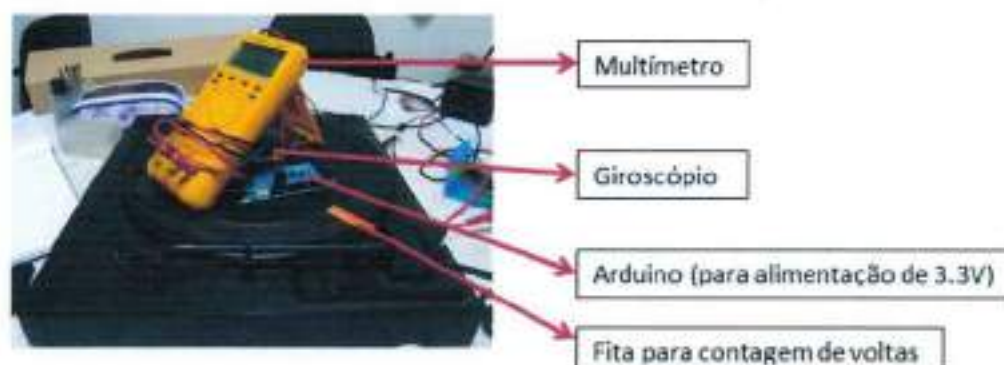


Figura 9-2 - Sistema para Calibração do Giroscópio

Para medir a frequência de rotação da vitrola, foi utilizada uma fita adesiva como marcação de início e fim. Para reduzir o erro devido a imprecisões de medidas (pela largura da fita e precisão de início e fim da marcação do tempo) foram contadas 20 voltas para cada teste e o tempo resultante foi dividido por 20.

Medido o tempo de cada volta, a velocidade angular era obtida indiretamente através da seguinte conta:

$$\omega = \frac{360^\circ}{T_{\text{volta}}}$$

Com isto pôde-se obter a velocidade angular da vitrola e, com o multímetro, o valor de saída do giroscópio. Feito isto, alterava-se o ajuste de 33RPM da vitrola para obter outros pontos e, em seguida, passava-se para os ajustes de 45RPM.

Para cada velocidade padrão (33RPM e 45 RPM) foram feitos quatro ajustes de velocidade. Para cada ajuste foram realizadas três medidas em que a média das três será considerada como o valor daquele ponto da curva de calibração. Os valores encontrados podem ser vistos nas Tabelas 5 a 8 a seguir:

Tabela 5 - Valores da Calibração do Giroscópio para rotação de 273,421°/s

	Tensão de saída (V)	Rotação (°/s)
	0,331	274,809
	0,332	272,727
	0,333	272,727
Média	0,332	273,421

Tabela 6 - Valores da Calibração do Giroscópio para rotação de 203,008°/s

	Tensão de saída (V)	Rotação (°/s)
	0,566	202,247
	0,565	203,389
	0,564	203,389
Média	0,565	203,008

Tabela 7 - Valores da Calibração do Giroscópio para rotação de 183,052°/s

	Tensão de saída (V)	Rotação (°/s)
	0,634	182,741
	0,632	183,673
	0,633	182,741
Média	0,633	183,052

Tabela 8 - Valores da Calibração do Giroscópio para rotação de 175,897°/s

	Tensão de saída (V)	Rotação (°/s)
	0,655	175,610
	0,654	176,471
	0,655	175,610
Média	0,655	175,897

Como a vitrola utilizada não tinha a opção de rodar no sentido inverso, após obter estes quatro valores, o giroscópio foi invertido (colocado de cabeça para baixo) e mais quatro medições foram feitas.

Os valores encontrados podem ser vistos nas Tabelas 9 a 12 a seguir:

Tabela 9 - Valores da Calibração do Giroscópio para rotação de -243,795°/s

	Tensão de saída (V)	Rotação (°/s)
	2,033	-243,243
	2,034	-244,898
	2,028	-243,243
Média	2,032	-243,795

Tabela 10 - Valores da Calibração do Giroscópio para rotação de -214,499°/s

	Tensão de saída (V)	Rotação (°/s)
	1,941	-214,925
	1,940	-214,286
	1,938	-214,286
Média	1,940	-214,499

Tabela 11 - Valores da Calibração do Giroscópio para rotação de -185,249°/s

	Tensão de saída (V)	Rotação (°/s)
	1,845	-185,090
	1,844	-185,090
	1,844	-185,567
Média	1,844	-185,249

Tabela 12 - Valores da Calibração do Giroscópio para rotação de -176,183°/s

	Tensão de saída (V)	Rotação (°/s)
	1,810	-175,610
	1,809	-176,470
	1,807	-176,470
Média	1,809	-176,183

Obtidos estes valores, a seguinte tabela (Tabela 13) de pontos para a calibração pôde ser obtida:

Tabela 13 - Pontos para Calibração do Giroscópio

Rotação (°/s)	Tensão (V)
-243,795	2,032
-214,499	1,94
-185,249	1,844
-176,183	1,809
0	1,238
175,897	0,655
183,052	0,633
203,008	0,565
273,421	0,332

Utilizando o Matlab para interpolar os valores da tabela acima o seguinte polinômio pode ser obtido:

$$Tensão = -0.0033 * Rotação + 1.2333$$

O gráfico abaixo demonstra o resultado da interpolação:

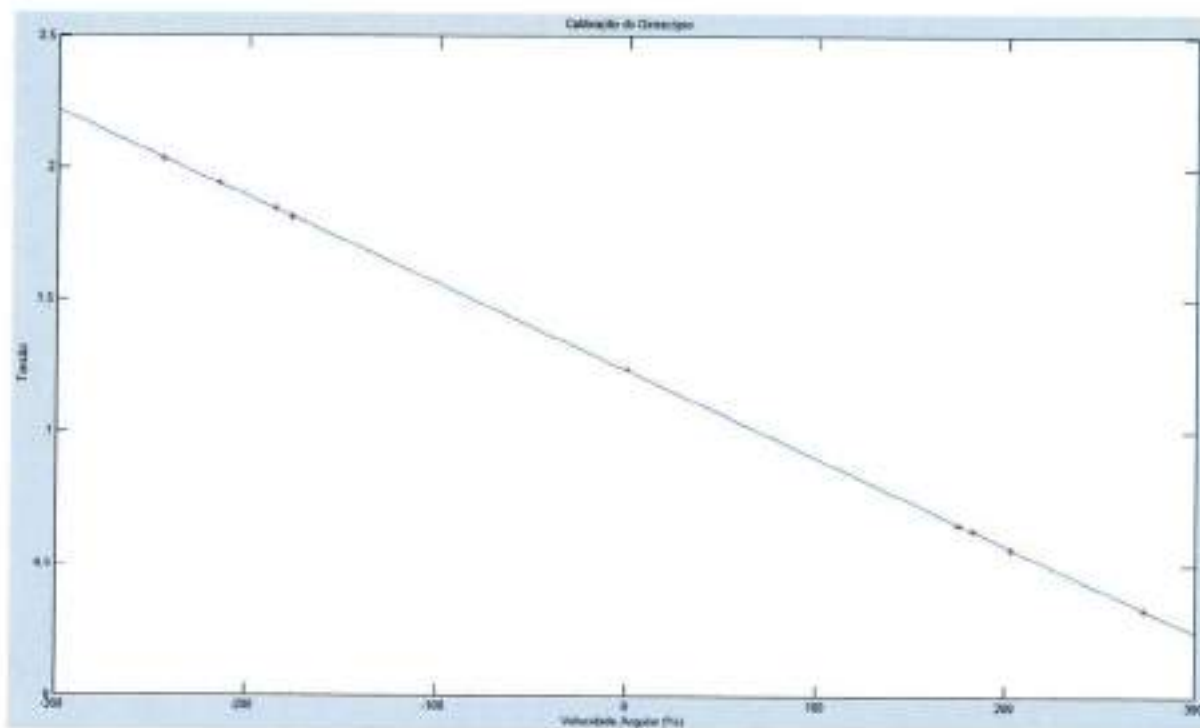


Figura 9-3 - Resultado da Calibração do Giroscópio

Com estes pontos seria possível levantar uma curva entre a tensão de saída do giroscópio e a velocidade angular. Contudo, como o sinal de saída é analógico e deverá ser interpretado pelo conversor AD do microcontrolador, a equipe preferiu levantar uma curva entre o valor resultante da conversão (0-1023) e a velocidade angular. É importante notar que a alimentação do giroscópio é de 3,3V e a tensão de referência do microcontrolador fica em torno de 5V, assim sendo os valores mais altos da conversão (1023, por exemplo) não serão atingidos. Fazendo a interpolação dos pontos obtidos com a ajuda do Matlab, a seguinte função foi encontrada:

$$\omega_{medido} = 374,244 - 1,460 * ADC$$

Sendo ω_{medido} o valor obtido para a velocidade angular através do giroscópio e ADC o valor resultante da conversão AD da saída do giroscópio. É importante perceber que esta curva pode variar dependendo da tensão de referência do microcontrolador (que depende do tipo de bateria utilizada e do circuito de potência) e também do giroscópio utilizado.

10. Obtenção e Filtragem da Imagem

Apesar de não ser um sistema de controle, a obtenção e filtragem da imagem são de vital importância para o projeto. Isto porque seu sistema de posicionamento, que faz com que o carrinho saiba para onde se deslocar, obtém o erro a partir da leitura da câmera. Uma leitura que não seja confiável fará com que o carrinho se perca, ou seja, não saiba para onde deve virar suas rodas.

Tanto a obtenção da imagem como a filtragem da mesma, influenciam diretamente o desempenho do carrinho. Como as imagens não confiáveis são descartadas, quanto maior a taxa de imagens confiáveis menor será o tempo médio que o carrinho ficará com uma imagem desatualizada. Por exemplo, se para cada 4 imagens obtidas apenas 1 for confiável o carrinho andarà o tempo de 3 imagens sem atualizar sua leitura de posição. Já se 1 imagem for confiável a cada 2 obtidas, o carrinho andarà o tempo de apenas 1 foto sem atualizar sua leitura de posição.

10.1 Obtenção da Imagem

10.1.1 Por Largura de Pulso (IPWM)

Um dos módulos do microcontrolador escolhido (Freescale MPC5604B) é o eMIOS. Este módulo possui diversas funcionalidades, todas ligadas aos contadores em geral. Entre estas funcionalidades está a capacidade de se medir a largura de um pulso ou IPWM (Medida da Largura de um Pulso de Entrada). Uma das vantagens deste módulo é o fato dele possuir DMA (Acesso Direto a Memória), ou seja, ele é capaz de alterar o valor de um registrador mesmo que o programa esteja executando outra operação.

Outro ponto interessante a ser notado é que nosso projeto, a princípio, necessita apenas da posição da linha preta em relação a um ponto C (Capítulo 6). O sinal da câmera ao observar a pista (branca com uma linha preta) se resume a um único pulso negativo com a largura da faixa preta. Então, ignorando os ruídos (serão tratados mais a frente), basta encontrar o ponto inicial desta faixa preta em relação à câmera, o que a funcionalidade IPWM do módulo eMIOS do microcontrolador faz sem ocupar tempo de processamento. A Figura 10-1 abaixo exemplifica a situação:



Figura 10-1 - Sinal da Câmera em Relação à posição do Carrinho

Na figura, vê-se que a seta verde indica o erro em relação à pista. O IPWM do microcontrolador utiliza dois registradores: registradorA e registradorB. No registradorB ele armazena o valor de um contador no início do pulso e no registradorA ele armazena o valor do contador no fim do pulso. Assim sabemos qual foi a duração do pulso (diferença entre o registradorA e o registradorB) e quando o mesmo começou (registradorB). Apesar de estes valores serem dados em tempo, podemos calcular em espaço (já que estes pulsos significam a linha preta, física, no chão) como foi visto no capítulo 9.

A cada nova medida adquirida (há um *flag* no microprocessador que indica caso a medida seja nova) basta ler o valor contido no registradorB para obter o erro atual (que será corrigido pelo controle de posição explicado anteriormente).

Esta ideia funciona bem quando temos um único pulso, claro e limpo como demonstrado na figura anterior. Infelizmente, na maioria dos casos, o sinal encontrado é mais parecido com o da Figura 10-2 abaixo:



Figura 10-2 - Exemplo de sinal de saída da câmera

Para que o pulso seja limpo (sem as oscilações de alta frequência) um comparador pode ser utilizado. A ideia do comparador é fazer com que apenas quando o sinal está abaixo de uma tensão pré-definida é que o pulso apareça. Como estas oscilações dependem

da luz do ambiente, é necessário que haja alguma possibilidade de configuração no local. Assim sendo a outra entrada do comparador analógico deve ser conectada a um potenciômetro e o mesmo deve ser configurado no ambiente de teste. O resultado pode ser visto na Figura 10-3:



Figura 10-3 - Filtragem da imagem por Hardware (comparador)

Esta primeira técnica de obtenção da imagem é mais simples e, por isso, foi utilizada nos testes iniciais. Ela também ocupa menor tempo de processamento já que sua filtragem é feita por hardware.

Contudo, apesar de ser muito simples ela apresenta um problema sério. Mesmo utilizando o comprimento do pulso como parâmetro de qualidade da imagem (se for menor que 2,7cm ou maior que 3,3cm a medida é ignorada) o erro de medida era algo muito comum, acontecendo em uma a cada três imagens medidas. Isto porque qualquer ruído mais elevado do que a tensão prevista no potenciômetro irá ocasionar um pulso que será dado como medida. A Figura 10-4 mostra um exemplo onde ruídos podem causar problemas na leitura do sinal utilizando o módulo IPWM:



Figura 10-4 - Filtragem da imagem por Hardware (comparador) com Ruído

Nela, vê-se que um pequeno ruído ocasionou dois pulsos na saída do comparador. O módulo IPWM irá fornecer a medida do último pulso lido, podendo ser o pulso da pista (terceiro da Figura 10-4 acima) ou um de ruído (os dois primeiros da Figura 10-4). Como explicado acima, caso um dos dois primeiros pulsos fosse o pulso armazenado esta imagem seria desconsiderada, pois nenhum dos dois possui uma largura próxima à da pista (se possuísse o carrinho ainda estaria sujeito a um erro de medida ao invés de uma imagem desperdiçada).

Para atenuar estes ruídos o tempo de integração da câmera tem de ser elevado, mas mesmo com 10ms de tempo de integração este erro ainda é considerável (apenas 66% das imagens eram aproveitáveis).

10.1.2 Por Conversão A/D

A conversão A/D apresenta uma série de vantagens em relação à técnica anterior, as quais serão explicadas a seguir. Sua desvantagem, além da maior complexidade de código, é que ela exige um tempo maior de processamento. Mas este tempo maior de processamento acaba compensando no resultado final.

Como a imagem da câmera escolhida possui 128 pixels, são necessárias 128 conversões A/D. Dado que a conversão A/D leva menos de $4\mu\text{s}$ (tempo de transmissão de um pixel) para ocorrer, o tempo que se leva para obter a imagem será definido pelo tempo que a câmera leva para transmitir a imagem e não pelo tempo de cada conversão. No caso de projetos futuros em que o tempo de transmissão de cada pixel for menor que o tempo necessário para cada conversão, será importante levar este tempo em consideração.

Para sincronizar a conversão A/D com a transmissão da imagem o mesmo sinal de *clock* utilizado para câmera será utilizado para a conversão. Como cada pixel é enviado a cada sinal de *clock*, então o sinal de *clock* servirá de gatilho para se iniciar cada conversão. Considerando que cada pixel leva $4\mu\text{s}$ para ser transmitido e que a câmera possui 128 pixels, o tempo total de conversão é de $512\mu\text{s}$.

Para avaliar este tempo é importante ter em mente o tempo de integração da câmera. Deseja-se que a câmera tenha uma taxa de aproveitamento das imagens obtidas de, no mínimo, 50%. Em outras palavras, 1 imagem boa para cada 2 imagens obtidas.

Como a linha preta possui 3cm e, no pior caso, ela estará perpendicular ao carrinho, deve-se garantir que o carrinho consiga lê-la a tempo. Como o carrinho tem que, no mínimo,

tirar uma foto boa a cada duas, deve-se garantir que enquanto ele percorre estes 3cm da fita ele deve tirar 2 fotos. Isto equivale a uma foto a cada 1,5cm.

Nos capítulos anteriores viu-se que a velocidade máxima do carrinho é de 3,5m/s. Se for desejado que o carrinho tire 1 foto a cada 1,5cm, cada foto deve ser tirada em $1,5/350$ (s) = $1500/350$ (ms) = 4,3ms. Ou seja, o tempo máximo entre as fotos é de aproximadamente 4ms, o que inviabiliza a primeira técnica. Como o tempo de obtenção da imagem é de 0,512ms, sobra aproximadamente 3,5ms (87,5% do tempo total) para filtrar esta imagem e realizar os controles de posição e velocidade. Portanto conclui-se que esta técnica é melhor para o projeto, pois apesar de demandar mais tempo do que a anterior este tempo é pequeno e aceitável, o que elimina a única desvantagem com relação à técnica anterior.

Utilizando a conversão A/D os ruídos passaram a ser facilmente percebidos, podendo ser eliminados através da detecção de borda na filtragem da imagem. A Figura 10-5 abaixo mostra a ordem em que cada passo é feito pelo microcontrolador:



Figura 10-5 - Rotina do Microcontrolador

10.2 Filtragem da Imagem

Após a obtenção da imagem, a mesma é armazenada em um vetor de 128 posições sendo que cada posição guarda o valor do respectivo pixel obtido pela câmera. Serão analisados o funcionamento de duas técnicas de filtragem para detecções de bordas nas imagens.

10.2.1 Detecção de Bordas

Um dos conceitos mais importantes quando se trabalha com detecção de objetos em uma imagem é a detecção de bordas. A detecção de bordas nada mais é do que encontrar pontos na figura onde há uma variação brusca de alguma medida como cor ou intensidade luminosa.

No caso deste projeto, utilizando uma câmera linear, há duas bordas na pista. A primeira é quando a faixa preta se inicia onde a intensidade luminosa que era alta passa a ser baixa. A segunda borda é no final da faixa preta, onde a intensidade luminosa que era baixa passa a ser alta. Ao se detectar estas duas bordas, vê-se que a faixa preta está bem definida na imagem já que se pode identificar seu começo e seu fim.

Neste projeto 2 filtros com a capacidade de detectar bordas serão estudados: um filtro diferencial isolado e o Algoritmo de Canny que utiliza o filtro gaussiano e o filtro diferencial em conjunto.

10.2.2 Filtro Diferencial Isolado

O filtro mais simples de ser utilizado para a detecção de bordas é o filtro diferencial de maneira isolada, ou seja, sem qualquer outro filtro sendo aplicado à imagem.

Isto porque é fácil entender seu funcionamento na detecção de bordas. Vamos considerar uma imagem com 10 pixels em linha como abaixo:

80	80	80	80	10	10	10	80	80	80
----	----	----	----	----	----	----	----	----	----

Aplicando uma diferença simples onde:

$$S_i = F_{i+1} - F_i; \forall i \in [0; 9[$$

$$S_9 = 0$$

sendo S_i a saída do filtro e F_i a imagem exemplificada acima com i pertencente ao intervalo entre 0 e 9, a seguinte imagem será a saída do filtro:

0	0	0	-70	0	0	70	0	0	0
---	---	---	-----	---	---	----	---	---	---

Vemos que o filtro funciona como esperado no exemplo proposto. Inclusive é possível perceber o pixel onde a faixa se inicia e onde a mesma termina já que os valores são pontos de mínimo e máximo da imagem. É possível ainda calcular o comprimento da faixa: o ponto de máximo se encontra no pixel de índice 6, enquanto o ponto de mínimo se encontra no pixel de índice 3. Assim, a faixa proposta possui a largura de 4 pixels, que podem ser transformados em uma medida de comprimento com a calibração proposta no capítulo 9.

Também se vê que o carrinho se encontra no centro da pista. O centro da faixa seria entre o pixel de índice 4 e o pixel de índice 5, pois:

$$CentroFaixa = \frac{(6 + 3)}{2} = 4,5$$

E o centro do carrinho será sempre o meio da imagem, pois com a câmera centralizada há o mesmo número de pixels para a direita e para a esquerda. Logo:

$$CentroCarrinho = \frac{(9 + 0)}{2} = 4,5$$

Analisando apenas por este exemplo o filtro parece funcionar bem. Mesmo considerando pequenos ruídos percebe-se que estes seriam atenuados pela diferença. Mas isto nem sempre acontece.

Um dos motivos é que muitas vezes o ambiente não é adequadamente iluminado, fazendo com que a diferença da intensidade luminosa da faixa preta e da parte branca não seja tão grande. Além disso, pequenos ruídos em torno da média podem se tornar maiores quando considerados apenas dois pontos. Considere o exemplo abaixo:

20	25	15	20	10	13	8	18	21	24
----	----	----	----	----	----	---	----	----	----

Vê-se que a parte branca tem uma intensidade média de 20 e a parte preta de 10. Ainda é possível ver que a faixa preta se encontra entre os pixels 3 e 6 novamente. Contudo, ao aplicarmos o filtro obtemos a seguinte imagem:

5	-10	5	-10	3	-5	10	3	3	0
---	-----	---	-----	---	----	----	---	---	---

Neste novo resultado percebe-se que não é mais possível saber onde a faixa preta se inicia. Mesmo considerando os mínimos globais ainda não é possível dizer se a faixa se inicia no pixel de índice 1 ou no pixel de índice 3. Ou seja, o resultado após a filtragem é pior do que a imagem original para a detecção de bordas.

A conclusão é que em uma primeira análise este filtro parece ser um eficiente detector de bordas, contudo utilizá-lo isoladamente não se mostra eficaz. Ruídos (que sempre existirão em protótipos e construções reais) podem fazer com que este filtro piore a imagem ao invés de melhorá-la.

10.2.3 Algoritmo de Canny

Como visto no capítulo 5 o algoritmo de Canny é a utilização de um filtro Gaussiano em conjunto com um filtro diferencial. O filtro gaussiano serve para atenuar os ruídos fazendo com que a imagem a ser filtrada pelo filtro diferencial se pareça mais com o primeiro exemplo do Filtro Diferencial Isolado e, portanto, tenha uma saída mais adequada para a detecção de bordas.

Ainda no capítulo 5 viu-se que é possível aplicar os dois filtros com uma única máscara, esta calculada pela seguinte equação:

$$G'(x) = \frac{-x}{\sigma^3 \sqrt{2\pi}} \cdot e^{-\frac{x^2}{2\sigma^2}}$$

O valor de σ vem da gaussiana escolhida para o filtro gaussiano. Quanto maior este valor mais aberta será a gaussiana e, portanto, maior será o número de pixels utilizados na média. O efeito prático do sigma é que quanto maior for seu valor mais diluído ficará o ruído. Contudo, também será maior o tamanho da máscara do filtro (o que aumenta o tempo de processamento durante a filtragem da imagem) e também mais diluída estará a borda.

Assim sendo, o valor ideal do σ será o menor valor para o qual os ruídos são atenuados o suficiente para restar apenas a borda na saída do filtro diferencial.

Calculando a máscara para $\sigma=0,5$ tem-se um total de $3*0,5=1,5$ pixels para cada lado. Aproximando para cima significa o pixel central mais 2 pixels para cada lado o que implica em uma máscara de dimensão 5. Utilizando a fórmula acima e normalizando para que haja apenas inteiros (para otimizar o microprocessador) fica-se com a seguinte máscara:

$$\sigma = 0,5$$

$$\frac{1}{467} * \begin{bmatrix} 1 & 202 & 0 & -202 & -1 \end{bmatrix}$$

Para $\sigma=1,0$ tem-se uma máscara com 7 posições ($3\sigma=3$):

$$\sigma = 1,0$$

$$\frac{1}{75} * \begin{bmatrix} 1 & 8 & 18 & 0 & -18 & -8 & -1 \end{bmatrix}$$

Para $\sigma=2,0$ tem-se uma máscara com 13 posições ($3\sigma=6$):

$$\sigma = 2,0$$

$$\frac{1}{301} * [1 \quad 3 \quad 8 \quad 15 \quad 18 \quad 13 \quad 0 \quad -13 \quad -18 \quad -15 \quad -8 \quad -3 \quad -1]$$

Como já explicado, quanto maior o tamanho da máscara maior será o tempo necessário para realizar a convolução. Utilizando a máscara com $\sigma=1,0$ foi visto, empiricamente, que as bordas ficam bem definidas e detectáveis aproximadamente 99% das vezes (média de um erro a cada cem imagens). Esta taxa de acerto já é muito superior do que a utilização exclusiva do filtro diferencial. A Figura 10-6 abaixo mostra um trecho da imagem (36 dos 128 pixels) em que a faixa preta de 3cm (aproximadamente 10 pixels) está contida:

Imagem Original	30	29	33	33	34	34	33	32	32	32	32	29	19	29	28	28	18	19	18	19	27	32	30	25	25	35	35	35	36	36	39	36	33	34	34	37
Gauss-Canny	0	0	0	0	0	0	0	0	0	-1	-3	-4	-2	-1	0	0	0	0	1	3	5	4	1	0	0	0	0	0	0	0	0	-1	0	3	0	0

Figura 10-6 Trecho de imagem obtida pela câmera linear.

Pela figura, é possível perceber que após a aplicação do filtro basta buscar os pontos de máximo e mínimo do vetor para encontrar as bordas da faixa preta.

O terceiro passo do algoritmo de Canny consiste na minimização dos valores abaixo de certo limite estabelecido para o projeto. Para escolha correta do limite, deve-se observar os valores encontrados após a segunda etapa. Por exemplo, se o valor 4 for escolhido como limite, a imagem acima terá todos os valores entre -4 e 4 igualados a zero, fazendo com que a borda esquerda da faixa preta (transição do branco para o preto e, portanto, um ponto de mínimo local) não seja detectada. Já se o valor 3 for escolhido como limite, a borda esquerda da faixa será detectada no décimo segundo pixel e a borda direita nos pixels no vigésimo primeiro e no vigésimo segundo pixel. No caso deste projeto a decisão sobre qual dos dois pixels será considerado como borda direita é feita analisando-se o instante em que o valor do pixel começa a diminuir (ponto de máximo). Ou seja, como o vigésimo segundo pixel (4) é menor que o vigésimo primeiro (5), o vigésimo primeiro será considerado como ponto de máximo local.

Este último passo do algoritmo de Canny é desnecessário caso se deseje que apenas as bordas sejam detectadas. Contudo, é importante lembrar que a pista pode conter ruídos

como reflexos da luz ambiente (causadas por sujeira, pela própria pista, fitas adesivas para unir as cartolinas, e outros fatores) que causem bordas com largura próxima de 3 centímetros. Estes ruídos seriam interpretados como pista pelo carrinho (pois formam bordas) a menos que se tenha algum meio de identificar a intensidade da borda. Uma borda do branco para o preto será consideravelmente mais intensa do que uma borda do branco para um reflexo ou alguma outra cor. Assim sendo este terceiro passo do algoritmo se mostrou de uma importância vital para a eficiência da detecção de imagens do projeto.

A escolha do limite deve ser feita através da análise da imagem e do resultado do filtro. Inicialmente o carrinho pode ser capaz de realizar uma calibração da câmera, buscando os valores mínimos e máximos da saída do filtro e utilizar um valor de, por exemplo, 90% dos valores encontrados como limite. Este método é eficiente quando a distribuição de luz é homogênea na pista, mas pode apresentar perda de desempenho quando a luz varia consideravelmente no decorrer do trajeto. Por exemplo, em uma sala comprida com iluminação indireta nas paredes, a parte central da pista será mais escura que suas laterais. A Figura 10-7 ilustra a situação do exemplo:

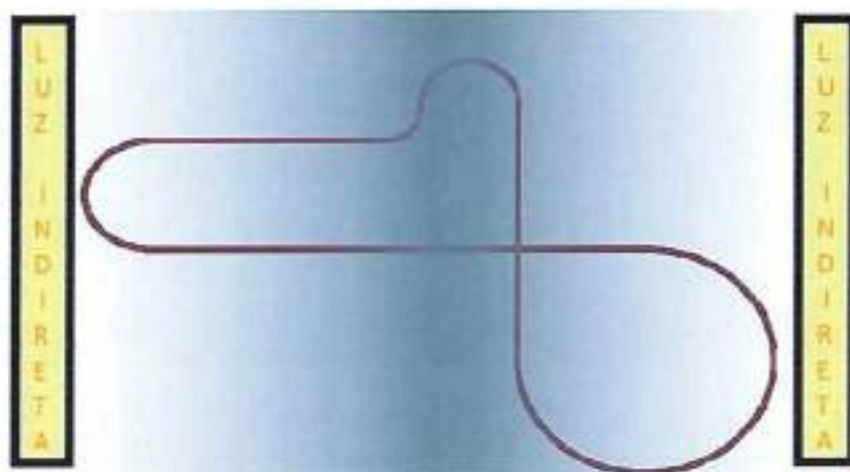


Figura 10-7 - Exemplo de Pista com variação na iluminação durante o trajeto

No caso desta figura, corre-se o risco do carrinho ser calibrado nos pontos claros da pista, o que gerará uma perda de eficiência na parte escura. Nestes casos é mais interessante realizar a calibração manualmente, vendo os valores de saída do filtro de Canny em pontos variados da pista e escolhendo um valor adequado (em geral os valores máximos e mínimos da parte mais escura).

Esta necessidade de calibração manual pode ser suprida pela adição de faróis/lanternas, pela utilização de uma câmera colorida, pelo ajuste de um maior tempo de integração entre uma foto e outra ou por um algoritmo adaptativo. A primeira ideia acarretaria em um aumento de peso e consumo da bateria, os três casos seguintes aumentariam o tempo de processamento necessário para se obter a imagem, consequentemente aumentando o tempo em que o carrinho fica sem uma atualização de sua posição. Como o objetivo deste projeto é elaborar e analisar sistemas de alto desempenho a configuração manual foi adotada, mas no caso de sistemas em escala natural essa técnica é inviável e recomenda-se a utilização de lanternas infravermelhas. Isto porque em um carro de tamanho natural as lanternas passam a ter um peso e um consumo muito pequeno perto do restante do carro, fazendo com que a perda de desempenho seja minimizada. Já a escolha pelo infravermelho se deve ao fato de que este tipo de luz não irá influenciar a visão de outras pessoas na rua ou mesmo dos passageiros do veículo autônomo. Como a câmera é capaz de captar o infravermelho os faróis convencionais passam a ser desnecessários nos carros autônomos.

11. Sistema de Controle de Posição

O Sistema de Controle de Posição é o sistema responsável por manter o carrinho na trajetória desejada. Para isso ele utiliza o servo motor que controla as rodas da frente de maneira a fazer com que o carrinho se mantenha na trajetória.

Este sistema é, sem dúvida, o principal sistema deste projeto. Depois de implementado ele permite que o carrinho siga uma faixa preta na pista (mesmo que em baixas velocidades) o que permite o desenvolvimento e testes dos demais sistemas.

Entradas: Erro de Trajetória (distância ortogonal entre o centro do carrinho e a trajetória à sua frente; ξ do capítulo 6, Figura 6-1) e Velocidade atual do carrinho (média das velocidades das duas rodas traseiras).

Saída: Sinal PWM para o Servo Motor que controla o ângulo das rodas frontais.

Este sistema é dividido em 3 blocos mais simples: Bloco de Controle, Bloco de Transformação $\alpha(\omega)$ e Bloco de Transformação PWM(α). Um esquemático pode ser visto abaixo na Figura 11-1:

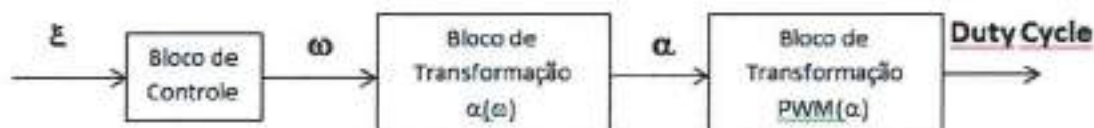


Figura 11-1. Diagrama de blocos do controle de posição.

O Bloco de Controle é o bloco principal deste sistema e foi projetado com base no modelo apresentado no item “Modelo Cinemático do Carrinho” do capítulo 6. Relembrando, a função de transferência deduzida naquele item foi:

$$\frac{\xi(s)}{U(s)} = \frac{Vo + dc * s}{s^2}$$

Também é importante lembrar, como também já foi dito no item “Modelo Cinemático do Carrinho” do capítulo 6, que por ser um sistema tipo II (possui dois polos na origem) ele apresentará erro nulo para a entrada em degrau e para entrada rampa (que são duas interpretações possíveis para o modo como ξ varia). Assim sendo, um controle proporcional será o suficiente para controlar a posição do carrinho.

Apesar de V_0 não ser constante é possível otimizar o controlador em torno de um ponto de operação. Feito isto, durante o funcionamento do sistema V_0 será considerado constante por trechos, ou seja, cada vez que o controle for utilizado, será considerado que o carrinho se encontra em uma velocidade constante igual a sua última velocidade medida.

Com base no capítulo 9, Tabela 3 (*Duty cycle X velocidade de regime*) a velocidade de operação escolhida foi a de $V_0 = 2,4\text{m/s}$ que é a mediana do valor das velocidades e também o primeiro ponto com velocidade de regime acima do estipulado pelo objetivo do projeto (velocidade de reta mínima de $2,0\text{m/s}$).

Realizando a medida do chassi chega-se ao valor de 39 cm para o parâmetro d_c (capítulo 6, figura 6-2, distância entre o eixo traseiro ao ponto onde ξ é calculado). Isto nos deixa com a seguinte função de transferência:

$$\frac{\xi(s)}{U(s)} = \frac{2400 + 390s}{s^2}$$

É importante lembrar que esta função de transferência é válida apenas para a velocidade de $2,4\text{m/s}$ e para a distância de 39 cm do eixo traseiro ao ponto onde ξ é calculado. Ao mudarmos a posição da câmera ou a velocidade do carrinho esta função não será mais válida e o ganho obtido com ela deixará de ser o ganho ótimo. Contudo, estes valores podem ser utilizados para uma primeira aproximação deste ganho e o mesmo pode ter um ajuste mais fino através de testes na pista. Também é interessante notar que, por ter sido escolhido um valor de velocidade intermediário, o ganho escolhido não será muito distante do ótimo enquanto a velocidade média for próxima de $2,4\text{m/s}$.

Outro ponto importante é com relação à dimensão de cada grandeza. No microcontrolador a medida de velocidade é feita em mm/s e a medida de distância em mm. Assim, para manter a consistência dos dados V_0 e d_c foram implementados com estas unidades.

Traçando o lugar geométrico das raízes (no Matlab) para a função acima, obtém-se o seguinte gráfico:

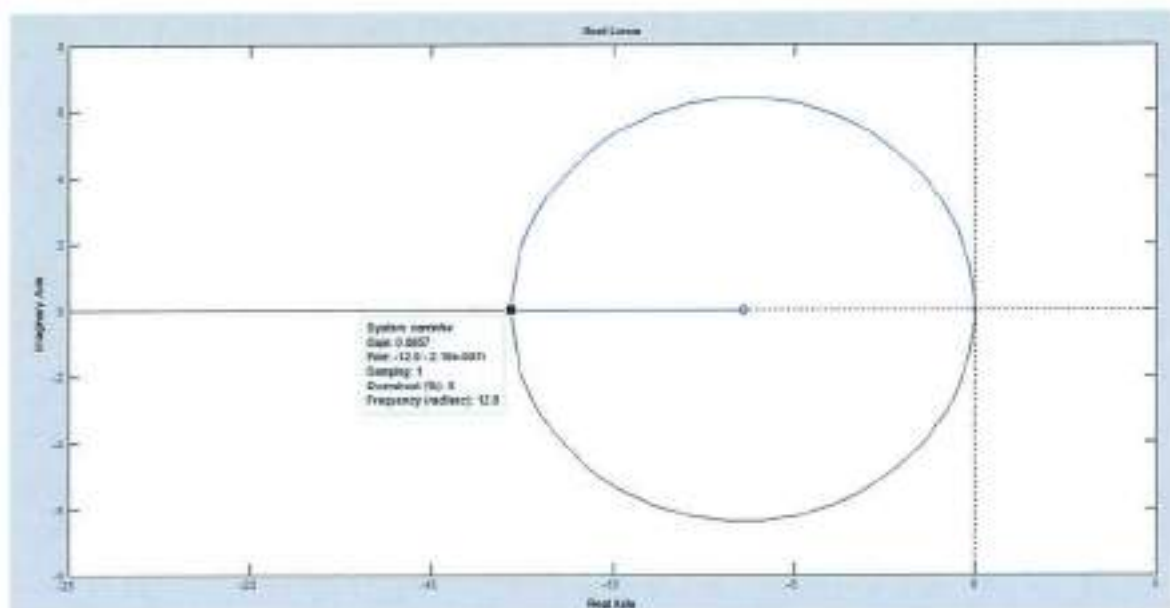


Figura 11-2. Lugar geométrico das raízes para a função de transferência do modelo do carrinho.

O ponto marcado na figura mostra um ganho interessante, pois é onde o sistema apresentará maior velocidade de resposta. Isto porque, movendo este ponto mais para a esquerda o polo da direita se tornará dominante devido à sua resposta mais lenta que o polo da esquerda. Ou seja, a velocidade de resposta do sistema irá diminuir ao invés de aumentar.

Com isso temos que o ganho ideal para nosso sistema será em torno de $K = 0.0657$.

Como o sistema atua nas rodas da frente e a saída do controle é a velocidade angular ω , devemos obter α de maneira indireta. Pela própria dedução do modelo (capítulo 6) vemos que:

$$\alpha(\omega) = \arctan\left(\frac{de}{V_0} * \omega\right)$$

Como está sendo utilizado um microcontrolador, a função $\arctg(x)$ foi interpolada em uma função polinomial de segundo grau (que é facilmente calculada pelo microcontrolador). O resultado desta interpolação pode ser visto na Figura 11-3, sendo o gráfico azul o valor de $\arctg(x)$ e o valor vermelho a aproximação pelo polinômio de segundo grau.

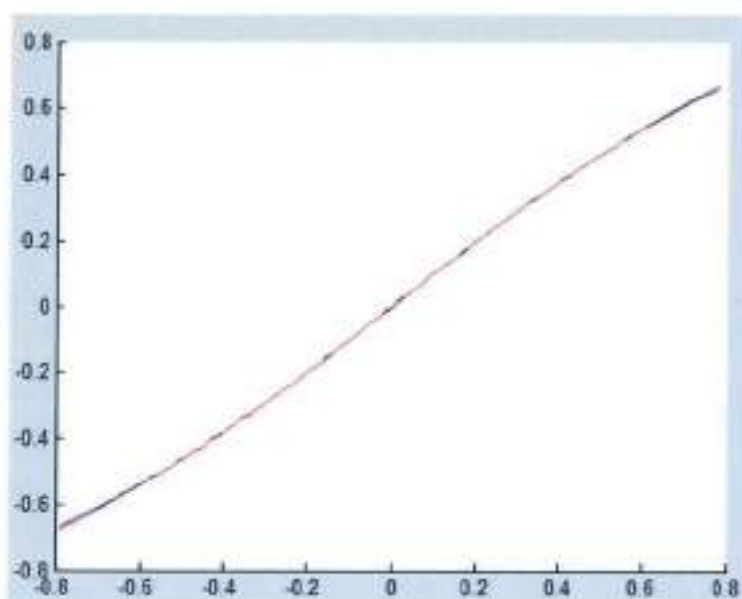


Figura 11-3. Interpolação da função $\text{arctg}()$ por polinômio de grau 2

Foram utilizados apenas ângulos entre -45° e 45° , pois o servo satura em -30° e 30° . Vê-se, pela figura acima, que a aproximação para esta faixa de valores é satisfatória. O script para o traçado deste gráfico (com os respectivos polinômios de interpolação) pode ser visto abaixo:

```
x = [-pi/4:0.01:pi/4];
yaprox = zeros(1,158);

for i=1:158
    if x(i) <=0
        yaprox(i) = (13.651441 * x(i)^2 + 60.177531 * x(i) + 0.132327)*pi/180;
    else
        yaprox(i) = (-13.651441 * x(i)^2 + 60.177531 * x(i) - 0.132327)*pi/180;
    end
    y(i) = atan(x(i));
end
```

Vale observar que há um polinômio para o caso de $x < 0$ e outro para o caso de $x > 0$. Outro ponto importante é que, como o modelo foi configurado em graus a interpolação foi feita para graus. Como a comparação do script acima é feita com radianos, o fator $\pi/180$ foi aplicado nos polinômios de interpolação.

O Bloco de Transformação $\alpha(\omega)$ é responsável pela transformação de ω em α . Contudo é importante perceber uma simplificação implícita nesta transformação: ela admite que para dados valores de V_o e de α , ω já estará determinado. Isto é válido *apenas no caso*

de não haver escorregamento, o que acontece em baixas velocidades, mas deixa de ser verdade em velocidades mais elevadas.

Para corrigir este problema será projetado o controle de estabilidade, que reduz o escorregamento em curvas e faz com que a afirmação anterior seja verdadeira na maior parte do tempo.

Por fim deve ser lembrado que se atua no servo motor enviando um sinal PWM para o mesmo e, portanto, o Bloco de Transformação PWM(α) converte α no *duty cycle* correspondente para virar a roda daquele ângulo. Esta conversão, como foi visto no capítulo 9, é dada pela função abaixo:

$$DutyCycle(\alpha) = 11,88 * \alpha + 1481,57$$

12. Sistema de Controle de Velocidade

Este controle é importante, pois para uma dada potência do motor, as velocidades em uma reta e em uma curva são diferentes, o que não é interessante para o projeto (baixas potências podem fazer com que o protótipo ande em retas, mas não tenha forças para se deslocar em uma curva de raio pequeno).

Esta diferença de velocidade se deve ao fato de que, em uma curva, a potência fornecida pelo motor se divide entre a força lateral (centrípeta) e a força longitudinal (que é quem define a velocidade de deslocamento do carrinho). Já em uma reta, toda a potência do motor pode ser direcionada para a força longitudinal, resultado em uma maior velocidade.

Como foi visto no capítulo 6, motores elétricos, caso sejam lineares, podem ser representados pela seguinte função de transferência:

$$\frac{V(s)}{PWM(s)} = \frac{K_4}{1 + s * T}$$

Já no capítulo 9 (Tabela 3) foram obtidas as velocidades de regime para cada *duty cycle*. Através destes pontos o seguinte gráfico pôde ser obtido:

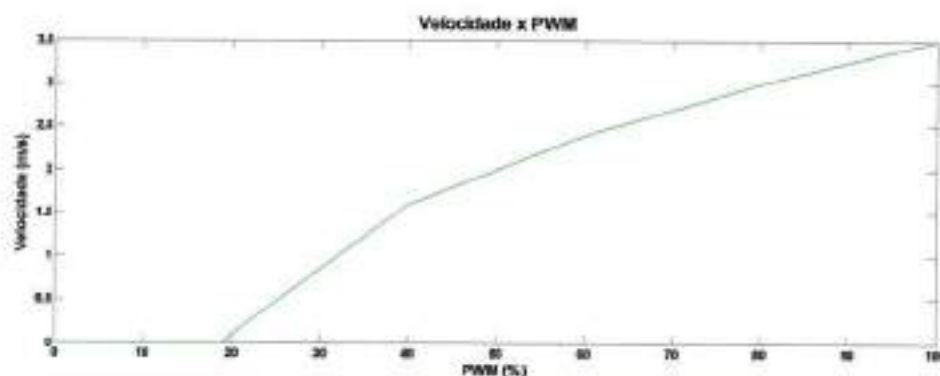


Figura 12-1. Gráfico das velocidades de regime.

Pelo gráfico pode-se ver que os motores não possuem característica linear, mas para *duty cycles* acima de 40% sua forma é aproximadamente linear e, se a linha for prolongada, ela passará próxima do 0. Esta deformação ocorre porque quanto maior a potência no motor, menos relevante será a massa inercial do carrinho. Como se deseja que o carrinho tenha alta performance, ou seja, que os motores funcionem em alta potência a maior parte do tempo, pode-se considerar o modelo linear para o projeto do controle de velocidade.

Para obter as constantes K_4 e T primeiramente divide-se o numerador e o denominador por T , ficando com:

$$V(s) = \frac{K_4/T}{s * (1/T + s)}$$

Anti transformando obtém-se a seguinte função temporal:

$$v(t) = K_4 * (1 - e^{-\frac{t}{T}})$$

Com esta função vê-se que, quando um degrau é aplicado na entrada PWM, $v(t)$ tenderá para K_4 no infinito. E quando tiver decorrido um tempo T o valor de $v(t)$ será de $0,632 * K_4$. Com estes dados é possível obter o modelo dos motores do carrinho através da obtenção dos gráficos de respostas temporais de velocidade do carrinho após a aplicação de um dado PWM.

Assim sendo os motores serão modelados com base nos dados obtidos quando o PWM aplicado foi de 60%, que é a mediana das velocidades e foi a velocidade utilizada para configurar o controle de posição.

Aplicando um PWM de 60% a resposta temporal obtida pode ser visualizada na Figura 12-2 a seguir:



Figura 12-2. Gráfico do PWM em 60%.

Este gráfico foi obtido junto com outros quatro semelhantes através de cinco testes na reta. Em cada um dos testes o software de telemetria era utilizado para obter as velocidades em cada instante. Os outros quatro gráficos podem ser encontrados no Anexo III.

Utilizando a abordagem descrita acima e o gráfico, vê-se que:

$$K_4 = 2.37$$

$$T = 1.556$$

Assim fica-se com o seguinte modelo para os motores do carrinho:

$$\frac{V(s)}{PWM(s)} = \frac{2,37}{1 + s * 1,556}$$

Uma simplificação adotada neste modelamento é a de ambos os motores do carrinho serem idênticos. Isto não é verdade e há uma diferença na curva de potência x velocidade entre eles. A velocidade considerada neste modelamento foi a média entre os dois motores (que pode ser considerada como velocidade do carrinho) por ser uma abordagem mais simples e por ter se mostrado suficiente para este projeto. Contudo, a implementação do controle de estabilidade exige o controle independente dos motores e um ajuste mais fino pode se fazer necessário em outros projetos.

Como este modelo não possui polos na origem ele é um modelo de tipo 0. Isso significa que ele apresenta um erro constante para uma entrada degrau o que não é interessante para o controle de velocidade. Para corrigir este problema um controlador PI (Proporcional-Integral) será implementado. O controlador possui o seguinte formato:

$$G_c(s) = K_{pi} * \left[1 + \frac{1}{s * T_i} \right]$$

Este controlador garante que o sistema seja de Tipo 1, impondo um polo na origem e permitindo que o erro na entrada degrau seja nulo. O parâmetro T_i pode ser escolhido de maneira a cancelar o polo da planta $(1+T)$. Para isto basta fazer com que T_i seja igual a T . Cancelando este polo a malha aberta do sistema passará a ter a seguinte função:

$$G_c(s) * G_p(s) = \frac{K * K_{pi}}{s * T_i}$$

Sendo $K = 2.37$, $T_i = 1.556$, temos que $K * K_{pi} / T_i$ é uma constante. O tempo de subida, de acordo com [13], pode ser calculado pela seguinte fórmula:

$$T_r = \frac{T_i}{K * K_{pi}}$$

Para um tempo de subida de 0,5 segundos K_{pi} deve ser de 1.31. Isto nos deixa com o seguinte controlador:

$$G_c(s) = 1,31 * \left[1 + \frac{1}{s * 1,556} \right]$$

Como o microcontrolador trabalha discretamente a parcela referente ao integrador deve ser aproximada por um somador. Além disso, é necessário evitar que o *duty cycle* do PWM enviado para o motor seja menor que zero ou maior que 100% (saturação). Também é necessário tomar cuidado para que o erro acumulado não "exploda" devido à saturação do atuador (*anti-windup*).

Com todos estes cuidados tomados, a seguinte função é responsável por controlar a velocidade do carrinho:

```
void controlaVelocidade(uint32_t vel_desejada)
{
    float Kpi = 1.31;
    float Ti = 1.556;

    int32_t pwm_motor = 0;
    uint32_t vel_atual = (velocidade[2]+velocidade[3])/2;
    int32_t erro = vel_desejada - vel_atual;
    erro_total = erro_total + erro;
    pwm_motor = Kpi * erro + (Kpi * erro_total)/Ti;

    if(pwm_motor>5000)
    {
        pwm_motor = 5000;
        erro_total = erro_total - erro;
    }
    else if (pwm_motor<0)
    {
        pwm_motor = 0;
        erro_total = erro_total - erro;
    }

    vfnMotor_R_Speed(pwm_motor);
    vfnMotor_L_Speed(pwm_motor);
}
```

Para controlar a velocidade dos motores de forma independente (para que tenham velocidades diferentes entre si) basta duplicar a função e as variáveis acima criando variáveis e funções exclusivas para cada motor. Isto será necessário para que o controle de estabilidade funcione como o esperado.

13. Sistema de Controle de Estabilidade

Como visto no capítulo 5, o controle de estabilidade controla a estabilidade do veículo controlando a velocidade angular do mesmo. Para isto, os equipamentos básicos necessários seriam sensores capazes de medir a velocidade angular do carrinho e atuadores capazes de frear as quatro rodas independentemente.

Normalmente quando se fala em carros autônomos se pensa em seus sistemas de controle de posição e velocidade somente. Contudo há muito mais que um motorista é capaz de fazer para manter o carro em sua trajetória.

Uma destas capacidades está com relação ao controle de estabilidade do carro. Quando sentimos que o carro não está virando tanto quanto queríamos (subviragem) instintivamente paramos de acelerar, reduzindo a força longitudinal, diminuindo o escorregamento das rodas e permitindo que a força de atrito haja estaticamente (já que o coeficiente de atrito estático é maior que o coeficiente de atrito dinâmico).

Um motorista com pouco mais de conhecimento não apenas irá reduzir a aceleração, mas também irá evitar pisar nos pedais de freios (que agem nas rodas frontais) reduzindo muito mais o escorregamento das rodas.

Já um motorista treinado poderá tomar todas as ações descritas e ainda puxar o freio de mão (que atua nas rodas traseiras) causando uma sobreviragem do veículo. Após a curva este efeito deve ser corrigido freando com os pedais novamente e liberando o freio de mão.

Um robô poderá atuar muito mais rápido que um ser humano e não apenas liberando ou acionando freios frontais e traseiros, mas nos 4 freios do veículo de maneira independente e com maior precisão. Isto permite que, quando o controle de estabilidade estiver bem configurado, o motorista sequer sentirá os efeitos de subviragem e sobreviragem enquanto o próprio robô os utilizará para corrigir a trajetória mantendo a velocidade o mais alta possível.

Devido à escolha inicial do chassi o carrinho não possui freios frontais, impossibilitando a correção da sobreviragem da maneira explicada no capítulo 5. Contudo, pelo fato do carrinho possuir motores independentes nas rodas traseiras, é possível realizar este controle através de uma diferença de velocidades entre as rodas traseiras, esta foi a abordagem adotada neste projeto.

O sensor utilizado neste projeto foi um giroscópio (LY530ALH) que possui uma amplitude de $+300^\circ/\text{s}$ até $-300^\circ/\text{s}$. Esta escolha foi feita depois de identificadas em alguns testes, velocidades angulares máximas de $200^\circ/\text{s}$.

O torque causado pela frenagem de uma roda traseira no conceito original é causado basicamente por uma diferença de velocidades entre as rodas traseiras. Como os motores são independentes e controlamos suas velocidades eficientemente, podemos controlar a diferença de velocidade ao invés de frear a roda. Isto resulta em um controle mais suave, o que deve compensar a ausência dos freios frontais.

A Figura 13-1 abaixo demonstra o efeito da diferença de velocidades na velocidade angular do carrinho:

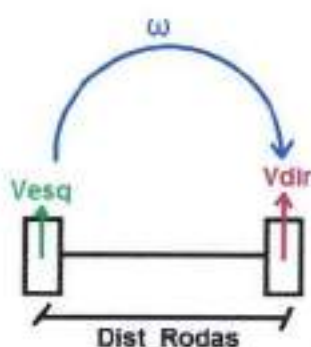


Figura 13-1 - Diagrama de velocidades nas rodas traseiras

Pela figura acima, conclui-se que:

$$\omega = \frac{(V_{esq} - V_{dir})}{\frac{Dist_Rodas}{2}}$$

Considerando que a velocidade desejada seja $V_0 = (V_{esq} + V_{dir})/2$, e isolando as velocidades tem-se que:

$$V_{esq} = V_0 + \frac{\omega * Dist_Rodas}{4}$$

$$V_{dir} = V_0 - \frac{\omega * Dist_Rodas}{4}$$

Estes valores servem para gerar certo ω , mas as rodas da frente já devem fornecer uma porcentagem da velocidade angular desejada. Assim sendo, essa diferença será utilizada para gerar o que falta para atingirmos a velocidade angular restante:

$$\omega_{erro} = \omega_{desejado} - \omega_{medido}$$

$$V_{esq} = V_o + \frac{\omega_{erro} * Dist_Rodas}{4}$$

$$V_{dir} = V_o - \frac{\omega_{erro} * Dist_Rodas}{4}$$

Desta maneira a diferença de velocidades gerada irá apenas fornecer a velocidade angular necessária para a correção já que $\omega_{desejado}$ é fornecido pelo modelo de controle adotado (Capítulo 6) e ω_{medido} é o valor obtido através da leitura do giroscópio.

Um último ajuste viria do fato de que há diversas aproximações neste modelo, como o fato de não haver escorregamento e o fato da velocidade de cada roda se modificar instantaneamente. Para compensar estas aproximações um fator K foi inserido na parcela referente à variação da velocidade desejada (segunda parcela das equações acima). Esta mudança gera uma espécie de controle proporcional. Como se trata de apenas uma constante de ajuste (K), o ajuste foi realizado através de testes de pista e o valor de K encontrado para este projeto foi de 2,5. O que resultou nas seguintes equações para as velocidades:

$$V_{dir} = V_o - \frac{\omega_{erro} * Dist_Rodas}{4} * 2,5$$

$$V_{esq} = V_o + \frac{\omega_{erro} * Dist_Rodas}{4} * 2,5$$

Com estes valores o carrinho corrigia a subviragem satisfatoriamente, mas em altas velocidades apresentava problemas de sobreviragem que não poderiam ser corrigidos sem os freios frontais. Dessa forma, foi decidido que o controle de estabilidade apenas reduziria a velocidade de uma das rodas nas curvas, sem aumentar a da outra. Esta decisão reduziu o efeito da sobreviragem e manteve o controle da subviragem de maneira satisfatória. Assim sendo, como resultado final do controle, as seguintes equações foram obtidas:

Se ω_{erro} for menor que zero:

$$V_{\text{dir}} = V_o$$

$$V_{\text{esq}} = V_o + \frac{\omega_{\text{erro}} * \text{Dist_Rodas}}{4} * 2,5$$

Se ω_{erro} maior que zero:

$$V_{\text{dir}} = V_o - \frac{\omega_{\text{erro}} * \text{Dist_Rodas}}{4} * 2,5$$

$$V_{\text{esq}} = V_o$$

Caso o projeto possua freios frontais é possível corrigir a velocidade angular acelerando a outra roda como proposto anteriormente desde que a sobreviragem seja corrigida com auxílio destes freios.

14. Resultados Obtidos com Sistemas de Controle

Para testar os sistemas desenvolvidos foi utilizada uma pista de cartolina com a faixa central sendo feita de fita isolante preta. Os sistemas foram testados e comparados utilizando-se a medida de qualidade proposta no capítulo 8.

14.1 Sistema de Controle de Posição

Por ser o controle mais básico para o funcionamento do carrinho foi o primeiro a ser testado. Este teste utilizou a obtenção de imagem por largura de pulso (descrito no capítulo 10) que também é a mais simples de ser implementada. Por ser a combinação de sistemas mais básicos para o funcionamento do carrinho, este teste servirá como referência para os seguintes.

Definidos todos os blocos descritos no capítulo 11, foi montada uma reta de 6,5 metros de comprimento (com a configuração dada no capítulo 2) para testar o sistema de controle de posição. Abaixo pode ser vista uma foto da montagem (Figura 14-1).

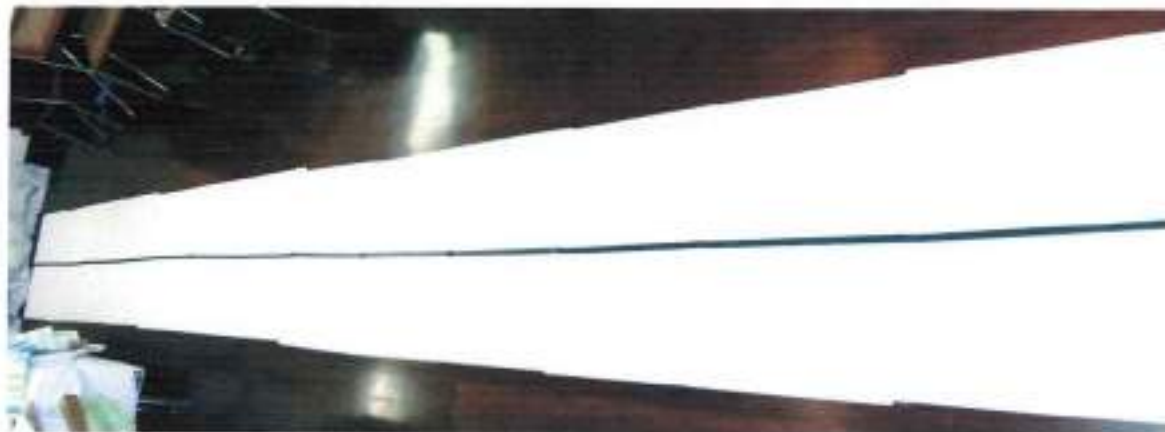


Figura 14-1. Reta montada para a realização dos testes.

O resultado obtido no teste foi satisfatório e pode ser visto abaixo. Como o controle de velocidade ainda não estava pronto foi imposto um PWM constante de 40% (velocidade final de 1,6 m/s). O valor do erro foi obtido do software de telemetria e o gráfico abaixo gerado no Matlab com o arquivo CSV gerado pelo software.

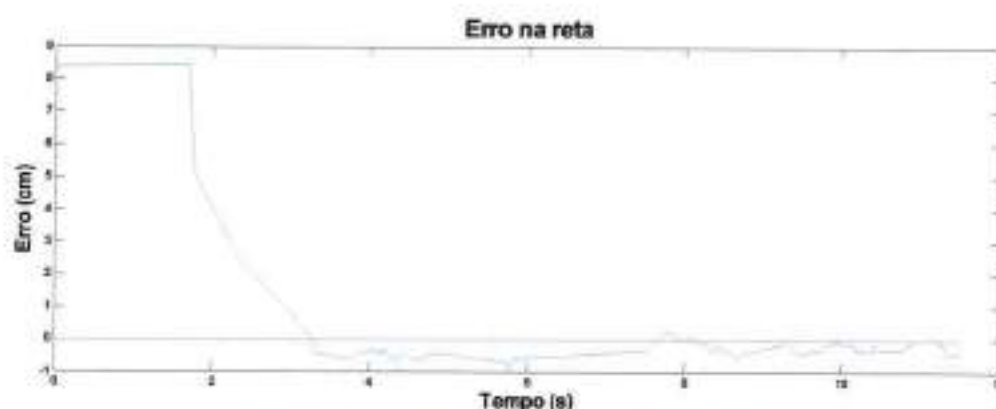


Figura 14-2. Gráfico do erro de posição na reta.

14.2 Sistema de Controle de Velocidade

Este controle não é básico para que o carrinho consiga se manter na pista, mas é essencial para melhorar seu desempenho. Foi o segundo sistema a ser implementado pelo fato de ser necessário para utilizar a medida de qualidade escolhida (é necessário que a velocidade se mantenha constante).

Para este teste novamente foi utilizada a pista reta, de cartolina, descrita anteriormente. O resultado obtido com este controlador para a velocidade de referência de 2 m/s pode ser visto abaixo:

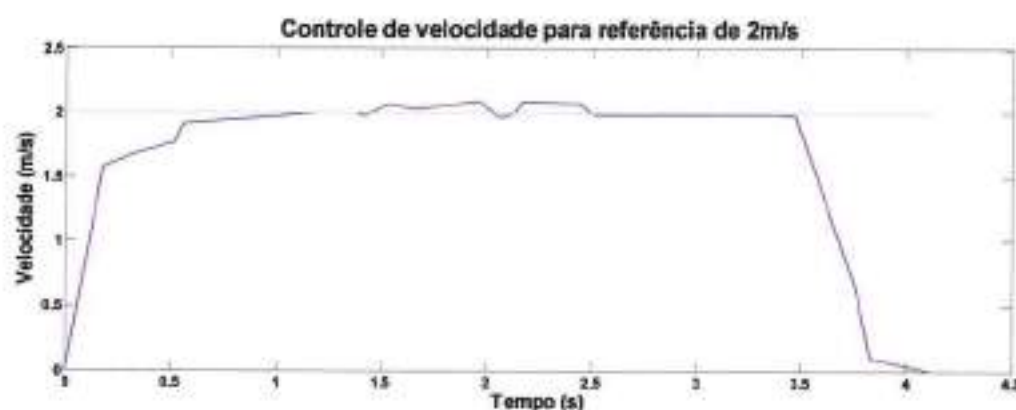


Figura 14-3. Gráfico do controle de velocidade de 2m/s.

Os resultados obtidos com este controlador para as velocidades de referências de 1 m/s e 2,5 m/s podem ser encontrados no Anexo III.

É possível perceber o controle de velocidade atuando e a velocidade se mantendo em torno dos 2m/s. O tempo de subida (intervalo de tempo que o sistema leva para ir de 0,2m/s

até 1,8 m/s) também foi bem próximo dos 0,5 segundos desejados. Assim sendo pode-se dizer que o controle de velocidade atingiu o objetivo estipulado.

14.3 Ganhos de Desempenho em Velocidade Constante

Para estes testes foi montada uma pista, também de cartolina e utilizando fita isolante preta como faixa central. A largura de 60 cm, contudo, não foi mantida durante a pista por problemas de espaço na montagem. Contudo, para garantir o funcionamento dentro dos padrões estabelecidos nos objetivos (capítulo 1), em todos os pontos da pista foi garantido que a largura não ultrapassasse os 60 cm. Ou seja, 60 cm passou a ser a largura máxima da pista, mesmo não sendo constante, o que garante o funcionamento de acordo com os objetivos estabelecidos. A foto abaixo (Figura 14-4) ilustra a pista de testes:



Figura 14-4 - Pista de Testes

14.4 Obtenção de Imagem por Largura de Pulso (referência)

Como explicado anteriormente este sistema é o mais básico e simples de ser implementado. Assim sendo ele servirá como o padrão de referência adotado para medir a melhora de desempenho dos sistemas estudados.

Com este método o carrinho foi capaz de cumprir o objetivo de percorrer a pista sem erros. Com o controle de velocidade e posição, e obtendo a imagem através desta técnica, a velocidade constante máxima alcançada foi de 1,0m/s.

Esta velocidade reduzida se deve à baixa taxa de imagens aproveitáveis. Mesmo com um tempo de integração alto (10ms) a taxa de imagens confiáveis foi, em média, de apenas 66%.

14.5 Obtenção de Imagem por Conversão A/D com Filtro Diferencial Isolado

Como visto no capítulo 10 este método permite o armazenamento da imagem fazendo com que a filtragem possa ser feita por software levando-se em conta a imagem inteira e não apenas valores instantâneos como na técnica de largura de pulso.

Deste modo era possível analisar todos os pontos da imagem e analisar onde estava o verdadeiro pulso negativo que representava a faixa preta. Para isto era analisado principalmente a variação entre os pixels (filtro diferencial) e a largura dos objetos encontrados através das bordas detectadas na imagem. O objeto que possuísse a largura mais próxima da largura da faixa preta seria considerado o referencial. Isto melhorou a taxa de imagens confiáveis para 83%.

Esta maior taxa de imagens melhorou o desempenho do carrinho fazendo com que ele conseguisse se manter em uma pista com velocidade constante máxima de 1,4m/s. Uma melhora de 40% em relação à referência.

14.6 Obtenção de Imagem por Conversão A/D com Algoritmo de Canny

Como visto no capítulo 10 o algoritmo de Canny se utiliza de 2 filtros (Gaussiano e Diferencial) além de um posterior afinamento de bordas. O valor para o qual as bordas devem ser minimizadas varia de acordo com a intensidade luminosa de cada ambiente e deve ser configurado de acordo, como explicado no capítulo 10. Já o valor de σ escolhido para este projeto foi constante e igual a 1,0.

Utilizando estes parâmetros, a taxa de imagens confiáveis teve uma melhora considerável, subindo para aproximadamente 99%. Esta melhor taxa faz com que o controle de posição possa se ajustar mais adequadamente na pista e também garante a detecção de curvas em um tempo hábil suficiente para o carrinho reagir. Por exemplo, com o carrinho andando a 3m/s na primeira técnica (largura de pulso) ele tiraria em média uma foto confiável a cada 18mm já que o carrinho anda 12mm entre cada foto e a chance de ser uma foto aproveitável é de apenas 66%. Já com o algoritmo de Canny e a nova taxa de atualização de 99%, ele passa a andar apenas 12,1mm entre as fotos confiáveis, uma diferença de 0,61 centímetros.

Esta diminuição no tempo em que o carrinho fica sem uma atualização de sua posição faz com que haja uma melhora de desempenho. Com o algoritmo de Canny o carrinho passou a se manter na pista com uma velocidade constante máxima de 1,6m/s. Isto representa uma melhor de 60% em relação à referência e de 14% em relação à utilização do filtro diferencial isolado.

14.7 Sistema de Controle de Estabilidade

Este talvez seja o sistema mais importante com relação ao ganho de desempenho. Primeiro porque ele naturalmente auxilia os sistemas de controle de posição e velocidade a funcionarem como o modelo já que uma das simplificações impostas pelo modelo é a ausência de escorregamento. E segundo porque ele, por conta própria, permite um melhor controle em curvas, fazendo com que o carrinho se mantenha na pista mesmo em velocidades mais altas.

Isto pode ser comprovado após os testes de pista. O resultado obtido foi uma velocidade máxima constante de 2,3 m/s. Mesmo com menor coeficiente de atrito foi obtida uma melhora de 130% em relação ao valor base e de 31,25% em relação ao valor medido com o algoritmo de Canny. Este ganho foi o maior obtido com todos os sistemas utilizados, evidenciando sua absoluta necessidade em projetos de carros autônomos.

O gráfico a seguir (Figura 14-5) demonstra o resultado do controle de estabilidade:

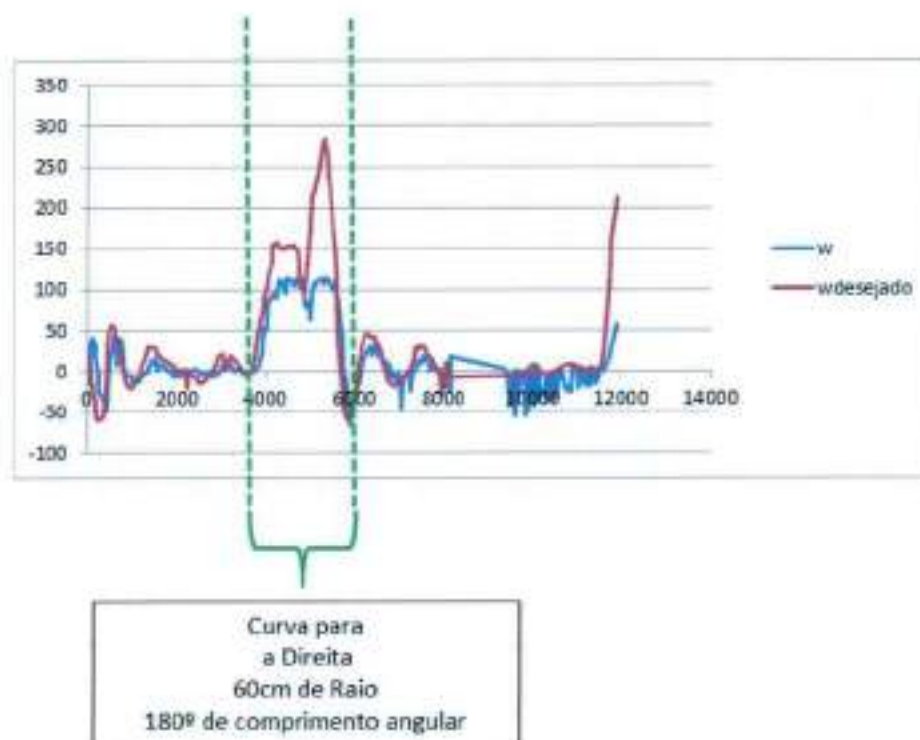


Figura 14-5 - Resultado do Controle de Estabilidade

Pelo gráfico acima (Figura 14-5) podemos ver que durante a curva para a direita o $\omega_{desejado}$ aumenta bruscamente devido ao aumento repentino do erro. Contudo, o controle de estabilidade não acompanha $\omega_{desejado}$ totalmente.

Este efeito ocorre pelo fato de reduzirmos a velocidade de uma das rodas, mas não acelerarmos a segunda (capítulo 13), evitando uma aceleração angular muito alta. Apesar de parecer uma desvantagem pelo fato da velocidade angular do carrinho não acompanhar seu valor de referência, esta limitação também impede o efeito da sobreviragem, aumentando a estabilidade do carrinho em altas velocidades.

Caso as rodas frontais do carrinho possuíssem um sistema de frenagem adequado, o efeito da sobreviragem poderia ser corrigido pelos freios frontais sendo desnecessária esta limitação de não aumentar a velocidade da outra roda no controle de estabilidade.

15. Armazenagem e Correção de Pontos (Mapeamento Local)

Com o giroscópio calibrado e testado, é possível mapear alguns pontos vistos pela câmera. Ou seja, é possível armazenar um ponto visto pela câmera em um determinado instante e, sabendo a velocidade angular e linear do carrinho, é possível identificar a posição deste mesmo ponto em relação ao carrinho em um instante posterior. Esta técnica fez com que a câmera linear passasse a ser mais eficiente que a câmera matricial já que, mirando a linear em um ponto mais a frente, é possível corrigir este ponto e atuar apenas quando ele estiver mais próximo do carrinho, evitando que o carrinho tente virar durante uma reta. A ideia deste funcionamento pode ser vista abaixo:



Figura 15-1 - Esquemático do Mapeamento de Pontos

No exemplo acima as fotos seriam tiradas a cada segundo (lembrando que no projeto este tempo é de 4ms). Os pontos são acumulados na memória do microprocessador e corrigidos de acordo com o movimento do carrinho. A 4 segundos do instante da figura, a foto tirada mostrava a pista aproximadamente no centro e este ponto estava há certa distância, diga-se de 5 metros. Após 4 segundos este ponto se encontra a apenas 1 metro do carrinho. Sabe-se disso porque, medindo a velocidade (através dos sensores Hall descritos anteriormente), vemos que o carrinho se locomove na velocidade de $V_o = 1\text{m/s}$. Fazendo isso constantemente podemos mapear uma série de pontos a frente do carrinho, formando uma imagem matricial do que está a sua frente.

É importante perceber que quando consideramos as correções entre fotos estamos considerando a velocidade linear e a velocidade angular como constantes. Como isto não é real, os deslocamentos angular e linear se tornam aproximados, fazendo com que após cada correção a incerteza com relação ao ponto corrigido aumente. Em outras palavras, quanto

maior o número de pontos acumulados, maior será o erro contido no ponto utilizado para se medir o erro de posição ξ .

Para se corrigir o movimento do carrinho é importante perceber que entre dois instantes de medição ocorre uma translação (devido à velocidade linear) e uma rotação (devido a velocidade angular). Para calcular seus efeitos utilizar-se um sistema cartesiano auxiliar.

Primeiramente consideraremos a translação do carrinho. É importante lembrar que as rodas traseiras não possuem mobilidade e a tração é traseira. Assim sendo, a velocidade em Y será V_0 (velocidade do carrinho) e em X será zero.

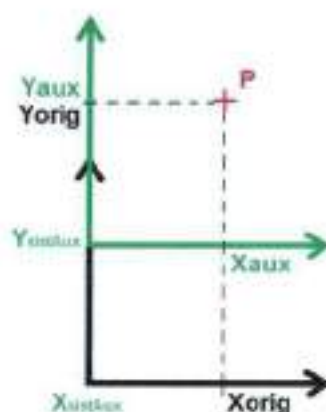


Figura 15-2 - Translação de Sistemas Cartesianos

Lembrando que a origem dos sistemas é o centro do eixo traseiro do carrinho e que sua velocidade é V_0 (sendo considerada constante entre estes 2 instantes). O ponto P é o ponto armazenado na memória com as coordenadas (X_{orig}, Y_{orig}) . Para corrigir o movimento de translação, tem-se que:

$$Y_{sistAux} = V_0 * t_{entreMedidas}$$

$$X_{sistAux} = 0$$

$$Y_{aux} = Y_{orig} - Y_{sistAux}$$

$$X_{aux} = X_{orig}$$

Substituindo a primeira equação na terceira fica-se com as seguintes coordenadas para o ponto P após a translação:

$$Y_{aux} = Y_{orig} - V_o * t_{entreMedidas}$$

$$X_{aux} = X_{orig}$$

Sendo $t_{entreMedidas}$ o tempo decorrido entre a correção anterior e esta (geralmente o intervalo de tempo entre duas fotos), os valores acima serão a posição corrigida do ponto P após uma translação.

Para deduzir a rotação o processo é um pouco mais complicado, mas a figura abaixo auxilia o entendimento:

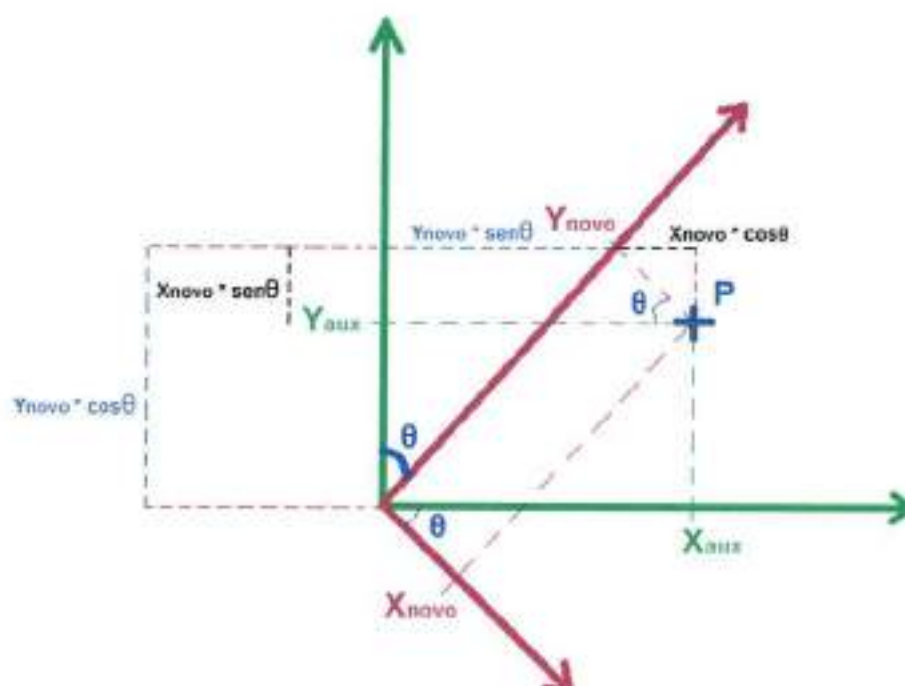


Figura 15-3 - Rotação de Sistemas Cartesianos

Vê-se pela Figura 15-3 acima que, após rotacionar o sistema de um ângulo θ e projetá-lo adequadamente, as novas coordenadas ficam:

$$X_{aux} = X_{novo} * \cos\theta + Y_{novo} * \sin\theta$$

$$Y_{aux} = Y_{novo} * \cos\theta - X_{novo} * \sin\theta$$

Como as coordenadas que interessam são as novas, reorganizando e resolvendo as equações em função de X_{novo} e Y_{novo} , temos:

$$X_{novo} = X_{aux} * \cos\theta - Y_{aux} * \sin\theta$$

$$Y_{novo} = X_{aux} * \sin\theta + Y_{aux} * \cos\theta$$

Matricialmente obtém-se a já esperada matriz de rotação:

$$\begin{bmatrix} X_{novo} \\ Y_{novo} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} X_{aux} \\ Y_{aux} \end{bmatrix}$$

Utilizando o princípio da superposição (válido por ser um problema linear) podemos substituir X_{aux} e Y_{aux} pelos valores obtidos na translação. Isto nos deixa com o seguinte sistema de correção de pontos:

$$\begin{bmatrix} X_{novo} \\ Y_{novo} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} * \begin{bmatrix} X_{orig} \\ Y_{orig} - V_o * t_{entreMedidas} \end{bmatrix}$$

Para obter θ deve-se considerar uma velocidade angular (medida com o giroscópio) constante e multiplicar pelo mesmo $t_{entreMedidas}$ pelo qual V_o é multiplicado na equação acima.

Um último detalhe para a correção é a obtenção dos valores dos cossenos e senos de θ . Assim como para o arco tangente, não é possível obter estes valores de forma precisa no microcontrolador. Assim sendo, foi elaborada, com ajuda do Matlab, uma interpolação polinomial de grau 3 (pois grau 2 apresentava erros consideráveis) para a aproximação do cosseno. A aproximação encontrada pode ser vista abaixo:

$$\cos\theta = 0.00000060516 * \theta^3 - 0.00018468 * \theta^2 + 0.00061287 * \theta + 0.9974$$

A interpolação acima é válida de 0° a 90° . Para os valores entre 90° e 180° é utilizada a propriedade da paridade ímpar em torno de 90° ($\cos(90^\circ + \theta) = -\cos(90^\circ - \theta)$), para ângulos negativos é utilizada a propriedade da paridade par em torno do 0° ($\cos\theta = \cos(-\theta)$) e para ângulos fora do intervalo entre -180° e $+180^\circ$ é utilizada a propriedade da periodicidade ($\cos\theta = \cos(\theta - N*360)$, para qualquer N inteiro).

A aproximação acima pode ser vista graficamente na Figura 15-4 a seguir:

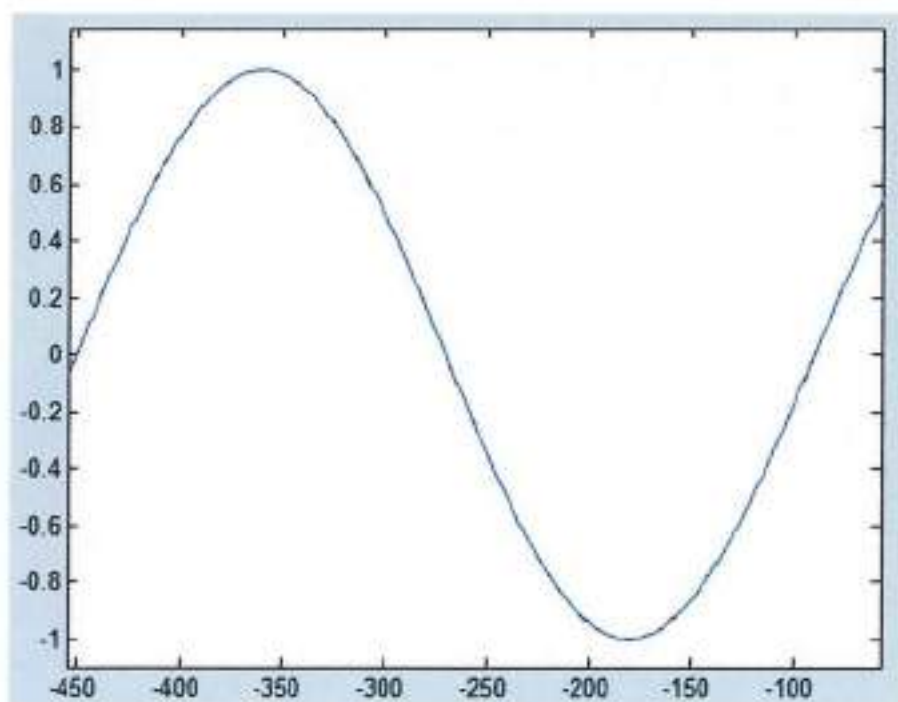


Figura 15-4 - Aproximação da função Cosseno

E com maiores detalhes (entre -15° e 15°) na Figura 15-5 seguinte:

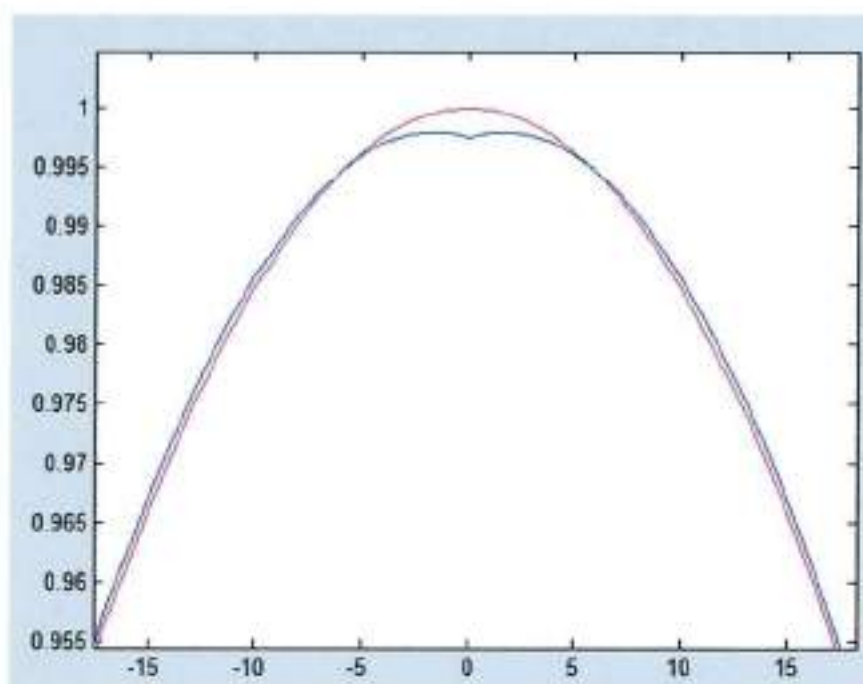


Figura 15-5 - Aproximação da função Cosseno entre -15° e 15°

Vê-se que mesmo no cume (onde a função apresenta uma descontinuidade devido à falta de paridade da função polinomial de terceiro grau escolhida) o erro é de cerca de 0,5%.

Para a função seno basta calcular o cosseno defasado de 90° já que $\sin\theta = \cos(\theta-90)$.

Com isto o sistema está apto a mapear localmente a pista. O mapeamento de grandes distâncias com esta técnica é desaconselhável visto que o erro acumulado em pontos corrigidos diversas vezes passa a ser muito alto.

15.1 Determinação de estado: Reta ou Curva

Um ponto importante para este projeto é a determinação se o carrinho se encontra em uma reta ou uma curva. É importante saber isto, pois dependendo de sua situação podemos tomar diferentes atitudes como a decisão da velocidade, o acionamento de freios, a referência do controle de posição entre outras coisas. Neste capítulo serão estudadas três técnicas para a determinação se o carrinho se encontra em uma reta ou em uma curva.

15.1.1 Pelo Raio de Curvatura

Uma das maneiras de se descobrir a situação atual do carrinho é calcular o raio de curvatura de 3 pontos armazenados já que 3 pontos definem uma circunferência. Se o raio estiver acima de um determinado limite (que deve ser escolhido de acordo com a escala do veículo) o carrinho se encontrará em uma reta. Para este projeto o valor limite escolhido foi de 3 metros já que em uma curva com este raio as ações tomadas seriam as mesmas que em retas.

A equação que define pontos pertencentes a uma circunferência pode ser vista abaixo:

$$R^2 = (x - x_c)^2 + (y - y_c)^2$$

Sendo R o raio da circunferência e (x_c, y_c) a localização do centro dela podemos considerar 3 pontos (x_1, y_1) , (x_2, y_2) , (x_3, y_3) pertencentes a esta circunferência. Substituindo estes pontos em X e Y da equação acima e resolvendo o sistema encontrado, obtém-se as seguintes equações para encontrar o raio desta circunferência:

$$\alpha = \frac{x_1^2 - x_2^2 + y_1^2 - y_2^2}{2(y_1 - y_2)}$$

$$\beta = \frac{x_1 - x_2}{y_1 - y_2}$$

$$x_c = \frac{x_1^2 - x_3^2 + (y_1 - y_3)(y_1 + y_3 - 2\alpha)}{2[x_1 - x_3 - \beta(y_1 - y_3)]}$$

$$y_c = \alpha - \beta x_c$$

$$R = \sqrt{(x_1 - x_c)^2 + (y_1 - y_c)^2}$$

Com estas equações podemos calcular o valor de R. Diferente de outras funções não lineares abordadas neste projeto, a raiz quadrada não pode ser aproximada por um polinômio. Assim sendo a maneira escolhida para se calcular o valor de R foi numericamente, através do algoritmo de Newton-Raphson com a derivada aproximada por secantes [14].

O algoritmo (feito no Matlab e posteriormente aplicado ao microcontrolador) que demonstra a aplicação do método para encontrar a raiz quadrada do número Rquad pode ser encontrado no Anexo V.

Este método é interessante para quando há uma boa precisão no algoritmo de correção de pontos e quando é possível obter pontos mais espaçados do que neste projeto. Com o tempo de 4ms por foto os pontos estão distanciados de apenas 1,2cm (considerando uma velocidade de 3m/s). Em uma curva de raio 50cm (considerado para este projeto como sendo o mínimo raio de curvatura) um espaçamento de 1,2cm (na direção do movimento do carrinho) entre os pontos significa uma variação no erro de apenas 1,44mm. Para curvas com raio acima de 50cm esta diferença é ainda menor.

Considerando que a câmera possui uma imprecisão de pelo menos 1 pixel (cerca de 2,3mm) uma variação de 1,44mm pode ser considerada pequena demais para esta detecção. Isto sem considerar os erros acumulados devido à correção dos pontos armazenados.

De fato, após 100 tentativas de identificação de curvas e retas, o algoritmo obteve 54% de acerto, índice de acerto baixo para ser considerado confiável. Contudo, é importante lembrar que isto se deve a características próprias deste projeto, mas este método pode ser utilizado em projetos semelhantes onde há uma maior precisão na leitura e/ou uma maior distância entre os pontos analisados.

15.1.2 Pelo Coeficiente Angular

Um método prático e mais simples que o anterior. Ao invés de serem considerados três pontos são considerados apenas dois. Feito isto medimos o coeficiente angular destes pontos. Além de identificar se o carrinho se encontra em uma curva ou uma reta este método também é capaz de facilmente identificar para que lado (direita ou esquerda) é a curva já que curvas para a direita apresentam coeficiente angular positivo enquanto para a esquerda apresentam coeficiente angular negativo.

Além disso, medindo a variação destes coeficientes angulares ainda é possível perceber se o carrinho está entrando, saindo ou se está no meio de uma curva. Lembrando que o eixo y está na direção do movimento do carrinho (neste projeto), podemos obter o coeficiente angular entre dois pontos da seguinte maneira:

$$coefAngular = \frac{x_1 - x_2}{y_1 - y_2}$$

Este método é certamente o mais aconselhável dos três métodos expostos neste projeto. Ele é o mais simples e rápido de ser calculado além de apresentar as vantagens de saber se é a entrada ou saída de uma curva.

Infelizmente ele possui o mesmo problema que o anterior para ser aplicado a este projeto. Devido à baixa variação do erro entre dois pontos o coeficiente angular entre eles acaba se tornando impreciso. Após 100 tentativas de identificação de curvas e retas o algoritmo obteve 73% de acerto, a maior parte deles concentrada na entrada e saída das curvas.

15.1.3 Pelo Erro Acumulado

Um terceiro método pensado pelo grupo utiliza o erro acumulado para dizer se o carrinho se encontra em uma reta ou em uma curva. A ideia é simples e se mostrou muito eficiente para este projeto.

Durante uma reta o carrinho acumula pouco erro já que além dos erros serem pequenos, eles ocorrem do lado positivo e do lado negativo, fazendo com que um colabore com a diminuição do outro. Já em uma curva este erro passa a ser sempre para um lado e em valores maiores. Por exemplo, em uma curva de 50cm, mesmo com o carrinho seguindo perfeitamente a linha o erro 30cm à frente dele (local onde a câmera enxerga) é de 6cm.

Como o carrinho foi projetado para zerar este erro apenas quando o carrinho estiver passando por aquele ponto (e não quando a câmera estiver enxergando ele) este erro de 6cm será constante durante toda a curva.

O resultado é que este erro se torna consideravelmente alto durante as curvas e consideravelmente baixo durante as retas. Além disso, ainda é possível dizer se a curva é para a direita ou para esquerda (erro acumulado positivo ou negativo respectivamente).

Este método, para este projeto, foi o mais preciso. Em 100 tentativas de identificação de curva ele obteve um índice de acerto de 89%, errando principalmente nas entradas e saídas das curvas onde o erro acumulado era baixo por ainda não ter se acumulado (entrada) ou por já ter se reduzido demais (saída).

15.1.4 O Método Escolhido

Para este projeto foi decidido que a maneira mais eficiente de se detectar se o carrinho se encontrava em uma curva ou em uma reta era a mistura dos dois últimos métodos. Eles são simples e rápidos de serem calculados, e enquanto um deles apresenta maior precisão na detecção de entrada e saída das curvas o outro mostra maior precisão no meio delas.

A união dos métodos apresentou um índice de acerto de 96% em 100 tentativas. Os poucos erros eram referentes à entrada e saída das curvas, fato que não influencia muito já que, nas poucas vezes que erra, logo será acertado quando atingir o meio da curva.

16. Otimizações com Velocidades Variáveis

16.1 O Ganho Integral em Curvas

Como visto anteriormente (capítulo 6), o sistema pode ser controlado de maneira estável e eficaz com um controlador P. Contudo, pensando na melhora do desempenho, o ganho integral em curvas pode ser útil.

O erro acumulado em curvas possui um valor elevado. Quando este erro é multiplicado por um fator (ganho integral) e somado à entrada do sistema, o carrinho passa a virar mais que o necessário para seguir a pista. Se este efeito é ruim em retas (pois faria com que o carrinho saísse da linha) ele é altamente desejado em curvas já que faria com que o carrinho percorresse o lado de dentro das curvas. Quanto mais longo for o trecho com curvas, mais interno à curva o carrinho ficará. Isto auxilia não apenas o fato de percorrer o menor caminho, mas também o mantém mais afastado do ponto de saída mais arriscado em uma curva que é o lado de fora da mesma.

Com o auxílio de um detector de estado que indica se o carrinho se encontra em uma Curva ou uma Reta (descrito no capítulo 15) é possível que este ganho esteja ativo apenas nas curvas (otimizando a velocidade de curva) e não influencie nas retas (não diminuindo a velocidade de reta).

16.2 Decisão de Velocidade

No capítulo 14 os sistemas foram avaliados pela velocidade constante máxima atingida com eles. Esta velocidade é a velocidade mínima que o carrinho deve ter nas curvas, mas nas retas o mesmo pode utilizar uma maior potência dos motores.

Um controle de velocidade simples de ser pensado seria uma variação brusca de acordo com o erro de posição, por exemplo, para erros acima de 9 cm o carrinho poderia manter uma velocidade de 2,3m/s (velocidade máxima atingida com o controle de estabilidade) e para erros menores que este utilizar a velocidade máxima (cerca de 3,5m/s).

O problema desta abordagem é que ela pode instabilizar o carrinho visto que gera acelerações (mudando de 2,3m/s para 3,5m/s) e frenagens (mudando de 3,5m/s para 2,3m/s) de maneira repentina. Além disso, iniciar a frenagem que antecede uma curva

quando o erro já for de 9 cm não permite que a velocidade do carrinho diminua o suficiente antes de iniciar a curva.

A solução encontrada foi a utilização de uma tabela de velocidades que varia segundo o erro. A Tabela 14 abaixo ilustra o resultado final:

Tabela 14 - Variação de Velocidade de Reta Segundo o Erro de Posição Medido

Erro (mm)	Velocidade
0 - 30	velReta
31 - 40	velReta
41 - 50	0,9*velReta
51 - 60	0,8*velReta
61 - 70	0,7*velReta
71 - 80	0,5*velReta
81 - 90	0,4*velReta
91 - 100	0,4*velReta
100 - 110	0,4*velReta
Maior que 110	0,3*velReta

Nela a velReta é a velocidade máxima de reta configurada no carrinho. No caso deste projeto ela foi de 3,6m/s o que significa uma velocidade mínima de 1,08 m/s.

As porcentagens escolhidas para cada erro, assim como o espaçamento escolhido entre os erros foram escolhidos de maneira a deixar a variação de velocidade suave o suficiente para não ocorrer oscilações na reta devido a mudanças bruscas de velocidade e também para frear com tempo suficiente para realizar uma curva no final de uma longa reta. Os valores tabelados foram encontrados empiricamente.

Contudo, ao utilizar este método o carrinho acaba ficando muito lento nas curvas. Como explicado acima, durante uma curva o carrinho enxerga um erro mínimo de 6cm, mas este erro é, em geral, maior. Isto significa uma velocidade de 1,4 m/s em média. Este valor é muito abaixo da velocidade atingida com o controle de estabilidade (2,3m/s). Uma solução simples seria aumentar a velocidade para os erros de 7cm e 8cm, mas isto faria com que o carrinho não conseguisse frear no fim da reta.

Para resolver isto o sistema de decisão de velocidade considera o estado do carrinho (se o mesmo está em uma reta ou em uma curva) para decidir qual regra utilizar. Caso o carrinho esteja em uma reta ele utilizará a tabela acima. Caso o carrinho esteja em uma

curva ele utilizará a velocidade constante de 2,3m/s (que foi a melhor velocidade constante obtida com os sistemas desenvolvidos).

Esta decisão faz com que ele desacelere o suficiente antes de uma curva, e, uma vez nela, volte a acelerar ganhando velocidade para a reta seguinte.

Com este sistema de decisão de velocidade foi possível atingir uma velocidade máxima de 3,8 m/s na reta e de 2,3 m/s na curva. Durante as retas a velocidade média variava entre 2,4 m/s e 3,5 m/s dependendo do comprimento da reta, o que está de acordo com o objetivo proposto.

17. Conclusões

17.1 Análise do Projeto

Durante o desenvolvimento deste projeto pudemos analisar alguns dos sistemas necessários nos carros autônomos. Para comprovar os conceitos estudados foi desenvolvido um protótipo 1:16 que utilizava os sistemas desenvolvidos para tentar melhorar seu desempenho enquanto percorria a pista.

Vimos que o sistema mais básico para o funcionamento, controle de posição, exige um controle proporcional simples, devido ao modelo escolhido para este projeto. Com apenas esses sistemas e as estruturas elétricas e mecânicas já era possível fazer com que o protótipo se mantivesse na pista, mas em baixas velocidades.

Melhorando a metodologia de obtenção da imagem pudemos obter uma taxa maior de imagens confiáveis, o que influenciava diretamente no desempenho do protótipo. Isto porque quanto mais rápido estiver o protótipo, mais ele andarão no tempo entre duas fotos (tempo no qual sua referência estará desatualizada). Ou seja, quanto menos fotos forem desperdiçadas menos o protótipo andarão com a referência desatualizada, detectando a curva a tempo suficiente para frear mesmo em velocidades mais altas.

Neste sentido foram estudados dois modos de obtenção de imagem: um com circuito externo usando comparador, e um usando conversão analógica/digital. O da conversão analógica/digital apresentou vantagens significativas em relação à obtenção por largura de pulso. Com relação à filtragem após a obtenção da imagem pela conversão analógica/digital também foram estudados duas maneiras. Como esperado o algoritmo de Canny apresentou um desempenho muito superior ao filtro diferencial isolado, até mesmo por incluir o filtro diferencial em um dos passos do algoritmo.

Terminados os sistemas básicos e otimizado a obtenção e filtragem de imagens, o sistema de estabilidade foi estudado. Este sistema não é necessário para baixíssimas velocidades, mas é fundamental quando desejamos velocidades práticas para o dia a dia.

Apesar da grande maioria dos carros atuais não contarem com sistemas de controle de estabilidade automático, os seres humanos tomam as atitudes básicas necessárias para manter a estabilidade do carro. Quando retiramos o ser humano do volante torna-se necessário substituir não apenas as tarefas básicas como controle de posição e velocidade, mas também as secundárias como controle de estabilidade e frenagem (não abordado neste

projeto), por exemplo. Se considerarmos que a principal motivação para os carros autônomos seria a possibilidade de andar em velocidades maiores e de maneira mais segura, estes sistemas secundários como o controle de estabilidade se tornariam imprescindíveis, justificando seu estudo neste projeto.

De fato foi possível perceber que o maior ganho de desempenho no protótipo foi visto quando o controle de estabilidade proposto foi implementado. É interessante perceber que há diversas maneiras de se implementar este controle e a utilizada neste projeto foi simplificada, visto a ausência de freios frontais. Ainda assim o ganho foi considerável, evidenciando a importância do sistema em pesquisas do gênero.

Foi também estudada a possibilidade de mapeamento de pontos, fazendo com que um pequeno mapa pudesse ser desenhado na memória do protótipo. Isto torna possível analisar a trajetória ideal e a antecipação de obstáculos (como curvas). Contudo é importante que os sensores utilizados para o mapeamento (giroscópio e o conjunto de sensores Hall e ímãs de neodímio) tenham medidas confiáveis do deslocamento angular e do deslocamento linear. Devido à incerteza dos sensores deste projeto, foi possível mapear cerca de 6 pontos de maneira confiável, o que equivale a uma distância de aproximadamente 7 cm a frente do ponto de medição sem mapeamento. Ou, em outras palavras, foi possível enxergar os 5 próximos pontos que seriam utilizados como referência para calcular o erro da posição. Quanto maior for a precisão dos sensores, maior será o número de pontos que podem ser armazenados e corrigidos antes de serem utilizados para calcular o erro de posição.

Com este mapeamento de pontos foi possível estudar algumas técnicas para detectar se o veículo se encontra em uma reta ou em uma curva. Este conhecimento é importante já que, normalmente, o comportamento em uma reta é diferente do comportamento em uma curva. Após estudarmos algumas possibilidades vimos que uma combinação das técnicas de detecção pelo coeficiente angular e pelo erro acumulado seria o ideal para este projeto. Todavia, caso o projeto consiga pontos mais afastados entre si (caso mais provável em protótipos de escala maiores) a equipe aconselha a utilização da técnica do coeficiente angular por sua simplicidade e eficiência.

Por fim, foi analisada a decisão sobre a velocidade desejada. Esta etapa é de importância fundamental já que a velocidade deve mudar o tempo todo. No caso deste

projeto isto acontece na entrada e saída de curvas onde o protótipo deve reduzir e aumentar sua velocidade, respectivamente. Ele também deve manter, durante curvas longas, uma velocidade maior que a da freada antes da curva, mas menor do que a velocidade de reta. Em carros reais além das curvas há também o fato de haver outros carros e objetos em volta, o que torna a variação de velocidade ainda mais importante.

Para este projeto foi desenvolvida uma tabela de variação de velocidade de acordo com o erro medido. Isto porque na entrada da curva este erro vai aumentando gradativamente, portanto, com este sistema, o protótipo passará a reduzir sua velocidade ao se aproximar da entrada da curva.

O detalhe de se adicionar um ganho proporcional na curva foi uma solução simples e eficiente de se fazer com que o protótipo fizesse as curvas de maneira mais fechada, com raio menor. Contudo, caso seja possível a obtenção de mais pontos no mapeamento, é recomendável a detecção de entrada e saída de curva com maior precisão do que a deste projeto fazendo com que seja possível controlar o raio da curva executada pelo veículo de maneira mais precisa e mais eficiente. Ou seja, abrindo antes da curva, fechando durante a curva e abrindo ao final da curva. Executar curvas com o maior raio possível e passar pela tangente faz com que o veículo tenha maior aderência (pois a força centrípeta necessária para uma curva de raio grande é menor) e, portanto, possa passar pela curva a uma velocidade maior.

Além do protótipo desenvolvido para testar estes sistemas também foi desenvolvido um software de telemetria para quantificar a melhoria adquirida com cada sistema. Este software pode ser adaptado a outros projetos semelhantes, pois sua interface com o protótipo se faz através de códigos com 3 letras, então basta o novo projeto enviar as 3 letras correspondentes da medida, ou reprogramar o software para aceitar mais códigos e/ou códigos diferentes.

Tendo em vista os resultados finais, a equipe conclui que o resultado obtido com o projeto foi muito satisfatório, pois foi atingido o objetivo proposto inicialmente e também foi gerada uma documentação sobre sistemas que podem ser utilizados em carros autônomos, incentivando futuras pesquisas na área, que era a motivação inicial.

17.2 Propostas de Continuidade

Por ser um tema amplo há diversas maneiras de se continuar este projeto, contudo iremos nos focar em próximos passos que podem ser dados a partir deste projeto.

17.3 Aumento da Escala

Este projeto foi feito em uma escala 1:16, o que serve para uma análise do funcionamento e integração dos sistemas, mas não é possível considerar todas as propriedades de carros reais, como a massa do veículo, de forma satisfatória.

Assim sendo, a primeira sugestão de continuidade do projeto seria com relação ao aumento da escala para uma escala onde a massa passa a ser um fator a ser levado em consideração e também onde a pista utilizada já possa ser asfaltada.

Apesar de um carro em escala natural ser a principal opção, também podem ser estudados veículos menores como um kart, um quadriciclo (ATV) ou mesmo veículos em escala 1:10 (ou maior) rádio controlados com motores de combustão interna. Estes veículos possuem mecânicas mais próximas de carros reais e, por serem menores e mais baratos, podem servir como um próximo passo.

Como nestes casos as forças atuantes passam a ser maiores um modelo envolvendo as forças pode vir a ser necessário. Uma referência possível para este modelo seria [1].

17.4 Microcontrolador para Telemetria

Neste projeto utilizamos um único microcontrolador para executar as funções necessárias e também para enviar dados da telemetria para o software de telemetria. Como havia poucas medidas a serem realizadas e a velocidade do protótipo não era elevada, a taxa de amostragem poderia ser baixa e ainda assim suficiente para os propósitos deste projeto.

Contudo, ao modificarmos o veículo para algum que se move mais rapidamente, uma baixa taxa de amostragem poderá ser prejudicial e uma alta taxa de amostragem poderá influenciar no funcionamento dos sistemas. Assim, uma sugestão seria a utilização de um segundo microcontrolador (não necessariamente de 32 bits) capaz de realizar as conversões analógicas/digitais necessárias e enviar e receber dados do PC. Este microcontrolador receberia os dados dos sensores de maneira paralela ao microcontrolador principal. Estes dados seriam retransmitidos ao software de telemetria que realizaria cálculos semelhantes

ao microcontrolador principal, mas de maneira mais eficiente devido à maior capacidade de processamento.

Esta modificação faria com que os dados fossem obtidos com uma precisão maior e não influenciaria o funcionamento do carrinho já que não teria comunicação com o microcontrolador atual e nem com os atuadores.

17.5 Interpretação de Sinais na Pista

Neste projeto nos preocupamos apenas em detectar a linha preta, o que é o suficiente para cumprir o objetivo proposto. Contudo, sabemos que no dia a dia há diversas sinalizações que devem ser captadas e interpretadas por um carro autônomo.

Os algoritmos estudados neste projeto para a obtenção e filtragem da imagem permitem detectar as bordas das faixas pretas no chão. Uma parte da imagem é considerada objeto entre um mínimo local e um máximo local sem que haja outros mínimos e máximos entre eles. Ou seja, os algoritmos desenvolvidos são capazes de detectar não apenas a faixa preta central da pista, mas qualquer faixa preta que tiver na pista. Cada objeto possui sua posição de início, fim e sua largura.

Assim sendo, uma terceira sugestão de continuidade seria o estudo de sinalização e a identificação desta sinalização pelo carrinho. Por exemplo, uma faixa preta de 10 cm distante 3 cm da faixa central pode significar que se deve parar, mas caso haja duas faixas pretas de 10 cm distantes 3 cm da faixa central (uma faixa de cada lado da faixa central, como ilustra o desenho abaixo) o protótipo deverá reverter os motores.

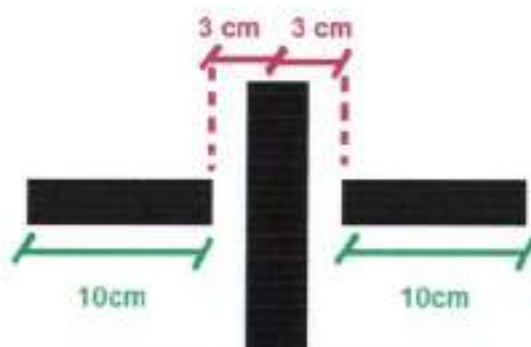


Figura 17-1 - Exemplo de Sinalização

A utilidade disso é que, em estradas onde apenas carros autônomos seriam permitidos pode ser que sinalizações no chão sejam mais facilmente interpretadas do que no alto (que é mais facilmente interpretada por seres humanos).

17.6 Identificação de Obstruções da Pista

Uma última sugestão de continuidade seria referente à possibilidade de obstrução da pista. Neste projeto o ambiente era controlado e, portanto, garantido que a pista estaria sempre livre. Contudo esta hipótese não é verdadeira em um ambiente real e, portanto, o veículo deve ser capaz de desviar ou pelo menos evitar a obstrução a sua frente, seja ela móvel ou fixa.

Para que isto seja possível uma sugestão seria a utilização de sensores de ultrassom já que os mesmos informam a distância até a obstrução e, através do efeito Doppler, ainda seria possível medir a velocidade da obstrução, se esta for móvel. Este dado é útil caso se considere a possibilidade de desviar da obstrução, ou mesmo a possibilidade da obstrução já ter passado (no caso de obstruções móveis) quando o veículo atingir aquele ponto.

18. Bibliografia principal para o trabalho

- 1) CASTRUCCI, Plínio B. L.; BITTAR, Anselmo; SALES, Roberto M. **Controle automático**. 1ªed. Rio de Janeiro: LTC, 2011.
- 2) OGATA, Katsuhiko. **Engenharia de controle moderno**. 5ªed. São Paulo: Pearson Education, 2010.
- 3) RAJAMANI, Rajesh. **Vehicle dynamics and control**. Londres: Springer, 2005.
- 4) SILVA, Rodney A. B., **Implementação de um robô móvel dotado de um sistema de guiagem baseado em processamento digital de imagens e técnicas avançadas de controle**. 2004. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, 2004.
- 5) RUSS, John C. **The image processing handbook**. 5ªed. Nova Iorque: CRC, 2006.
- 6) SIEGWART, Roland; NOURBAKHS, Illah R.; SCARAMUZZA, Davide. **Introduction to autonomous mobile robots**. 2ªed. Londres: MIT, 2011.
- 7) KASSAB JR, F.; CRUZ, J.J.; AYA, J. C.; MARQUES, R. P. **Laboratório de controle**. São Paulo: Epusp. 2011. Apostila para a disciplina de graduação do Departamento de Automação e Controle. PTC- 2512 – Laboratório de Controle.
- 8) RUGGIERO, Márcia A. Gomes; LOPES, Vera Lúcia da Rocha. **Cálculo numérico: Aspectos Teóricos e Computacionais**. 2ªed. São Paulo: Pearson Makron Books, 1996.

19. Resumo do conteúdo da bibliografia principal

Os números a seguir são correspondentes às referências citadas no item anterior.

- 1) Controle automático será uma excelente fonte de consulta quando estivermos projetando a malha de controle ou quando estivermos caracterizando o sistema.
- 2) Engenharia de controle moderno é uma das principais referências mundiais na área de controle. O livro possui as técnicas clássicas de controle (controlador por atraso/avanço de fase, PID, entre outros), de caracterização de sistemas entre outras coisas.
- 3) Vehicle dynamics and control apresenta as modelagens cinemáticas longitudinal e lateral, e dinâmicas longitudinal e lateral de carros. Além da modelagem, ele também apresenta alguns sistemas de controle de estabilidade utilizados atualmente nos carros.
- 4) O autor fez um trabalho semelhante ao nosso, mas com velocidade baixa constante e um único motor. Além disso, seu controle era feito através do Matlab. Os dados obtidos pelos sensores eram enviados ao PC e a saída do controle era devolvida ao carrinho através de um controle remoto. Contudo, apesar das diferenças, há semelhanças na obtenção de dados e na modelagem servindo, portanto como uma referência valiosa no decorrer do projeto.
- 5) Este livro aborda os principais tópicos relacionados ao processamento de imagens. Nele tem-se desde a aquisição de imagens por diversos tipos de câmera, até visualização de imagens 3D. Sendo assim, pretende-se utilizá-lo para consulta durante todo o processo de aprendizagem da câmera matricial que deverá ser adotada no projeto (CM – 26N/P CMOS).
- 6) Este livro abrange de forma bem específica todos os assuntos tratados neste projeto, desde a locomoção, cinemática e percepção até o planejamento de trajetória.
- 7) A apostila do Laboratório de Controle desenvolve toda a familiarização com um servomecanismo, e sua teoria é aplicável para os motores do Gryphon, sendo de grande utilidade no desenvolvimento do controlador de velocidade.
- 8) Este livro aborda os principais tópicos de cálculo numérico.

20. Referências

- [1] RAJAMANI, Rajesh. **Vehicle dynamics and control**. Londres: Springer, 2005.
- [2] **Understanding autonomous ground vehicles (AGVs):** concepts in robots that can think and function for themselves . Disponível em:
<http://archive.darpa.mil/grandchallenge/docs/darpa_lesson_plans.pdf>. Acesso em: abr. 2012.
- [3] BOHREN, Jonathan; FOOT, Tully; KELLER, Jim; KUSLEYER, Alex; LEE, Daniel; STEWART, Alex; VERNAZA, Paul; DERENICK, Jason; SPLETZER, John; SATTERFIELD, Brian. Little Ben: The Franklin Racing Team's Entry in the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, v.25, n.9, p.598-614, 2008.
- [4] **Facts and figures** - Abu Dhabi and the Yas Marina Circuit . Disponível em:
<<http://www.formula1.com/news/features/2009/10/10143.html>>. Acesso em: 21 maio 2012.
- [5] **MPC5604B/C Microcontroller product brief** . Disponível em:
<http://cache.freescale.com/files/32bit/doc/prod_brief/MPC560XBCPB.pdf>. Acesso em: 21 maio 2012.
- [6] SEDRA, Adel S.; SMITH, Kenneth C. **Microeletrônica**. 5ªed. São Paulo: Pearson Prentice Hall, 2009.
- [7] SIEGWART, Roland; NOURBAKHS, Illah R.; SCARAMUZZA, Davide. **Introduction to autonomous mobile robots**. 2ªed. Londres: MIT, 2011.
- [8] SILVA, Rodney A. B., **Implementação de um robô móvel dotado de um sistema de guiagem baseado em processamento digital de imagens e técnicas avançadas de controle**. 2004. 148p. Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo, 2004.
- [9] PARIS, Sylvain; KORNPORST, Pierre; TUMBLIN, Jack; DURAND, Frédo. Bilateral Filtering: Theory and Applications. *Computer Graphics and Vision*, v. 4, n.1, p.1-73, 2008.
- [10] **Six degrees of freedom** . Disponível em :
<http://en.wikipedia.org/wiki/Six_degrees_of_freedom>. Acesso em: 28 nov. 2012.
- [11] **Informações institucionais:** Bosch destaca segurança no Congresso SAE 2012 Disponível em: <<http://www.bosch.com.br/Imprensa/Releases/Detalhes.aspx?idRelease=10211>>. Acesso em: 27 out. 2012.
- [12] FUENTES, Francisco Ramirez Fuentes; TRUJILLO, Marco; PADILLA, Cuauhtli; MENDOZA, Rodrigo. Using Parallax TSL1401-DB linescan camera module for line detection. **Freescale Semiconductor Application Note**, v. 0, 2011.

[13] KASSAB JR, F.; CRUZ, J.J.; AYA, J. C.; MARQUES, R. P. **Laboratório de controle**. São Paulo: Epusp. 2011. Apostila para a disciplina de graduação do Departamento de Automação e Controle. PTC- 2512 – Laboratório de Controle.

[14] RUGGIERO, Márcia A. Gomes; LOPES, Vera Lúcia da Rocha. **Cálculo numérico: Aspectos Teóricos e Computacionais**. 2ªed. São Paulo: Pearson Makron Books, 1996.

21. ANEXOS

ANEXO I

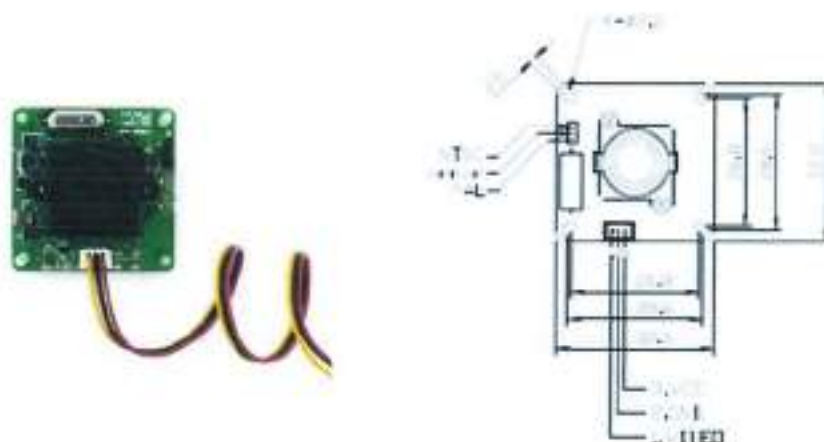
Características da câmera matricial CMOS:

- Entrada de 5 a 15V
- Corrente de 50mA a 12V
- Saída NTSC ou PAL através de chaveamento

Datasheet (o modelo do projeto é o CM-26N):

Model no	SPEC	
	CM-26N	CM-26P
Description		
IMAGE SENSOR	1/4 inch 668(H) x 496(V) MICRON C-MOS IMAGE SENSOR	
Effective pixel	640(H) x 480(V)	
Unit Pixel Size	5.6 μm X 5.6 μm	
Scanning system	525 Line 2:1 INTERLACE	625 Line 2:1 INTERLACE
SYNC system	Internal	
Scanning frequency	Horizontal : 15.734 KHz $\pm$ 1%	Horizontal : 15.625 KHz $\pm$ 1%
	Vertical : 59.94Hz $\pm$ 1%	Vertical : 50Hz $\pm$ 1%
Horizontal resolution	420 TV LINES above (center)	
Video output	1.0 $\pm$ 0.2Vp-p / 75 Ω	
	(Video signal : 0.7Vp-p, Sync. signal : 0.3Vp-p)	
S/N RATIO	More than 45dB (AGC off)	
Dynamic Range	60dB	
Min. Illumination	0.1 lx / F1.6(50% of Video Level, AGC ON)	
Shutter Type	Electronic rolling shutter(ERS)	
Lens angle	70°(Default) or Option	
A G C CONTROL	AUTO	
B.I.C	Built-In	
Dimensions	32(W) x 32(H) x 22(D) $\pm$ 0.3mm	
Operation Temper.	-20°C ~ 60°C	
Power consumption	DC 12V(Tolerance 5V~15V)	
	MAX 50mA (12V)	

Dimensões:



Características da câmera linear:

- Entrada de 3.3 a 5V
- Saída analógica
- Visão 1-D com 128 pixels de resolução
- Campo de visão aproximadamente igual à distância para o objeto
- Não possui foco automático
- Para operação normal devem ser considerados apenas 3 sinais:
 - SI: sinal de saída, começa com *scan/exposure*
 - CLK: sinal de saída, trava o SI e sincroniza a saída dos pixels
 - AO: entrada analógica do sensor



Para maiores informações, vide *datasheet* disponível em:

<http://www.parallax.com/Portals/0/Downloads/docs/prod/acc/TSL1401-DB_manual.pdf>

ANEXO II

Para a escolha do tipo de câmera, foi utilizado o método AHP (*Analytic Hierarchy Process*), mostrado a seguir:

Critérios:

C1: Visão proporcionada

C2: Tempo de processamento

C3: Ajustes (tempo de exposição, foco)

Alternativas:

Câmera de linha

Câmera matricial

Matriz de Prioridades:

Critério	Visão proporcionada	Tempo de processamento	Ajustes
Visão proporcionada	1	1/5	1/3
Tempo de processamento	5	1	4
Ajustes	1/4	3	1

Justificativa das notas:

1) Visão proporcionada x Tempo de processamento

Uma vez que o projeto é de um sistema embarcado, é importante que todos os módulos tenham consistência de tempo entre si. Dado que o microcontrolador gasta muito tempo no processamento da imagem, então a velocidade de processamento se torna mais importante que a visão mais ampla da trajetória, visto que não adiantaria ter uma excelente visão se não for possível trabalhá-la por causa do tempo de processamento.

2) Visão proporcionada x Ajustes

O carrinho, uma vez colocado na pista, deve seguir a trajetória de forma autônoma e, para isso, deve possuir uma boa visão do ambiente. Sendo assim, é interessante que o

ajuste do foco e do contraste seja automático, visto que não é possível alterar esses parâmetros se o carrinho já estiver na pista.

3) Tempo de processamento x Ajustes

Novamente, a questão do tempo de processamento prevalece sobre outro critério, visto que os recursos do microcontrolador são limitados.

Para se conseguir uma classificação a partir da matriz de prioridades, calcula-se o autovalor, inicialmente elevando-se a matriz ao quadrado, o que gera o seguinte resultado:

2,0833	1,4000	1,4667
11,0000	14,0000	9,6667
15,5000	6,0500	13,0833

Somando-se as linhas, para obter o autovalor:

4,9500
34,6667
34,6333
TOTAL: 74,2500

Normalizando a matriz com o total (74,2500):

0,0281	0,0189	0,0198
0,1481	0,1886	0,1301
0,2088	0,0815	0,1762

E, normalizando o autovalor:

0,0667
0,4669
0,4664

Repetindo o processo:

42,4736	31,3900	35,7778
326,7500	269,8833	277,9389
301,6333	185,5541	252,3903

Somando-se as linhas, para obter o autovalor:

109,6414
874,5722
739,5777
TOTAL: 1.723,7913

Normalização da matriz:

0.0246	0.0182	0.0208
0.1896	0.1566	0.1612
0.1750	0.1076	0.1464

Do autovalor:

0,0636
0,5074
0,4290

Computando a diferença entre eles:

0,0031
-0,0405
0,0374

Dado que a diferença já é bem pequena nessa iteração, prossegue-se para o próximo passo usando o último autovalor obtido, que dá as relações entre os critérios:

C1: Visão proporcionada	0,0636
C2: Tempo de processamento	0,5074
C3: Ajustes (tempo de exposição, foco)	0,4290

Com esses pesos obtidos, passa-se para a comparação das alternativas segundo cada critério.

1) Em termos de Visão Proporcionada:

	Câmera de linha	Câmera matricial
Câmera de linha	1	1/3
Câmera matricial	3	1

2) Em termos de Tempo de Processamento:

	Câmera de linha	Câmera matricial
Câmera de linha	1	3
Câmera matricial	1/3	1

3) Em termos de Ajustes:

	Câmera de linha	Câmera matricial
Câmera de linha	1	1/2
Câmera matricial	2	1

Fazendo o mesmo processo que foi feito para a matriz de prioridades, mas agora para cada critério, tem-se:

1) Autovalor em termos de visão proporcionada:

0,25

0,75

2) Autovalor em termos de tempo de processamento:

0,7500

0,2500

3) Autovalor em termos de ajustes:

0,3333

0,6667

Por fim, faz-se uma matriz composta por esses pesos vezes o autovalor dos critérios:

	Visão Proporcionada	Tempo de Processamento	Ajustes
Câmera de linha	0,25	0,7500	0,3333
Câmera matricial	0,75	0,2500	0,6667

Pesos dos critérios	
0,0636	Visão Proporcionada
0,5074	Tempo de Processamento
0,4290	Ajustes

Que resulta em:

Câmera matricial	0,4606
Câmera de linha	0,5394

Com isso, conclui-se que a melhor escolha é a câmera linear, dado que ela obteve o maior valor após a aplicação do método AHP.

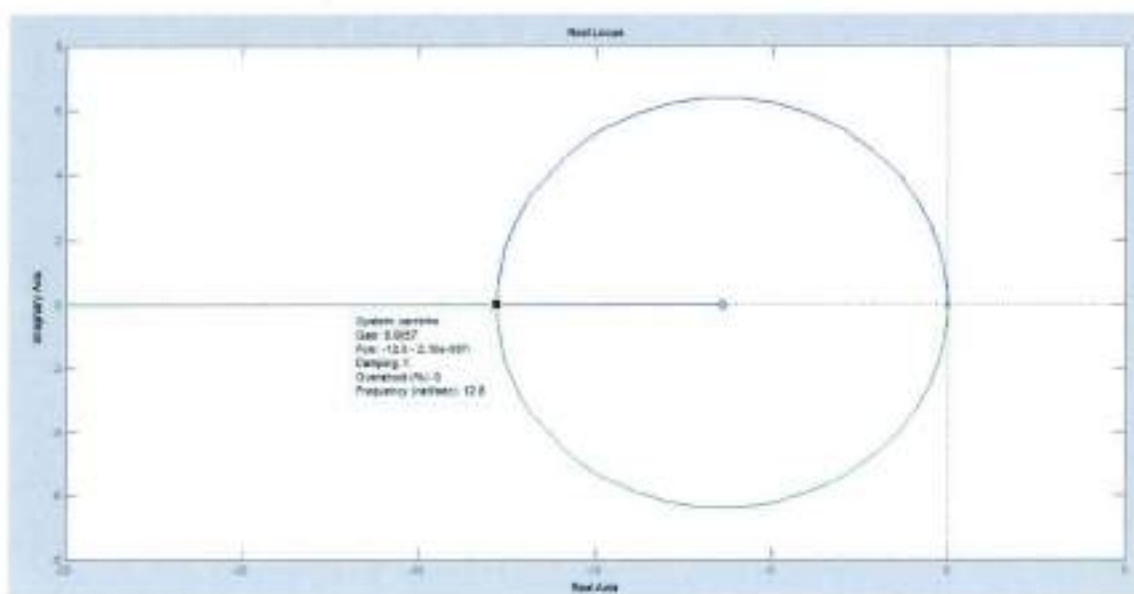
ANEXO III - Dados Experimentais

1) Modelagem do Controle de Posição

Parâmetros do Modelo:

- ➔ $DC = 39 \text{ cm}$
- ➔ $V_0 = 2,4 \text{ m/s}$

a) Lugar Geométrico das Raízes do Modelo

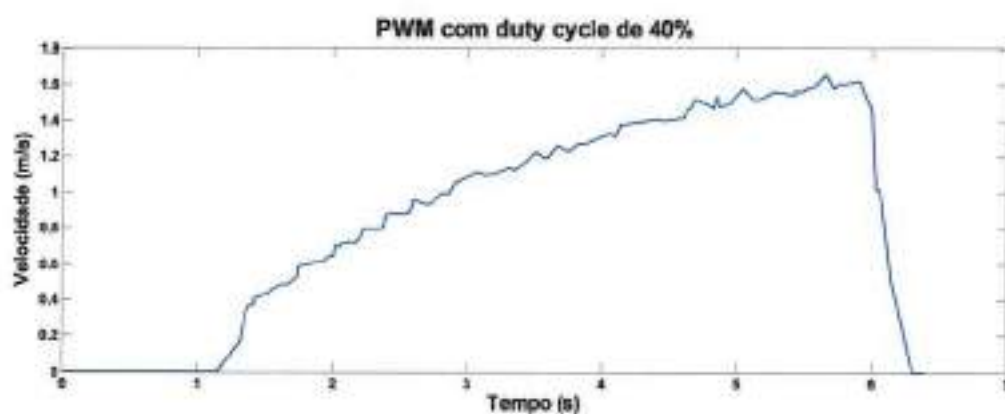


2) Dados do Controle de Velocidade

a) Velocidade x Tempo (Potência de 20%)



b) Velocidade x Tempo (Potência de 40%)



c) Velocidade x Tempo (Potência de 60%)



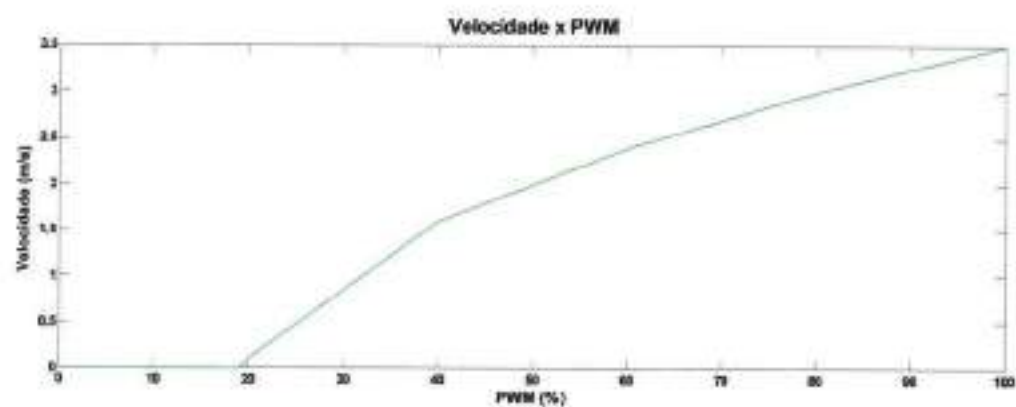
d) Velocidade x Tempo (Potência de 80%)



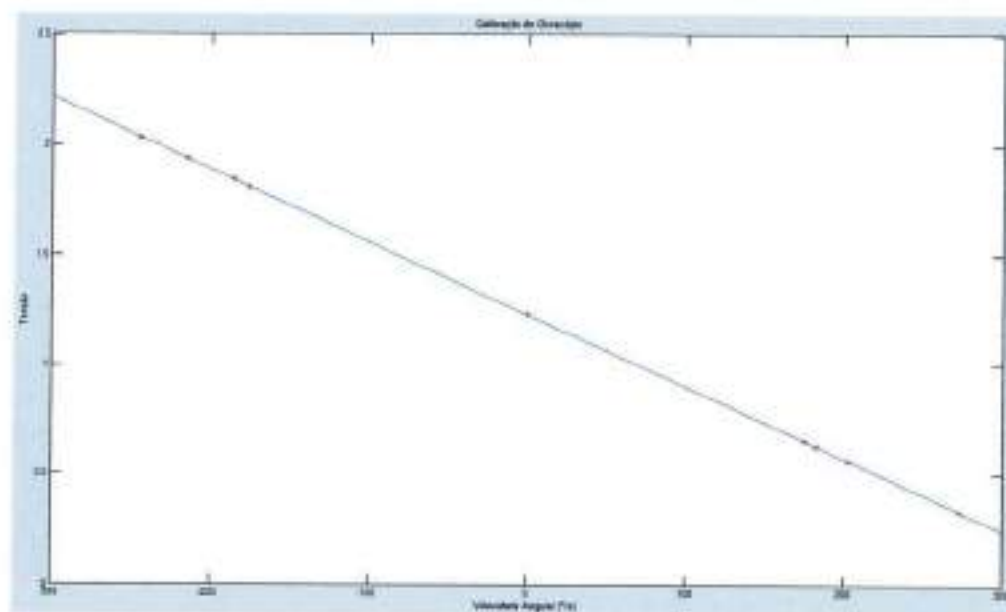
e) Velocidade x Tempo (Potência de 100%)



f) PWM x Velocidade de Regime Permanente



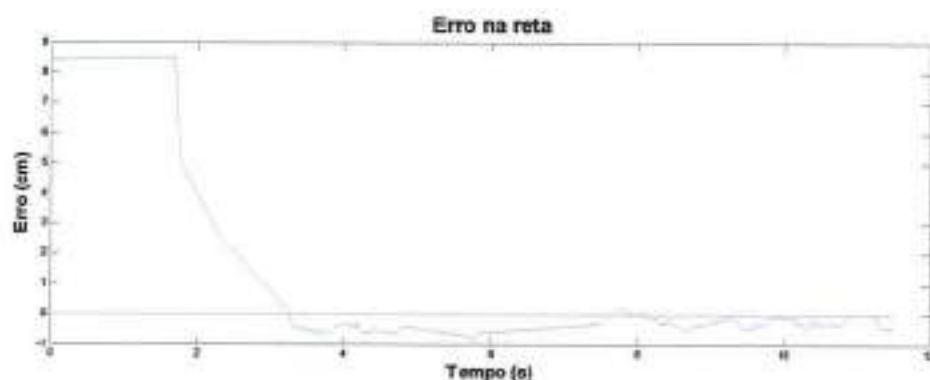
3) Dados da Calibração do Giroscópio



ANEXO IV – Resultados

1) Resultado do Controle de Posição

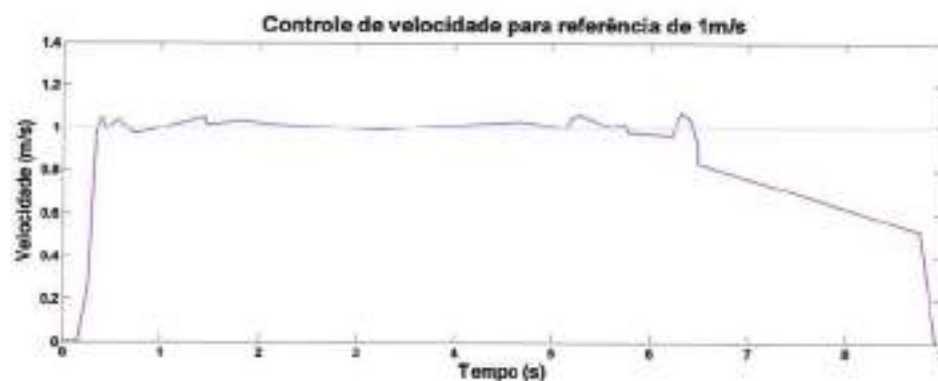
- Ajustes feitos para Velocidade de 2,5 m/s e Distância da Visão da Câmera de 39 cm
- *Duty cycle* do teste de 40% (velocidade final de 1,6 m/s)



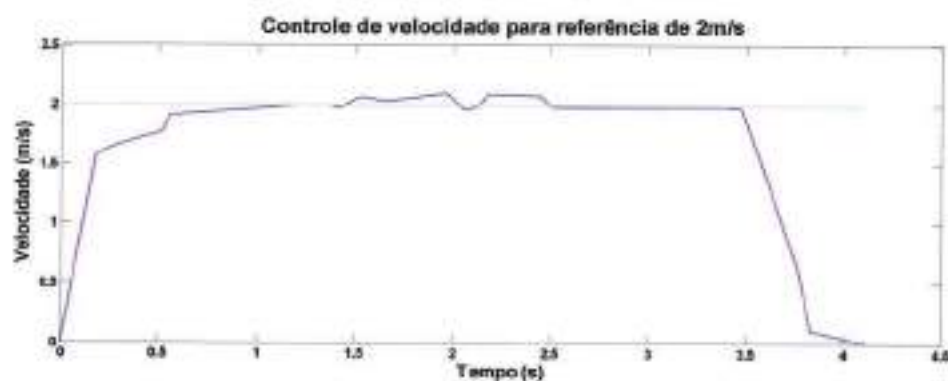
2) Resultado do Controle de Velocidade

- Ajustes feitos com base na resposta do degrau de PWM com 60% de *duty cycle*.

a) Velocidade de 1,0 m/s



b) Velocidade de 2,0 m/s

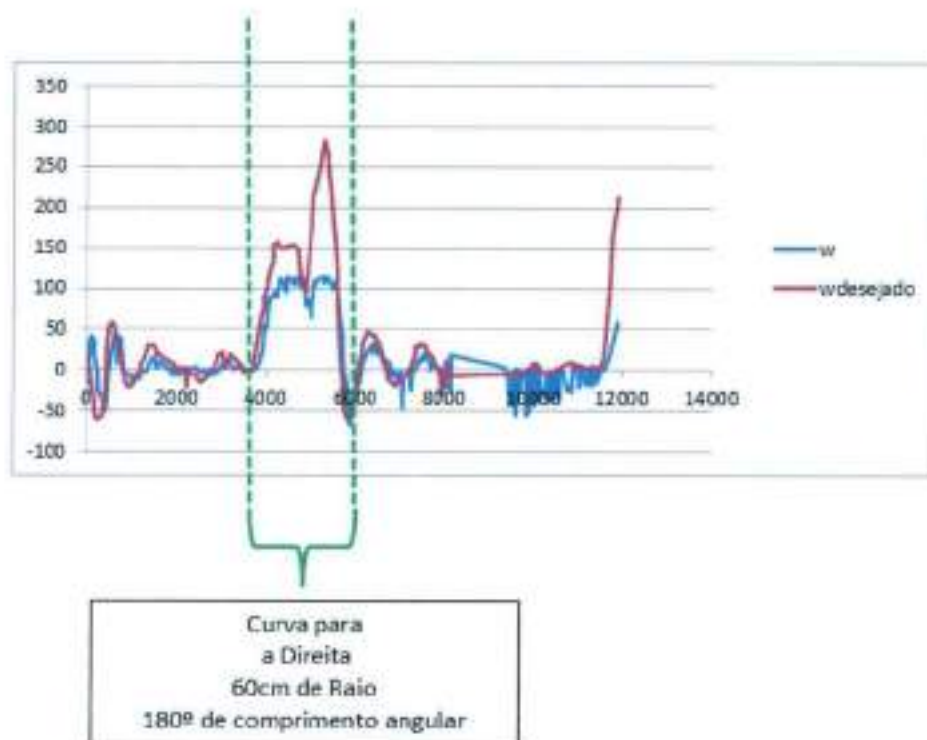


c) Velocidade de 2,5 m/s



3) Resultado do Controle de Estabilidade

- Velocidade de 2,3 m/s em uma reta seguida de uma curva de 180° e 60 centímetros de raio. Após a curva o carrinho segue para uma nova reta.



ANEXO V - Método de Newton-Raphson por secantes para o Cálculo de Raiz quadrada

```
%Metodo de Newton com secantes

i=0;
k=0;

chuteAnt = Rquad/2;

dfchuteAnt = 2*chuteAnt;

fchuteAnt = chute*chute - Rquad;

chute = chuteAnt - fchuteAnt/dfchuteAnt;

difChutes = chute-chuteAnt;

if difChutes < 0
    difChutes = (-1)*difChutes;
end

erro = chute*chute - Rquad;

if erro < 0
    erro = (-1)*erro;
    sinal = 1;
else
    sinal = 0;
end

while (erro > precisao || difChutes > precisao)
    i=i+1;
    k=k+1;

    chuteAnt = chute;
    k=k+1;

    dfchuteAnt = 2*chuteAnt;
    k=k+1;

    if sinal == 0
        fchuteAnt = erro;
        k=k+1;
    else
        fchuteAnt = (-1)*erro;
        k=k+1;
    end
end
```

```

chute = chuteAnt - fchuteAnt/dfchuteAnt;
k=k+1;

difChutes = chute-chuteAnt;
k=k+1;

if difChutes < 0
    difChutes = (-1)*difChutes;
    k=k+1;
end

erro = chute*chute - Rquad;
k=k+1;

if erro < 0
    erro = (-1)*erro;
    k=k+1;
    sinal = 1;
else
    sinal = 0;
end

end

Rnewton = chute

```