

UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
PROGRAMA DE PÓS-GRADUAÇÃO PROFISSIONAL EM ENGENHARIA ELÉTRICA
– PPGPEE

GUILHERME DO NASCIMENTO

CONTROLE PREDITIVO ESTOCÁSTICO APLICADO A ASSISTENTE DE DIREÇÃO
EM CARRO EM ESCALA REDUZIDA

JOINVILLE

2022

GUILHERME DO NASCIMENTO

**CONTROLE PREDITIVO ESTOCÁSTICO APLICADO A ASSISTENTE DE DIREÇÃO
EM CARRO EM ESCALA REDUZIDA**

Dissertação apresentada ao Programa de Pós-Graduação Profissional em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Dra. Mariana Santos Matos Cavalca
Coorientador: Dr. Celso José Faria de Araújo

JOINVILLE

2022

**Ficha catalográfica elaborada pelo programa de geração automática da
Biblioteca Setorial do CCT/UEDESC,
com os dados fornecidos pelo(a) autor(a)**

Nascimento, Guilherme do

Controle preditivo estocástico aplicado a assistente de direção
em carro em escala reduzida / Guilherme do Nascimento. -- 2022.
58 p.

Orientadora: Mariana Santos Matos Cavalca

Coorientador: Celso José Faria de Araújo

Dissertação (mestrado) -- Universidade do Estado de Santa
Catarina, Centro de Ciências Tecnológicas, Programa de
Pós-Graduação , Joinville, 2022.

1. Assistente de direção. 2. Smc. 3. Processamento de imagem.
4. Sistemas embarcados. 5. Ros. I. Cavalca, Mariana Santos Matos.
II. Araújo, Celso José Faria de. III. Universidade do Estado de Santa
Catarina, Centro de Ciências Tecnológicas, Programa de
Pós-Graduação . IV. Título.

GUILHERME DO NASCIMENTO

**CONTROLE PREDITIVO ESTOCÁSTICO APLICADO A ASSISTENTE DE DIREÇÃO
EM CARRO EM ESCALA REDUZIDA**

Dissertação apresentada ao Programa de Pós-Graduação Profissional em Engenharia Elétrica do Centro de Ciências Tecnológicas da Universidade do Estado de Santa Catarina, como requisito parcial para a obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Dra. Mariana Santos Matos Cavalca

Coorientador: Dr. Celso José Faria de Araújo

BANCA EXAMINADORA:

Profa. Dra. Mariana Santos Matos Cavalca
CCT/UEDESC
(Presidente/orientadora)

Membros:

Prof. Dr. José de Oliveira
CCT/UEDESC

Prof. Dr. Luiz Antonio Celiberto Junior
UFABC

Prof. Dr. Celso José Faria de Araújo
CCT/UEDESC
(Suplente/coorientador)

Joinville, 21 de julho de 2022

À minha família, pois nela encontro a força
diária para enfrentar quaisquer obstáculos!

AGRADECIMENTOS

A minha esposa Fabrine, pelo apoio diário a esta jornada que chamamos de vida.

Aos meus pais Carmem e Manoel, e minha irmã Ketlin, vocês são a estrutura que me proporciona voar cada vez mais longe.

Ao meu grande amigo Áureo, que mesmo de longe sempre esteve disponível para discutir qualquer tópico técnico ou pessoal.

Aos meus orientadores Mariana dos Santos Matos Cavalca e Celso José Faria de Araújo pela orientação e suporte.

Por fim, a Universidade do Estado de Santa Catarina pela educação pública e de qualidade.

"Se podes?", disse Jesus. "Tudo é possível
àquele que crê." (Marcos 9:23, Bíblia Sagrada)

RESUMO

Sistemas Avançados de Assistência ao Motorista - ADAS, são sistemas inteligentes embarcados em veículos com objetivo de promover a segurança e conforto, evitando acidentes através do uso de algoritmos conectados a dispositivos sensores, como radares de média e longa distância, câmeras, entre outros. Este trabalho propõe uma solução de assistente de direção através de um Controlador Preditivo Estocástico Baseado em Modelo - SMPC com restrições de segurança, aplicado a um veículo em escala reduzida. Para tal desafio utiliza-se o conceito de *chance-constraint* para adicionar conservadorismo ao controlador, fazendo com que o veículo faça uma trajetória mais segura. Para realização de testes práticos inspirado na plataforma *Donkey-car*, desenvolveu-se um veículo em escala reduzida 1/16, que conta com uma plataforma em ROS embarcada a um Raspberry Pi, onde constam o processamento de imagem baseado em Python e OpenCV e o controlador preditivo embarcado. Por fim, através de simulações em Matlab e resultados práticos utilizando o protótipo criado, mostra-se que a técnica é efetiva na resolução do problema de assistente de direção.

Palavras-chave: Assistente de direção. SMPC. Processamento de imagem. Sistemas embarcados. ROS.

ABSTRACT

Advanced Driver Assistance Systems (ADAS) are embedded systems for intelligent vehicles, with the goal of promoting safety and comfort, avoiding accidents through the use of algorithms connected to sensor devices, such as medium and long distance radars, cameras, among others. This paper proposes a solution for driving assistant through a Stochastic Model Predictive Controller - SMPC, with safety restrictions applied to a small scale vehicle. For this challenge, the concept of chance-constraint is used to add conservatism to the controller, causing the vehicle to follow a more conservative trajectory. To carry out practical tests, inspired by the Donkey-car platform, it is developed a 1/16 scale vehicle that has a ROS platform embedded in a Raspberry Pi, with image processing based on Python and OpenCV and an embedded predictive controller. Finally, through Matlab simulations and practical results using the created prototype, it is shown that the technique is effective in solving the driving assistant problem.

Keywords: Driving assistant. SMPC. Image processing. Embedded systems. ROS.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de blocos esquemático de um sistema ADAS	17
Figura 2 – Modelo cinemático	23
Figura 3 – Modelo dinâmico	24
Figura 4 – Exemplo de <i>Donkey Car</i>	27
Figura 5 – <i>SainSmart Wide Angle Fish-Eye Camera</i>	27
Figura 6 – <i>Raspberry Pi 4 Model B</i>	28
Figura 7 – <i>SunFounder PCA9685 16 Channel 12 Bit PWM</i>	29
Figura 8 – <i>Exceed Racing Desert Short Course Truck 1/16 Scale</i>	29
Figura 9 – Peças impressas 3D	30
Figura 10 – Bateria de controle	31
Figura 11 – Bateria de atuação	31
Figura 12 – ROS	32
Figura 13 – Distorção radial	33
Figura 14 – Resultado de calibração da câmera com tabuleiro	33
Figura 15 – Resultado de calibração da câmera com superfície	34
Figura 16 – Coordenadas de fonte e destino	35
Figura 17 – Transformação de perspectiva	35
Figura 18 – Janelas deslizantes	36
Figura 19 – Ensaio de conversão metros por pixel	37
Figura 20 – Modelagem do carro	38
Figura 21 – Resultado de simulação - ensaio com condutor - sinal de saída	43
Figura 22 – Resultado de simulação - ensaio com condutor - sinal de controle	44
Figura 23 – Resultado de simulação - ensaio sem condutor - sinal de saída	45
Figura 24 – Resultado de simulação - ensaio sem condutor - sinal de Controle	45
Figura 25 – Estrutura ROS simplificada	46
Figura 26 – Custo computacional do processamento de imagem	47
Figura 27 – Filtro HLS dinâmico - gráfico	48
Figura 28 – Filtro HLS dinâmico - fotos	48
Figura 29 – Circuito de teste prático	49
Figura 30 – Carro em miniatura desenvolvido	50
Figura 31 – Ensaio demonstrativo	50
Figura 32 – Ensaio demonstrativo - foto 1	51
Figura 33 – Ensaio demonstrativo - foto 2	51
Figura 34 – Ensaio demonstrativo - foto 3	52
Figura 35 – Ensaio demonstrativo - foto 4	52
Figura 36 – Ensaio demonstrativo - foto 5	53
Figura 37 – Resultado do ensaio prático - sinal da trajetória	54

Figura 38 – Resultado do ensaio prático - sinal de controle	54
---	----

LISTA DE TABELAS

Tabela 1 – Dimensões.	29
Tabela 2 – Parâmetros do sistema e do controlador.	42

LISTA DE ABREVIATURAS E SIGLAS

LQR	<i>Linear Quadratic Regulator</i>
MIMO	<i>Multiple Input Multiple Output</i>
MPC	<i>Model Predictive Control</i>
PWM	<i>Pulse Width Modulation</i>
RC	<i>Radio-Controlled</i>
RMPC	<i>Robust Model Predictive Control</i>
ROS	<i>Robot Operating System</i>
SISO	<i>Single Input Single Output</i>
SMPC	<i>Stochastic Model Predictive Control</i>

LISTA DE SÍMBOLOS

x	Vetor de estados do sistema
u	Vetor de entradas do sistema
y	Vetor de saídas do sistema
A	Matriz de dinâmica do sistema em espaço de estados
B	Matriz de entradas do sistema em espaço de estados
C	Matriz de saídas do espaço de estados
D	Matriz de transmissão do espaço de estados
J_N	Função Custo
Q	Matriz de ponderação dos estados
R	Matriz de ponderação das entradas
N	Horizonte de predição
w	Sinal estocástico de distribuição normal
z	Componente determinística do vetor de estados
e	Componente estocástica do vetor de estados
K	Ganho calculado como LQR para a função afim de controle
c	Sinal de controle estocástico, [rad]
w	Sinal estocástico de distribuição normal
Φ	Dinâmica do sistema em malha fechada
e_y	Erro da posição lateral com relação a pista, [m]
\dot{e}_y	Derivada do erro da posição lateral com relação a pista
e_ψ	Erro angular com relação a pista, [rad]
\dot{e}_ψ	Derivada do erro angular com relação a pista
$C_{\alpha f}$	Coefficiente de deslizamento das rodas frontais
$C_{\alpha r}$	Coefficiente de deslizamento das rodas dianteiras
m	Massa total do veículo, [kg]
V_x	Velocidade longitudinal, [m/s]
l_f	Distância entre o centro de massa e eixo frontal, [m]
l_r	Distância entre o centro de massa e eixo traseiro, [m]
I_z	Momento de inércia com relação ao eixo Z, [kg m ²]
δ	Ângulo de direção das rodas, [rad]

$\dot{\Psi}_{des}$	Taxa de variação angular desejada com relação a pista
ξ	Vetor de estados do sistema
$\dot{\xi}$	Derivada do vetor de estados do sistema
δ_C	Ângulo de direção gerado pelo controlador, [rad]
δ_D	Ângulo de direção gerado pelo condutor, [rad]
F_D	Vetor de ganhos do modelo do condutor
K_y	Ganho do modelo do condutor com relação ao <i>offset</i>
K_ψ	Ganho do modelo do condutor com relação ao ângulo
A_{dm}	Matriz de dinâmica do sistema condutor mais veículo

SUMÁRIO

1	INTRODUÇÃO	17
1.1	CONTRIBUIÇÕES	18
1.2	ORGANIZAÇÃO DO TRABALHO	19
2	REVISÃO DA LITERATURA	20
2.1	CONTROLE PREDITIVO	20
2.1.1	Controle preditivo baseado em modelo	20
2.1.2	Controle preditivo estocástico	21
2.2	MODELAGEM DA DINÂMICA LATERAL DE VEÍCULOS	23
2.2.1	Modelagem cinemática	23
2.2.2	Modelagem dinâmica	24
2.2.3	Modelagem dinâmica em termos de erros com relação a pista	25
3	DESENVOLVIMENTO	26
3.1	DESCRIÇÃO DO <i>HARDWARE</i> E <i>SOFTWARE</i>	26
3.1.1	Câmera	26
3.1.2	<i>Raspberry Pi 4 Model B</i>	28
3.1.3	<i>PWM Servo Driver</i>	28
3.1.4	<i>RC Car</i>	28
3.1.5	Peças impressas 3D	30
3.1.6	Baterias	30
3.1.7	<i>Robot Operating System – ROS</i>	30
3.2	PROCESSAMENTO DE IMAGEM PARA DETECÇÃO DE PISTA	32
3.2.1	Correção de distorção	32
3.2.2	Transformação de perspectiva	34
3.2.3	Filtro HLS dinâmico	34
3.2.4	Técnica das janelas deslizantes	36
3.2.5	Conversão pixels por metro e levantamento dos polinômios descritivos	37
3.3	MODELO DE ASSISTENTE DE DIREÇÃO ESTOCÁSTICO	37
3.3.1	Modelagem	37
3.3.2	Função custo	40
3.3.3	Restrições	40
4	RESULTADOS	42
4.1	SIMULAÇÃO	42
4.2	PRÁTICO	46
4.2.1	Estrutura de software	46
4.2.2	Custo computacional e tempo de amostragem	46

4.2.3	Aplicação do Filtro HLS Dinâmico	47
4.2.4	Resultado Prático	49
5	CONCLUSÕES E TRABALHOS FUTUROS	55
	REFERÊNCIAS	56
	ANEXO A – ESTRUTURA ROS COMPLETA	58

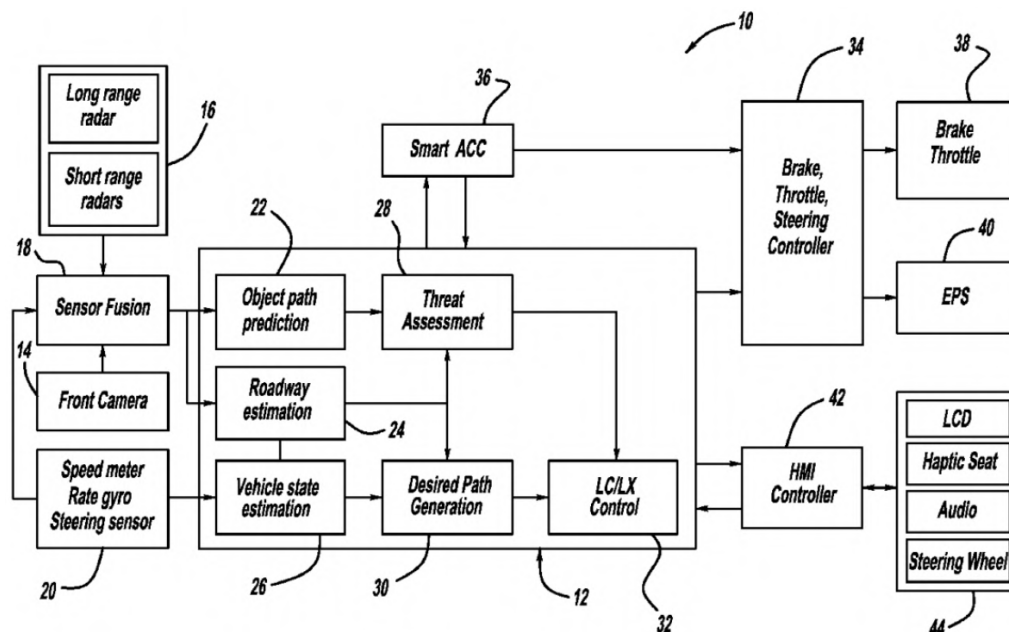
1 INTRODUÇÃO

Considerando um contexto global cada vez mais competitivo e tecnológico (GVR, 2019), a área de controle e automação encontra seu valor não apenas na otimização da manufatura, para que projetos se tornem mais rentáveis e produtivos, mas também nas tarefas diárias que podem ser executadas com mais eficiência e segurança, através de sistemas especializados em tarefas específicas (VTPI, 2022). Com este foco, aplicações de controle no ramo automotivo estão sendo desenvolvidas e aplicadas constantemente no mercado, como por exemplo, controle de velocidade “*Cruise control*”, estacionamento automático “*Auto parking*”, assistentes de direção, entre outros (LEFÈVRE; CARVALHO; BORRELLI, 2016)(JHANG; LIAN, 2020a)(LAURENSE; GERDES, 2022).

Os sistemas avançados de assistência ao motorista, do Inglês ADAS (*Advanced Driver Assistance Systems*), são sistemas inteligentes embarcados no veículo com diversas finalidades, dentre elas promover a segurança e conforto evitando acidentes através do uso de algoritmos conectados a diversos sensores, como radares de média e longa distância, câmeras, entre outros.

Diversos esforços estão sendo aplicados na indústria e na academia com o objetivo de conquistar a direção autônoma de veículos terrestres, mas para se chegar a este resultado existem múltiplas aplicações que devem ser executadas simultaneamente para garantir a segurança (LEE, 2012). Como pode ser visto na Figura 1.

Figura 1 – Diagrama de blocos esquemático de um sistema ADAS



Fonte: (LEE, Jin-woo, 2012)

Desta maneira neste trabalho pretende-se focar na parte inferior esquerda da imagem, onde através de um sensoramento baseado em câmera frontal e um sistema de controle preditivo, pretende-se seguir uma rota ótima respeitando restrições de segurança, levando em conta a

predição dos estados do sistema.

Para tais aplicações plataformas de simulação já trazem um ótimo desempenho para testes com dados reais, afim de diminuir os custos e a aumentar a segurança na implementação das soluções (MathWorks, 2020). Além das simulações encontram-se artigos e patentes com diversas topologias de aplicação de controle em diferentes cenários (LEE, 2012)(NILSSON, 2018).

Para tal aplicação de controle de sistemas uma técnica largamente utilizada é a estratégia de Controle Preditivo Baseado em Modelos (MPC), que é baseada no cálculo das ações de controle em tempo real, de modo a otimizar o comportamento futuro do processo. Por isso, tal comportamento é predito utilizando um modelo matemático representativo do processo. Dessa forma, o controlador preditivo é composto basicamente de um modelo matemático do processo e um otimizador, no qual, para realizar esta otimização faz-se necessária a definição de uma função custo, que geralmente leva em conta erros de rastreamento e esforços de controle, sendo possível ponderá-las da maneira mais eficiente para a aplicação (CAMACHO; ALBA, 2007).

Porém, em sistemas práticos é comum a presença de incertezas, como ruídos, perturbações externas e erros de modelagem. Nesse contexto, controladores estocásticos demonstram potencial para lidar com sistemas incertos, visto que na natureza é comum as incertezas serem dadas por distribuições probabilísticas. Dessa forma, Controladores Preditivos Estocásticos, ou *Stochastic Model Predictive Control* (SMPC), oferecem uma forma de considerar descrições probabilísticas no projeto do controlador (MESBAH, 2016). Portanto, esta dissertação teve como objetivo geral desenvolver um sistema prático de assistência de direção para veículos, capaz de contornar restrições de segurança aplicando um controlador preditivo estocástico a um carro em escala reduzida, para isso conta com os seguintes objetivos específicos:

- Estudo e aplicação da estratégia de um controlador preditivo estocástico embarcado;
- Modelagem da dinâmica lateral de veículos;
- Desenvolvimento e prototipagem de um veículo em escala reduzida;
- Estudo e aplicação embarcada de processamento de imagem para detecção de pistas.

1.1 CONTRIBUIÇÕES

Este trabalho propõe uma solução para um assistente de direção, implementado via Controlador Preditivo Estocástico Baseado em Modelo. A estrutura obtida permite o controle de trajetória respeitando restrições de segurança, onde através da adição de um sinal estocástico no sinal de direção do veículo, juntamente com a aplicação do conceito de *chance-constraint*, varia-se o conservadorismo do controlador fazendo com que uma margem de segurança seja adicionada ao trajeto. Os resultados simulados e práticos demonstram que a técnica aplicada é capaz de resolver o problema de maneira efetiva.

Ainda com relação as contribuições, cria-se um carro em escala reduzida para testes de controle, onde este protótipo conta com sistema embarcado de processamento de imagem para detecção de pista, controlador preditivo embarcado e sistema ROS. O sistema desenvolvido se mostra eficaz no teste de controladores, tendo a possibilidade de implementação de diferentes arquiteturas e aplicações de malhas de controle.

1.2 ORGANIZAÇÃO DO TRABALHO

O trabalho está dividido da seguinte maneira: no Capítulo 2 é realizada a revisão da literatura através do estudo de controladores preditivos e modelagem do sistema. No Capítulo 3 está descrito o desenvolvimento do trabalho através da montagem do protótipo de veículo em pequena escala, processamento de imagem para detecção de pista e aplicação do controle preditivo estocástico ao assistente de direção. No Capítulo 4 estão descritos os resultados obtidos por meio de simulações e ensaios práticos. Por fim, as conclusões e perspectivas futuras deste trabalho estão discutidas no Capítulo 5.

2 REVISÃO DA LITERATURA

Neste capítulo estão descritos os fundamentos teóricos utilizados para produção deste trabalho, na Seção 2.1 descreve-se duas técnicas de controle preditivo baseado em modelos e na Seção 2.2 exploram-se três métodos de modelagem para o sistema utilizado.

2.1 CONTROLE PREDITIVO

A estratégia empregada no controle preditivo é baseada em dois principais componentes, um modelo matemático descritivo do sistema, onde através do estado atual é possível calcular os estados futuros e um otimizador, que é responsável por otimizar os estados, saídas e sinais de controle do sistema, conforme os critérios definidos pelo usuário através de uma função custo e opcionalmente pela adição de restrições (CAMACHO; ALBA, 2007).

2.1.1 Controle preditivo baseado em modelo

Para a modelagem matemática de sistemas SISO e MIMO é possível utilizar um modelo discreto em espaço de estados conforme a Equação (1).

$$\begin{aligned} x_{k+1} &= A_d x_k + B_d u, \\ y_k &= C_d x_k. \end{aligned} \quad (1)$$

onde x_{k+1} representa o vetor de n estados do sistema no instante $k+1$ calculado no instante k , y_k sendo o vetor com as p saídas, u_k o vetor com as m ações de controle, a matriz A_d representa a dinâmica do sistema discretizado, B_d a matriz de entrada e C_d a matriz de saída. Tal que $A_d \in \mathbb{R}^{n \times n}$, $B_d \in \mathbb{R}^{n \times m}$ e $C_d \in \mathbb{R}^{p \times n}$.

Com posse do modelo descritivo do sistema, se faz necessário definir uma função custo para que o otimizador possa calcular as ações de controle, para tal define-se a Equação (2), que leva em conta a minimização dos estados e dos sinais de controle (CAMACHO; ALBA, 2007).

$$J_N(x, u) = \sum_{i=0}^{N-1} \left[x'_{k+i} Q x_{k+i} + u'_{k+i} R u_{k+i} \right], \quad (2)$$

em que Q e R são matrizes de ponderação que permitem o ajuste em cada um dos estados e ações de controle respectivamente, onde $Q = Q' \geq 0$ e $R = R' \geq 0$ para garantir a minimização.

Na prática, todo e qualquer processo possui restrições físicas e/ou operacionais, para isto é possível implementá-las diretamente no otimizador com objetivo de manter a integridade do sistema (CAMACHO; ALBA, 2007).

Definindo as restrições nas ações de controle, conforme:

$$F u \leq W, \quad (3)$$

Na Equação (3), F descreve como as variáveis de otimização u afetam determinada característica do sistema, sendo que esta possui W como seus valores máximos.

Pode-se definir também restrições nos estados do sistema, conforme:

$$g'x_k \leq h. \quad (4)$$

onde de forma análoga, na Equação (4), g' descreve como as variáveis de otimização x_k afetam determinada característica do sistema, sendo que esta possui h como seus valores máximos.

Por fim, com posse do modelo do sistema, função custo e restrições é formulado um problema de otimização de forma a minimizar a função custo da Equação (2), respeitando as restrições das Equações (3) e (4):

$$\begin{aligned} \min_u \quad & J_N(x_k, u) \\ \text{sujeito a} \quad & Fu \leq W, \\ & g'x_k \leq h. \end{aligned} \quad (5)$$

Através desta otimização é possível achar uma ação de controle ótima, em que como resultado é obtido um vetor $U = \{u_1, \dots, u_N\}$ com a sequência de ações de controle ao longo do horizonte N , em que apenas a primeira ação de controle u_k é aplicada, e no próximo ciclo é realizado novamente o cálculo em todo o horizonte, estratégia denominada de horizonte retrocedente.

2.1.2 Controle preditivo estocástico

Mesmo que o controlador MPC ofereça certa robustez a incertezas e distúrbios devido a sua estratégia de horizonte retrocedente, outras técnicas como controladores estocásticos surgem para que estas incertezas sejam levadas em conta no projeto do controlador (MESBAH, 2016). Uma maneira clássica de implementar o controle estocástico é considerar um termo extra na representação do sistema dado pela Equação (1), onde o vetor $w_k \in \mathbb{R}^{n_w}$ é responsável pela adição de um sinal estocástico de distribuição conhecida, resultando no sistema descrito pela Equação (6):

$$\begin{aligned} x_{k+1} &= Ax_k + Bu_k + Dw_k, \\ y_k &= Cx_k. \end{aligned} \quad (6)$$

tal que, $D \in \mathbb{R}^{n \times m}$.

Em abordagens de Controladores Preditivos Robustos (RMPC) é possível considerar o sinal w_k como o pior cenário possível, porém com esta consideração leva-se o controlador a um estado muito restritivo e conservador, tendo em vista que o pior caso possível pode ter uma probabilidade muito pequena de acontecer. Com base neste cenário o controlador SMPC leva em conta este sinal estocástico ilimitado no modelo do sistema, criando-se o conceito de *chance-constraint*, que é definido como a probabilidade aceitável de violação das restrições do sistema.

Em sistemas estocásticos se torna comum a decomposição dos estados x_k em duas componentes, a primeira z_k responsável por descrever o comportamento determinístico, e a

segunda componente e_k que descreve o comportamento estocástico, possuindo valor esperado $\mathbb{E}[e_k] = 0$. De forma a introduzir maior robustez, a ação de controle u_k é definida como uma função afim, com um ganho K constante ao longo do horizonte calculado como um ganho de um LQR, e uma componente c_k . Dessa forma, obtém-se:

$$\begin{aligned} x_k &= z_k + e_k, \\ u_k &= Kx_k + c_k, \\ z_{k+1} &= \Phi z_k + c_k, \\ e_{k+1} &= \Phi e_k + Dw_k, \end{aligned} \tag{7}$$

em que $\Phi = A + BK$ descreve a dinâmica do sistema em malha fechada, tal que $\Phi \in \mathbb{R}^{n \times m}$.

Com relação a função custo, uma das abordagens mais comuns é considerar o valor esperado da função custo já definida anteriormente pela Equação (2), resultando na Equação (8):

$$J_N^{SMP} (x, U) = \mathbb{E}[J_N] = \mathbb{E} \left[\sum_{i=0}^{N-1} \left[x'_{k+i} Q x_{k+i} + u'_{k+i} R u_{k+i} \right] \right], \tag{8}$$

onde os termos $Q \in \mathbb{R}^{nN \times nN}$ e $R \in \mathbb{R}^{pN \times pN}$ sendo matrizes simétricas, semi-definidas positivas, que ponderam respectivamente os termos das saídas e estados do sistema e N o horizonte de predição.

Devido a adição do comportamento estocástico no modelo do sistema, se torna necessária a adição do termo λ_i na equação de restrições com a finalidade de compensar a influência do vetor w_k , resultando na Equação (9).

$$g'x_k = g'(z_k + e_k) \leq h - \lambda_i. \tag{9}$$

Em controladores estocásticos, é comum redefinir as restrições nos estados como uma probabilidade aceitável que a restrição original seja violada (FARINA; GIULIONI; SCATTOLINI, 2016), técnica denominada *chance-constraint*:

$$\mathcal{P}\{g'(z_k + e_k) \leq h\} \geq 1 - p, \tag{10}$$

em que $\mathcal{P}\{X\}$ indica a probabilidade do evento X ocorrer, e p indica a máxima probabilidade aceitável para violação da restrição.

Considerando w_k como um sinal ilimitado com distribuição Gaussiana, valor esperado $\mathbb{E}[w_k] = 0$ e desvio padrão W , λ_i pode ser calculado exatamente como:

$$\lambda_i(1 - p) = \sqrt{g'E_i \mathbf{W} E_i' g} \mathcal{N}^{-1}(1 - p), \tag{11}$$

em que $\mathbf{W} = \text{diag}(W, \dots, W)$ e \mathcal{N} é a função de probabilidade acumulada para uma variável Gaussiana com média zero e variância unitária, e E_i mapeia a influência de $\{w_{k+1}, w_{k+2}, \dots, w_{k+i}\}$ em e_{k+i} .

A vantagem dessa abordagem, é o custo computacional, visto que os valores para λ_i podem ser computados offline e apenas adicionados às restrições originais.

2.2 MODELAGEM DA DINÂMICA LATERAL DE VEÍCULOS

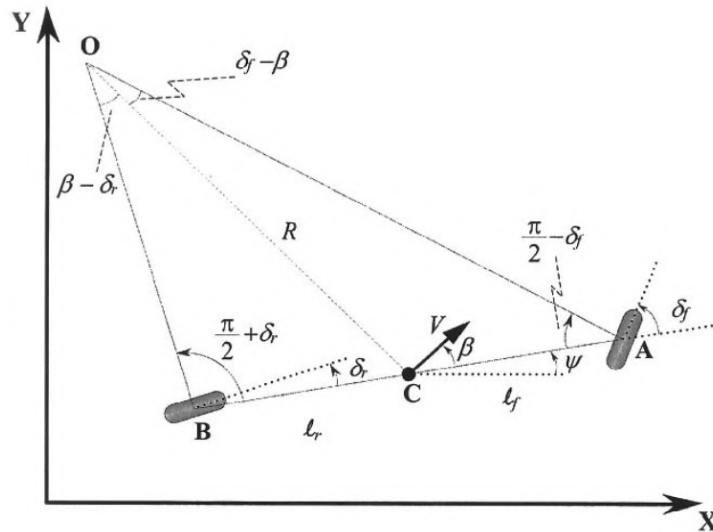
Com objetivo de controlar um sistema de direção veicular se faz necessário o levantamento do modelo matemático descritivo do sistema, e para este levantamento deve-se analisar cada aplicação, resultando na utilização de um modelo mais simplificado ou mais complexo, dependendo de elementos como região de operação e limitação computacional.

2.2.1 Modelagem cinemática

Em certas aplicações simplificadas, como veículos em relativa baixa velocidade e com baixo deslizamento com o solo, pode-se utilizar modelos cinemáticos simples onde levam em conta apenas a geometria do sistema, desprezando as forças que interferem no movimento dinâmico (RAJESH, 2006).

Uma simplificação comum dentre os trabalhos da área, como mostrado nos trabalhos de (YAMAUCHI; SAITO; ONO, 2019) e (JHANG; LIAN, 2020b), é considerar que as duas rodas frontais e traseiras podem ser representadas como apenas uma roda frontal e traseira deslocada para o centro do veículo, gerando o modelo denominado como modelo de bicicleta, conforme mostrado na Figura 2.

Figura 2 – Modelo cinemático



Fonte: (RAJESH, 2006)

Nesta representação do sistema considera-se que o veículo tem movimentos apenas no plano 2D, portanto três variáveis são suficientes para descrição de sua posição e orientação, sendo (X, Y) as coordenadas de posição e Ψ a orientação angular, para simplificação de notação

a componente (t) será ocultada, conforme:

$$\begin{aligned}\dot{X} &= V \cos(\Psi + \beta), \\ \dot{Y} &= V \sin(\Psi + \beta), \\ \dot{\Psi} &= \frac{V \cos(\beta)}{l_f + l_r} (\tan(\delta_f) - \tan(\delta_r))\end{aligned}\quad (12)$$

onde, V , δ_f e δ_r são sinais de entrada sendo respectivamente a velocidade do veículo, ângulo de direção da roda frontal e ângulo de direção da roda traseira, que em grande maioria das aplicações é considerado zero, devido ao eixo fixo traseiro. Os parâmetros l_f e l_r descrevem respectivamente a distância do centro de massa até o eixo frontal e traseiro, e por fim o parâmetro β representa o ângulo de deslizamento que pode ser definido por:

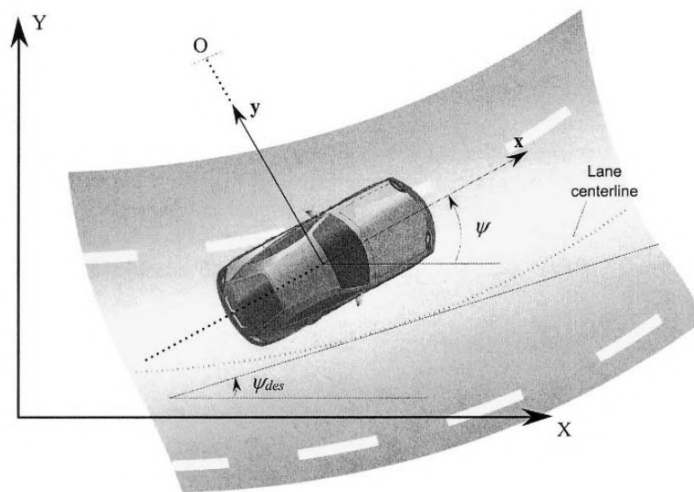
$$\beta = \tan^{-1} \frac{l_f \tan \delta_r + \tan l_r \delta_f}{l_f + l_r}, \quad (13)$$

2.2.2 Modelagem dinâmica

Em aplicações com velocidade mais elevada, se torna inviável a consideração feita de que a velocidade em cada roda está na direção da mesma, para isto se torna necessário o desenvolvimento de um modelo dinâmico que leva em consideração a dinâmica dos pneus (RAJESH, 2006).

O modelo apresentado pela Figura 3 pode ser descrito como modelo de bicicleta da dinâmica lateral com dois graus de liberdade, sendo os graus de liberdade a posição lateral do veículo y e a orientação angular, também chamada de ângulo de guinada representado por Ψ .

Figura 3 – Modelo dinâmico



Fonte: (RAJESH, 2006)

Aplicando a segunda lei de Newton a equação do momento angular em torno do eixo y , e considerando que para pequenos ângulos de deslizamento a força lateral dos pneus pode ser

considerada proporcional ao mesmo, mostra-se que o sistema pode ser descrito pelo modelo em espaço de estados, conforme a Equação (14).

$$\frac{d}{dt} \begin{bmatrix} y \\ \dot{y} \\ \Psi \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & 0 & -V_x - \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z V_x} & 0 & -\frac{2C_{\alpha f}l_f^2 + 2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta, \quad (14)$$

onde, $C_{\alpha f}$ e $C_{\alpha r}$ são classificados como os coeficientes de deslizamento das rodas frontais e traseiras, m a massa total do veículo, V_x a velocidade longitudinal, I_z o momento de inércia do veículo com relação ao eixo z e δ o ângulo de direção das rodas.

2.2.3 Modelagem dinâmica em termos de erros com relação a pista

Em aplicações como a deste trabalho, como um assistente de direção, se torna útil considerar um modelo onde suas variáveis de estado sejam o erro de posição e orientação com relação ao centro da pista. Baseando-se novamente na Figura 3 e com objetivo de se obter um modelo linear invariante no tempo, considera-se que a velocidade V_x é constante e que o raio R da pista é muito grande, gerando pequenos ângulos de deslizamento (RAJESH, 2006).

Com as considerações mostradas acima chega-se ao modelo descrito pela Equação (15), onde os estados e_y e e_Ψ descrevem respectivamente o erro de posição lateral com relação ao centro da pista e o erro angular com relação ao ângulo da pista.

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} e_y \\ \dot{e}_y \\ e_\Psi \\ \dot{e}_\Psi \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f} + 2C_{\alpha r}}{m} & \frac{-2C_{\alpha f}l_f + 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z V_x} & \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2 + 2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} \begin{bmatrix} e_y \\ \dot{e}_y \\ e_\Psi \\ \dot{e}_\Psi \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{mV_x} - V_x \\ 0 \\ -\frac{2C_{\alpha f}l_f^2 + 2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix} \dot{\Psi}_{des}, \end{aligned} \quad (15)$$

3 DESENVOLVIMENTO

Neste Capítulo, são descritos três tópicos importantes para o desenvolvimento do trabalho. O primeiro tópico está focado na escolha dos componentes de hardware e software para a construção do carro em escala reduzida, no segundo tópico descreve-se as etapas e técnicas utilizadas no processamento de imagem para detecção de pista. Por fim, no terceiro tópico descreve-se a aplicação da técnica de controle preditivo estocástico baseado em modelo a um assistente de direção.

3.1 DESCRIÇÃO DO *HARDWARE* E *SOFTWARE*

Para especificação do *hardware* deste projeto, primeiramente realizou-se uma pesquisa focada em comunidades já existentes de desenvolvedores de soluções para veículos autônomos. Após o estudo e descarte de algumas soluções encontrou-se a comunidade chamada *DonkeyCar*.

Criada na Califórnia nos Estados Unidos, a comunidade *DonkeyCar* consiste em voluntários, que tem como objetivo compartilhar interesses para construir os seus próprios protótipos de carros autônomos em pequena escala, como pode ser visto na Figura 4. Para tal objetivo a comunidade conta com tutorias de construção de *hardware* e *software* para que qualquer indivíduo, com interesse, possa explorar este universo e construir o seu próprio protótipo (DONKEY CAR, 2020).

Baseando-se nos conceitos de reprodutibilidade, modularidade, relativo baixo custo e robustez, decidiu-se utilizar a plataforma de hardware descrita pela comunidade *DonkeyCar*, tendo em vista que o conceito é provado a partir do extenso uso pela comunidade.

Portanto, esta seção terá como objetivo apresentar de forma objetiva os componentes resultantes utilizados para criação do protótipo em pequena escala utilizado neste trabalho.

3.1.1 Câmera

Para criação de sistemas ADAS vários sensores diferentes podem ser utilizados em conjunto, através das técnicas de fusão sensorial, do Inglês, *sensor fusion* (KAWASAKI; KIENCKE, 2004) (TUCKER; HEENAN; BUCHANAN, 2005). Sendo que em aplicações automotivas os sensores mais utilizados são câmeras, radares e lidars (AUTOCRYPT, 2021).

Com o avanço da tecnologia de processamento de imagens, tratando-se de carros autônomos, um dos sensores mais poderosos capazes de gerar maior nível de interação com o ambiente são as câmeras, devido ao seu relativo baixo custo e disponibilidade limitada de espaço em um carro em miniatura, este projeto pretende utilizá-la para a geração de referência de trajetória.

A câmera escolhida para este projeto é baseada nas orientações da comunidade *Donkey Car* e é denominada *SainSmart Wide Angle Fish-Eye Camera*, pode ser vista na Figura 5, que conta com as características de 5 Mega Pixels, resolução de 2592 x 1944, lente para ajuste de foco e ângulo de abertura de 160 graus (AMAZON, 2022). Após a aquisição e fase de testes

Figura 4 – Exemplo de *Donkey Car*.



Fonte: (DONKEY CAR, 2020)

experimentais, durante o processo de remoção de distorções e calibragem, observou-se uma alta distorção radial devido ao seu alto ângulo de abertura da câmera. Baseado nisso para futuros trabalhos incentiva-se o estudo mais aprofundado de diferentes modelos de câmera. para verificar estes efeitos de distorção e possível otimização do modelo.

Figura 5 – *SainSmart Wide Angle Fish-Eye Camera*



Fonte: (AMAZON, 2022)

3.1.2 *Raspberry Pi 4 Model B*

O *Raspberry Pi* é considerado um computador de placa única de tamanho reduzido, onde neste projeto será responsável pelo processamento dos sinais e execução dos algoritmos de controle. O modelo escolhido para o projeto é chamado *Raspberry Pi 4 Model B* e pode ser visto na Figura 6. O modelo conta com processador Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) SoC de 64 bits a 1,5 GHz e memória Ram de 4GB (AMAZON, 2022).

Figura 6 – *Raspberry Pi 4 Model B*



Fonte: (AMAZON, 2022)

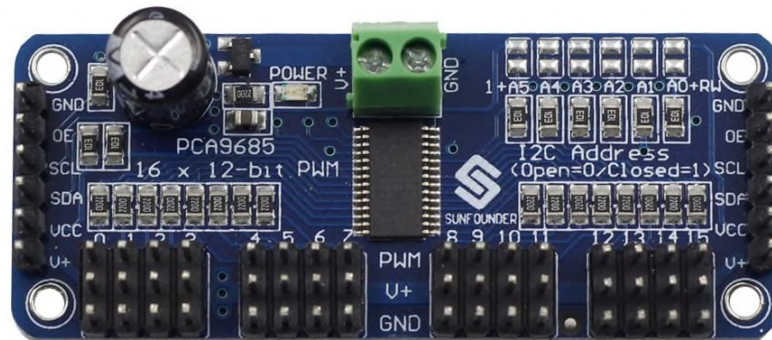
3.1.3 *PWM Servo Driver*

As placas de condicionamento de sinais PWM são necessárias tanto para variação de velocidade do motor de tração do protótipo, quanto para o acionamento do servo motor responsável pela direção do carro. O modelo escolhido para o projeto é chamado SunFounder PCA9685 16 Channel 12 Bit PWM Servo Driver e pode ser visto na Figura 7. O modelo conta com 16 canais de PWM de 12 bits de resolução, controlados via comunicação I2C com o controlador. A grande vantagem deste circuito é que ele contém circuito de *clock* próprio, fazendo com que o controlador envie apenas a referência de posição ou velocidade através da comunicação serial (AMAZON, 2022).

3.1.4 *RC Car*

Para a parte mecânica do protótipo, com o objetivo de torná-lo prático e eficiente, optou-se por adquirir um carro em escala reduzida 1/16, comumente chamado de *RC Car*. O modelo

Figura 7 – SunFounder PCA9685 16 Channel 12 Bit PWM



Fonte: (AMAZON, 2022)

escolhido para o projeto é chamado *Exceed Racing Desert Short Course Truck 1/16 Scale* e pode ser visto na Figura 8.

Figura 8 – *Exceed Racing Desert Short Course Truck 1/16 Scale*



Fonte: (AMAZON, 2022)

O modelo conta com as dimensões mostradas na Tabela 1, direção com tração nas quatro rodas, suspensão, motor escovado 380 e servo motor com capacidade de 3 kg (AMAZON, 2022).

Tabela 1 – Dimensões.

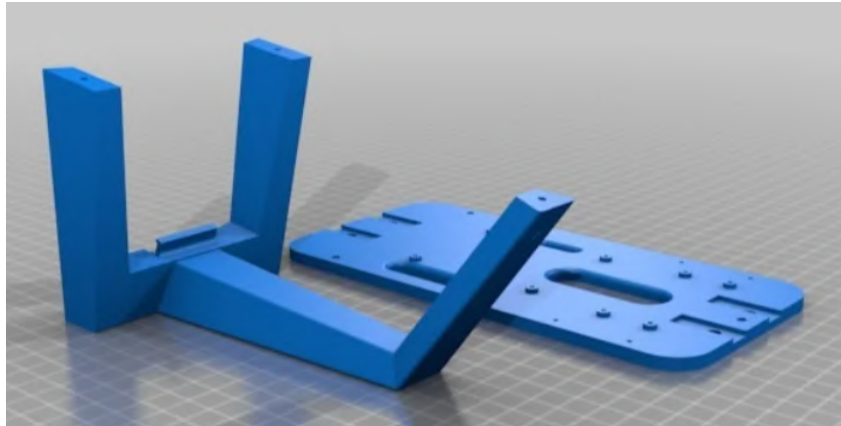
Elementos	Medidas
Comprimento	360 mm
Altura	132 mm
Largura	200 mm
Diâmetro da roda	72 mm
Largura da roda	30 mm
Distância entre eixos	176 mm
Peso	1.12 kg

Fonte: Próprio Autor

3.1.5 Peças impressas 3D

Com o avanço da tecnologia de prototipagem tornou-se mais fácil o acesso a impressoras 3D, que possibilitam a impressão de peças customizadas nos mais diversos formatos. Neste projeto são utilizadas peças disponibilizadas pela comunidade *Donkey Car*, como podem ser vistas na Figura 9, que tem como finalidade fixar as placas e câmera do protótipo.

Figura 9 – Peças impressas 3D



Fonte: (THINGIVERSE, 2017)

3.1.6 Baterias

As baterias são componentes essenciais quando se trata de robótica, neste projeto serão utilizadas duas baterias, uma para o sistema de controle e outra para a atuação dos drivers e motores. Para o sistema de controle será utilizada uma bateria LiPo comercial de 6700mAh, com saída de 5V 2A, como pode ser visto na Figura 10, as grandes vantagens são a segurança e carregador integrado (AMAZON, 2022).

Para o sistema de atuação será utilizada outra bateria LiPo de modelo LB2200LP2 de 7.4V e 2200mAh, com capacidade de descarga de 30C e plug de saída XT60, conforme a Figura 11 (MERCADO LIVRE, 2022).

3.1.7 Robot Operating System – ROS

Como já citado neste trabalho, a escolha da plataforma de software e linguagem de programação baseou-se principalmente no conceito de replicabilidade e código aberto. Para isto, após algumas pesquisas e indicações encontrou-se a plataforma chamada ROS.

Criado em 2007 na Universidade de Stanford, a sigla ROS do inglês *Robot Operating System* é definida como uma coleção de frameworks para desenvolvimento de robôs. Tendo como características (ROS, 2019):

- a) Modularidade, possibilitando compartilhamento de códigos;
- b) Larga utilização e constante desenvolvimento;

Figura 10 – Bateria de controle



Fonte: (AMAZON, 2022)

Figura 11 – Bateria de atuação

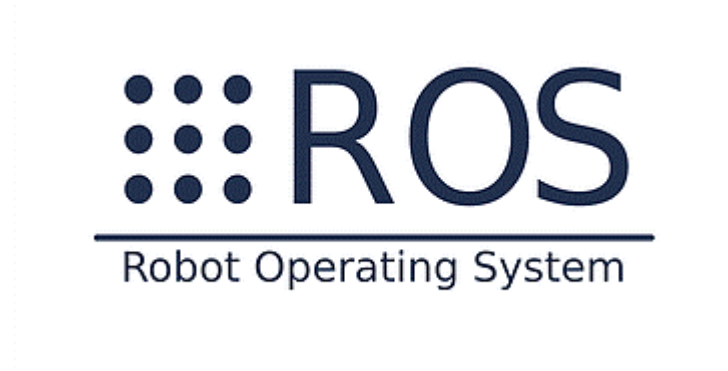


Fonte: (MERCADO LIVRE, 2022)

- c) Código aberto;
- d) Suporte a diferentes linguagens como python e C++;
- e) Variedade de bibliotecas e pacotes disponíveis.

O ROS é um sistema de código aberto e meta-operante para robôs. Ele fornece os serviços esperados de um sistema operacional, incluindo abstração de hardware, controle de dispositivos de baixo nível, implementação de funcionalidade de uso comum, passagem de mensagens entre processos e gerenciamento de pacotes. Ele também fornece ferramentas e bibliotecas para obter,

Figura 12 – ROS



Fonte: (ROS, 2019)

construir, escrever e executar código em vários computadores (ROS, 2019).

O sistema é composto por pacotes que disponibilizam nós com código embarcado, estes nós se comunicam entre si por meio de mensagens transportadas por tópicos, fazendo a interação entre os diferentes nós com diferentes aplicações dependendo da finalidade do sistema embarcado do robô. Estes pacotes podem ser criados pelo usuário ou reutilizados através do compartilhamento da comunidade, fornecendo agilidade e eficiência para a aplicação. A aplicação criada neste trabalho será mostrada com mais detalhes na Seção 4.2.1.

3.2 PROCESSAMENTO DE IMAGEM PARA DETECÇÃO DE PISTA

Para que se possa desenvolver um assistente de direção, é necessário utilizar uma tecnologia que consiga transformar a pista em uma referência para o controlador, neste trabalho optou-se por utilizar uma câmera, que fará a detecção de imagem das linhas e trará como saída o erro de deslocamento do veículo com relação ao centro da pista e o erro de orientação angular.

3.2.1 Correção de distorção

Câmeras são responsáveis por representar objetos do mundo real de três dimensões em um plano de imagem de duas dimensões, neste processo ocorrem diversas distorções de imagem. As distorções mais comuns e visíveis são denominadas de distorção radial, observada na Figura 13 e tangencial que diz respeito ao alinhamento da lente com o plano da imagem.

Com objetivo de reduzir as distorções é possível realizar um processo de calibração da câmera, que através de múltiplas fotos de um tabuleiro de xadrez de dimensões conhecidas, é capaz de levantar os coeficientes das distorções resultando em duas matrizes, uma de caracterização da câmera e outra de distorção. Neste trabalho a calibração da câmera foi realizada utilizando a linguagem Python juntamente com a biblioteca OpenCV, que utiliza as funções *calibrateCamera* e *findChessboardCorners* que são baseadas na técnica descrita em (ZHANG, 2000).

Após a criação do banco de aproximadamente trinta imagens com posições e ângulos

Figura 13 – Distorção radial



Fonte: (ATOMOCLAST, 2018)

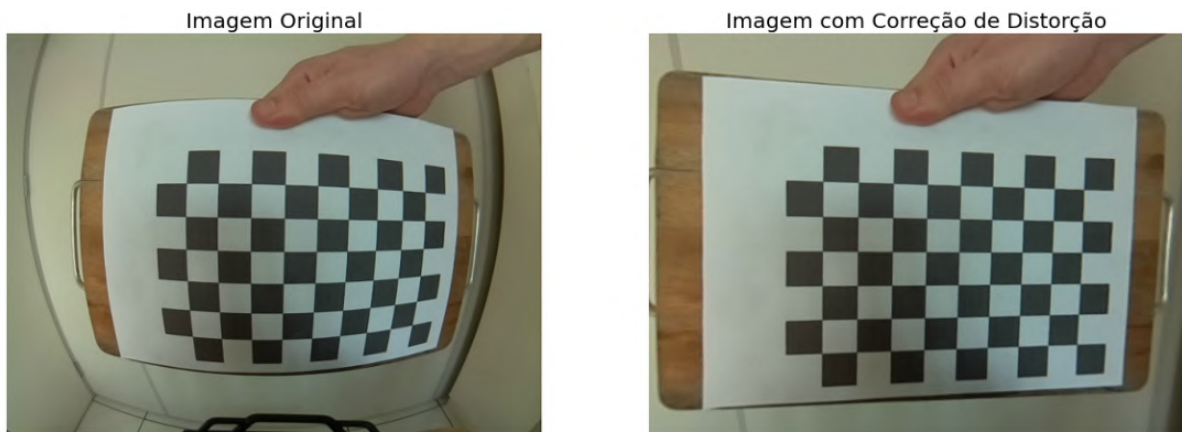
diferentes, conforme pode ser visto em (GITHUB, 2022), e aplicação da técnica descrita acima, chegam-se as Matrizes (16) e (17).

$$\begin{bmatrix} 309.5954735138005276 & 0 & 333.6373845671422487 \\ 0 & 309.9124724576500398 & 218.6567383907390933 \\ 0 & 0 & 1 \end{bmatrix} \quad (16)$$

$$\begin{bmatrix} -0.322988 & 0.114717 & -0.000266 & -0.000306 & -0.019262 \end{bmatrix} \quad (17)$$

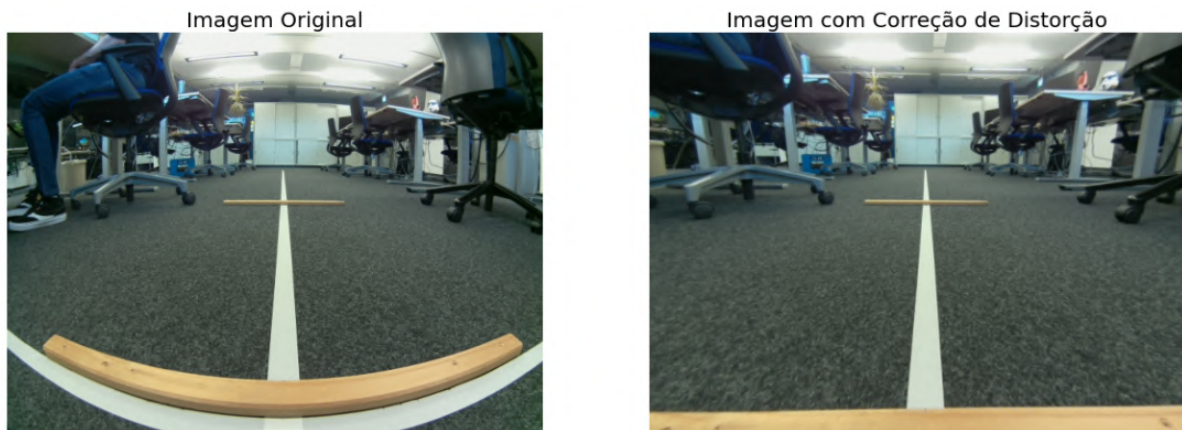
Após o levantamento das matrizes descritivas da câmera, se torna possível a utilização da função *undistort* da biblioteca OpenCV, gerando o resultados mostrados nas Figuras 14 e 15, onde do lado esquerdo apresenta-se a foto original e do lado direito a imagem com correção de distorção.

Figura 14 – Resultado de calibração da câmera com tabuleiro



Fonte: Próprio Autor

Figura 15 – Resultado de calibração da câmera com superfície



Fonte: Próprio Autor

3.2.2 Transformação de perspectiva

Após a correção de distorção da imagem, para que se possa realizar a identificação dos ângulos da pista, se faz necessário uma transformação de perspectiva para uma visão de topo, conhecida por *Bird's Eye View*. Para implementação desta transformação se faz necessário a delimitação de uma região de interesse, descrita por coordenadas em pixels chamadas de coordenadas fonte. Após definida a região de interesse, faz-se necessário definir as coordenadas de destino onde deve-se manter as proporcionalidades de largura e comprimento com relação ao mundo físico. Na Figura 17 destacado em vermelho, pode-se visualizar a região de interesse de fonte, onde os critérios utilizados para definição são a exploração máxima da dimensão horizontal da imagem, e uma distância de um metro a frente do veículo. Para definição das coordenadas de destino são mantidas as proporções do espaço real capturado. As coordenadas definidas neste projeto podem ser vistas na Figura 16, e os resultados da transformação de perspectiva podem ser vistos na Figura 17.

3.2.3 Filtro HLS dinâmico

Para que seja possível a detecção das linhas da pista, se faz necessário a conversão do espaço de cores da imagem, por padrão a biblioteca OpenCV utiliza o formato BGR, porém para este tipo de aplicação, devido as cores de faixa predominantemente amarelas e brancas, o formato HLS se torna mais interessante, onde os canais HLS representam respectivamente, matiz, luminosidade e saturação.

Neste trabalho após testes de performance de detecção, optou-se por utilizar apenas o canal de luminosidade, que devido ao contraste de faixas brancas e amarelas em um fundo mais escuro como o asfalto facilita-se a detecção. Através do canal de luminosidade é possível definir um valor de limiar e com isso criar uma nova imagem binária chamada de máscara, onde será contida apenas a pista.

Para a definição do limiar de luminosidade de forma mais robusta a técnica adotada é

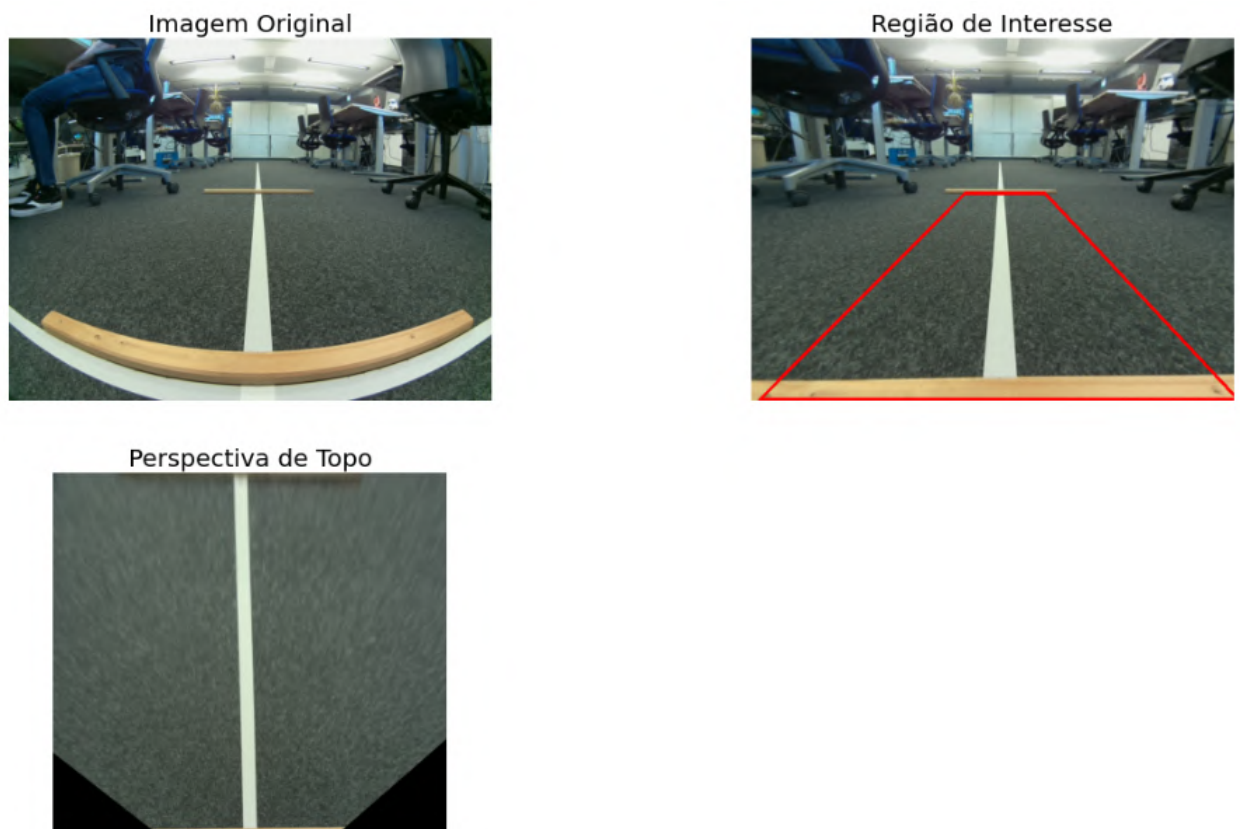
Figura 16 – Coordenadas de fonte e destino

```
src = np.float32([
    (389,205),
    (283,205),
    (11,478),
    (645,478)
])

dst = np.float32([
    (320, 0),
    (0,0),
    (0, 661),
    (320, 661)
])
```

Fonte: Próprio Autor

Figura 17 – Transformação de perspectiva



Fonte: Próprio Autor

o chamado filtro dinâmico, onde ao invés de assumir um valor fixo, utiliza-se uma leitura dos valores médios e máximos do canal de luminosidade, e através de uma porcentagem deste valor

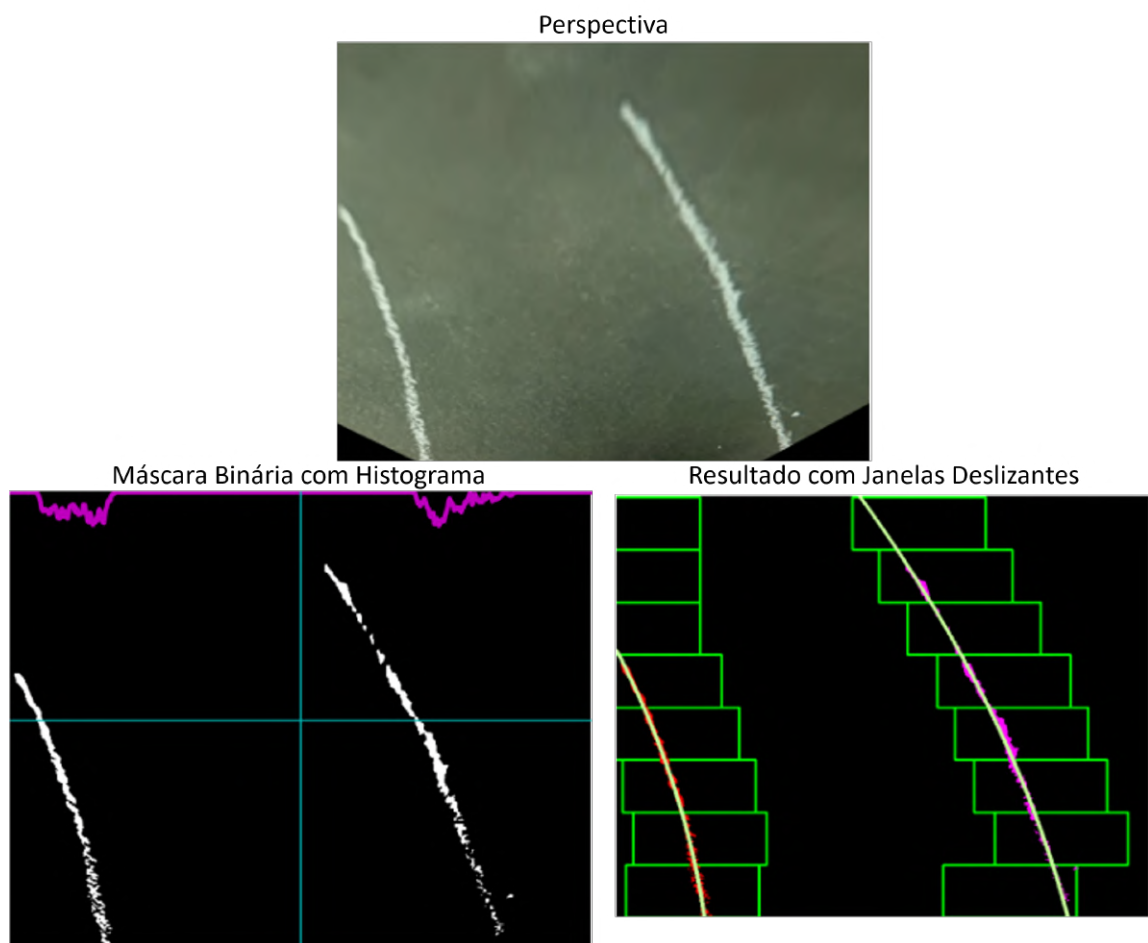
se escolhe o limiar, fazendo assim com que o filtro se adapte em uma mudança de luminosidade do ambiente. Alguns resultados da aplicação do filtro dinâmico serão apresentados na Seção 4.2.3.

3.2.4 Técnica das janelas deslizantes

Após a criação da máscara binária do passo anterior, pode-se implementar uma técnica chamada janelas deslizantes, que tem como objetivo rastrear e diferenciar as linhas esquerda e direita da rodovia.

A técnica consiste da criação de um histograma das colunas da imagem, onde os dois maiores picos podem ser caracterizados como o ponto inicial para o rastreamento das linhas da pista, que se dá através de janelas de comprimento e largura pré programadas pelo usuário. As janelas são responsáveis por classificar os pontos que pertencem a cada linha da pista, sendo que para a realocação do centro desta janela a cada novo *frame* é necessário um número mínimo de pontos detectados. O resultado pode ser visualizado na Figura 18.

Figura 18 – Janelas deslizantes

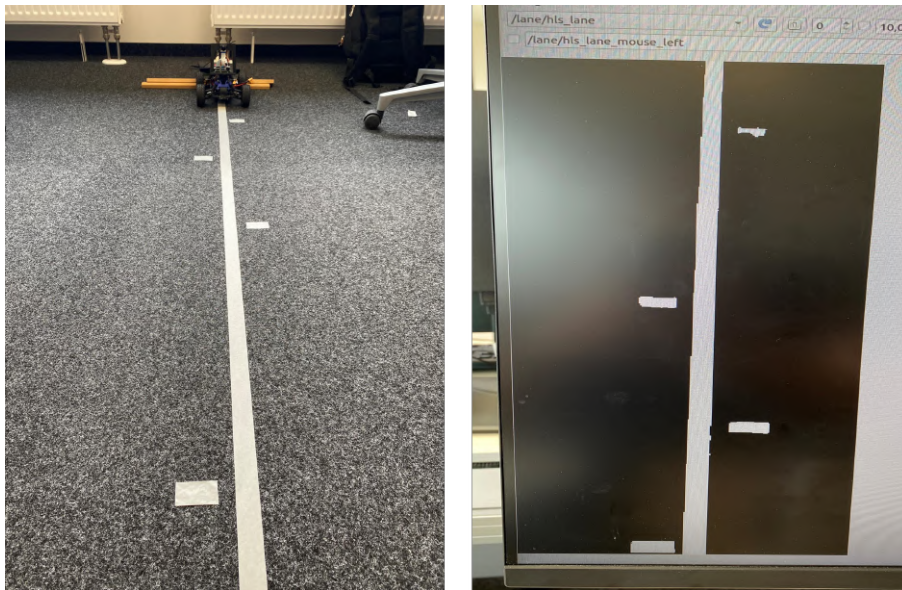


Fonte: Próprio Autor

3.2.5 Conversão pixels por metro e levantamento dos polinômios descritivos

Por fim, para que seja possível realizar o levantamento dos polinômios resultantes das linhas caracterizadas pela pista, se faz necessário uma conversão de pixels para metro, pois assim tem-se valores reais em metros que podem ser enviados ao controlador, para isto se faz uma série de fotos de um objeto de dimensões conhecidas em diferentes posições e tira-se uma média do parâmetro pixels por metro, nas direções horizontais e verticais, como mostrado na Figura 19.

Figura 19 – Ensaio de conversão metros por pixel



Fonte: Próprio Autor

Com posse dos parâmetros de conversão e da imagem binária resultante, torna-se possível aproximar um polinômio descritivo das linhas da pista, de onde se tira as informações de erro de posição com relação ao centro da imagem e erro angular da linha resultante.

3.3 MODELO DE ASSISTENTE DE DIREÇÃO ESTOCÁSTICO

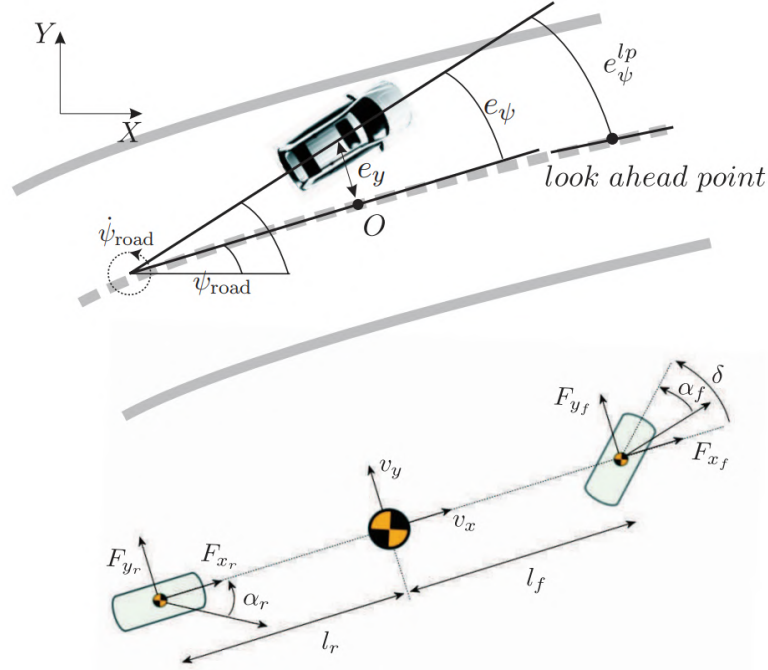
Através dos conceitos discutidos no Capítulo 2, neste trabalho aplicou-se de maneira prática um controlador preditivo estocástico em um assistente de direção, para isto se faz necessário incluir o comportamento do motorista juntamente no modelo do veículo, gerando assim o modelo final do sistema, as restrições e a função custo.

3.3.1 Modelagem

Considerando-se o modelo descrito pela Equação (15), que pode ser representado pela Figura 20. Para esta aplicação com a finalidade de obter um modelo linear invariante no tempo, define-se uma velocidade constante e sem mudanças no trajeto, resultando na anulação do termo Ψ_{des} da Equação (15).

Partindo destas modificações, chega-se ao modelo da dinâmica lateral do veículo descrito pela Equação (18).

Figura 20 – Modelagem do carro



Fonte: Gray et al. (2013)

$$\dot{\xi}(t) = A_c \xi(t) + B_c u(t), \quad (18)$$

onde os estados são definidos como $\xi(t) = [e_y, \dot{e}_y, e_\psi, \dot{e}_\psi]^\top \in \mathbb{R}^4$, e a ação de controle u sendo o ângulo de direção δ . As matrizes de dinâmica do sistema são descritas como:

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mV_x} & \frac{2C_{\alpha f} + 2C_{\alpha r}}{m} & -\frac{2C_{\alpha f}l_f + 2C_{\alpha r}l_r}{mV_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z V_x} & \frac{2C_{\alpha f}l_f - 2C_{\alpha r}l_r}{I_z} & -\frac{2C_{\alpha f}l_f^2 + 2C_{\alpha r}l_r^2}{I_z V_x} \end{bmatrix}, \quad (19)$$

$$B_c = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2C_{\alpha f}l_f}{I_z} \end{bmatrix}. \quad (20)$$

Com o objetivo da aplicação de um assistente de direção, se faz necessária a inclusão do comportamento do motorista no modelo, para isto define-se o ângulo de direção contendo duas componentes, δ_C sendo gerada pelo controlador e δ_D gerada pelo modelo do motorista, definido como:

$$\delta_D(t) = F_D \xi(t) + w(t), \quad (21)$$

em que $F_D = [K_y, 0, K_\psi, 0]$, e $w(t)$ é um sinal estocástico com distribuição normal, tal que $w(t) = \mathcal{N}(\bar{w}, \Sigma)$.

Com posse do novo sinal de controle pode-se definir o novo sistema como:

$$\dot{\xi}(t) = A_c \xi(t) + B_c(F_D \xi(t) + v(t) + w(t)), \quad (22)$$

resultando em:

$$\dot{\xi}(t) = A_{dm} \xi(t) + B_c v(t) + D_c w(t), \quad (23)$$

em que $A_{dm} = (A_c + B_c F_D)$ descreve o comportamento em malha fechada, $v(t) = \delta_C(t)$, $D_c = B_c$ e $w(t)$ um sinal estocástico de distribuição normal.

Discretizando o sistema descrito pela Equação (23) com uma taxa de amostragem T_s , é obtido o sistema:

$$\xi(k+1) = A_d \xi(k) + B_d v(k) + D_d w(k), \quad (24)$$

Conforme descrito na Seção 1.1.2 para maior robustez no sistema estocástico, a ação de controle é definida pela Equação (25) como uma função afim, com ganho K calculado como o ganho de um LQR:

$$v(k) = K \xi(k) + c(k), \quad (25)$$

De posse do sistema estocástico, utilizando a separação de estados:

$$\xi_k = z_k + e_k, \quad (26)$$

definindo $\mathbb{E}[\xi_k] = z_k$, obtém-se:

$$\begin{aligned} z_{k+1} &= \Phi z_k + B_d c_k, \\ e_{k+1} &= \Phi e_k + D_d w_k, \end{aligned} \quad (27)$$

em que $\Phi = A_d + B_d K$.

Expandindo ao longo do horizonte de predição N , encontra-se o sistema descrito pelas Equações (28) e (29).

$$\underbrace{\begin{bmatrix} z_{k+1} \\ z_{k+2} \\ \vdots \\ \vdots \\ z_{k+N} \end{bmatrix}}_Z = \underbrace{\begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \vdots \\ \Phi^N \end{bmatrix}}_{H_z} z_k + \underbrace{\begin{bmatrix} B_d & 0 & \dots & 0 \\ \Phi B_d & B_d & \ddots & 0 \\ \Phi^2 B_d & \Phi B_d & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \Phi^{N-1} B_d & \Phi^{N-2} B_d & \dots & B_d \end{bmatrix}}_{H_c} \underbrace{\begin{bmatrix} c_k \\ c_{k+1} \\ \vdots \\ \vdots \\ c_{k+N-1} \end{bmatrix}}_C, \quad (28)$$

onde $H_z \in \mathbb{R}^{nN \times n}$ descreve o comportamento da evolução dos estados e $H_c \in \mathbb{R}^{nN \times N}$ descreve o comportamento da evolução dos sinais de entrada determinísticos.

$$\underbrace{\begin{bmatrix} e_{k+1} \\ e_{k+2} \\ \vdots \\ \vdots \\ e_{k+N} \end{bmatrix}}_{\mathcal{E}} = \underbrace{\begin{bmatrix} \Phi \\ \Phi^2 \\ \vdots \\ \vdots \\ \Phi^N \end{bmatrix}}_{H_z} e_k + \underbrace{\begin{bmatrix} D_d & 0 & \dots & 0 \\ \Phi D_d & D_d & \ddots & 0 \\ \Phi^2 D_d & \Phi D_d & \ddots & 0 \\ \vdots & \vdots & \ddots & 0 \\ \Phi^{N-1} D_d & \Phi^{N-2} D_d & \dots & D_d \end{bmatrix}}_{H_w} \underbrace{\begin{bmatrix} w_k \\ w_{k+1} \\ \vdots \\ \vdots \\ w_{k+N-1} \end{bmatrix}}_W, \quad (29)$$

onde $H_w \in \mathbb{R}^{nN \times N}$ descreve o comportamento da evolução dos sinais de entrada estocásticos. resultando em:

$$\mathcal{E} = S_{\mathcal{E}} \xi_k + S_c C + S_w W. \quad (30)$$

3.3.2 Função custo

A função custo utilizada para o problema de otimização, responsável por ponderar os esforços com relação aos estados e sinal de controle, é descrita pela Equação (31):

$$\begin{aligned} J(\xi_k, v) &= \mathbb{E} \left[\sum_{i=0}^{N-1} \left[\xi'_{k+i} Q \xi_{k+i} + v'_{k+i} R v_{k+i} \right] \right] \\ &= \sum_{i=0}^{N-1} z_{k+i} (Q' + K' R K) z_{k+i} + c'_{k+i} R c_{k+i} + 2 z'_{k+i} K' R c_{k+i} \end{aligned} \quad (31)$$

Reescrita ao longo do horizonte, pode ser definida :

$$J(\xi_k, v) = \frac{1}{2} C' H C + z'_k f C, \quad (32)$$

onde:

$$\begin{aligned} H &= H'_c \bar{Q} H_c + (\bar{K} H_c + I)' \bar{R} (\bar{K} H_c + I), \\ f &= H'_z H_c + H'_z \bar{K}' + H'_z \bar{K}' H_c' \bar{K}', \end{aligned} \quad (33)$$

em que \bar{Q} , \bar{R} e \bar{K} são matrizes diagonais formadas por N matrizes Q , R e K , respectivamente.

3.3.3 Restrições

Devido a limitação física do ângulo de acionamento das rodas é definida uma restrição do tipo *hard* no sinal de controle, definida por:

$$v_{\min} \leq v_k = K \xi_k + c_k \leq v_{\max}, \quad (34)$$

que é reescrita ao longo do horizonte e em função da variável de decisão C :

$$C \leq v_{\max} \mathbf{1}_{N \times 1} - \overline{K} H_z z_k, \quad (35)$$

e

$$C \leq -v_{\min} \mathbf{1}_{N \times 1} + \overline{K} H_z z_k, \quad (36)$$

sendo $\mathbf{1}_{N \times 1}$ um vetor unitário de dimensão $N \times 1$.

Para simulação de obstáculos na pista e devido ao comportamento estocástico do sistema, definem-se restrições no erro lateral $e_{y_{\max}}$ e $e_{y_{\min}}$ denominadas de *chance-constraints*, descritas por:

$$\mathcal{P}\{g' \xi_k \leq e_{y_{\max}}\} \geq 1 - p, \quad (37)$$

em que

$$g' = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}. \quad (38)$$

Utilizando a decomposição das partes determinísticas e estocásticas e considerando w_k com distribuição Gaussiana, define-se a seguinte relação:

$$GH_c C \leq e_{y_{\max}} \mathbf{1}_{N \times 1} - GH_z z_k - \lambda_i. \quad (39)$$

sendo G a matriz g' expandida ao longo do horizonte N e a compensação λ_i sendo definida como:

$$\lambda_i(1 - p) = \sqrt{g' E_i \mathbf{W} E_i' g} \mathcal{N}^{-1}(1 - p), \quad (40)$$

sendo E_i a linha i da matriz H_w . Ainda. De forma análoga defini-se a restrição para o valor $e_{y_{\min}}$ como:

$$-GH_c C \leq -e_{y_{\min}} \mathbf{1}_{N \times 1} + GH_z z_k - \lambda_i. \quad (41)$$

4 RESULTADOS

Neste Capítulo, na primeira parte serão mostrados dois resultados de simulação, e na segunda parte serão mostrados alguns resultados práticos, como estrutura final do software, custo computacional do sistema de processamento imagem, aplicação do filtro HLS dinâmico e por fim um ensaio prático do veículo.

4.1 SIMULAÇÃO

A partir dos conceitos e modelos discutidos nos capítulos anteriores, desenvolveu-se uma simulação em Matlab para aprofundamento dos conceitos e testes dos parâmetros de otimização. Os resultados são apresentados em dois ensaios, ambos foram obtidos através da inserção de restrições de segurança que podem ser interpretados como obstáculos no trajeto do veículo. Para o estudo do comportamento estocástico varia-se a porcentagem de aceitação de quebra da restrição, fazendo com que o controlador varie o seu conservadorismo com relação ao obstáculo. Os parâmetros utilizados para construção da simulação podem ser visualizados na Tabela 2.

Tabela 2 – Parâmetros do sistema e do controlador.

Parâmetro	Valor	Unidade
m	1.24	kg
I_z	0.75	$\text{kg}\cdot\text{m}^2$
C_α	1	-
l_f, l_r	0.0885	m
T_s	142.8	ms
V_x	0.8	m/s
N	15	-
K_y	0.005	-
K_ψ	0.2	-
\bar{w}	0	rad
W	0.01	rad
u_{max}	0.262	rad
u_{min}	-0.262	rad

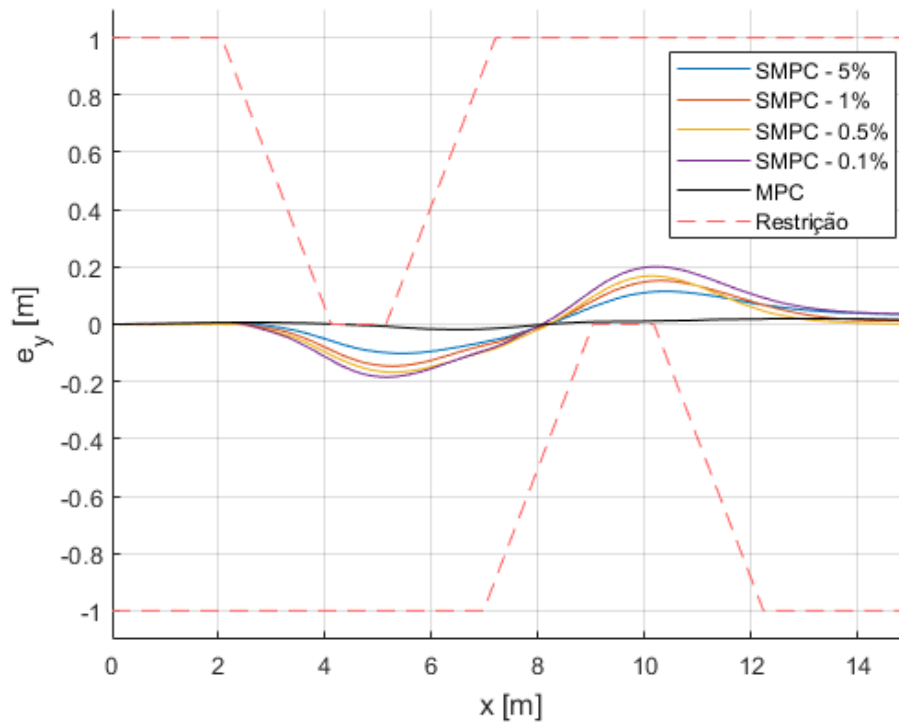
Fonte: Próprio Autor

A diferença entre o primeiro e segundo ensaio consta na presença do modelo do condutor, onde no segundo ensaio remove-se o comportamento do condutor anulando os parâmetros K_y e K_ψ . Tal mudança é motivada devido a limitação que será discutida na parte dos resultados práticos.

Os resultados do primeiro ensaio podem ser visualizados nas Figuras 21 e 22 e os resultados do segundo ensaio podem ser visualizados nas Figuras 23 e 24

Na Figura 21 apresenta-se de forma simultânea o resultado de cinco ensaios de simulação, onde nos quatro primeiros desenvolve-se o controlador preditivo estocástico baseado em modelo,

Figura 21 – Resultado de simulação - ensaio com condutor - sinal de saída



Fonte: Próprio Autor

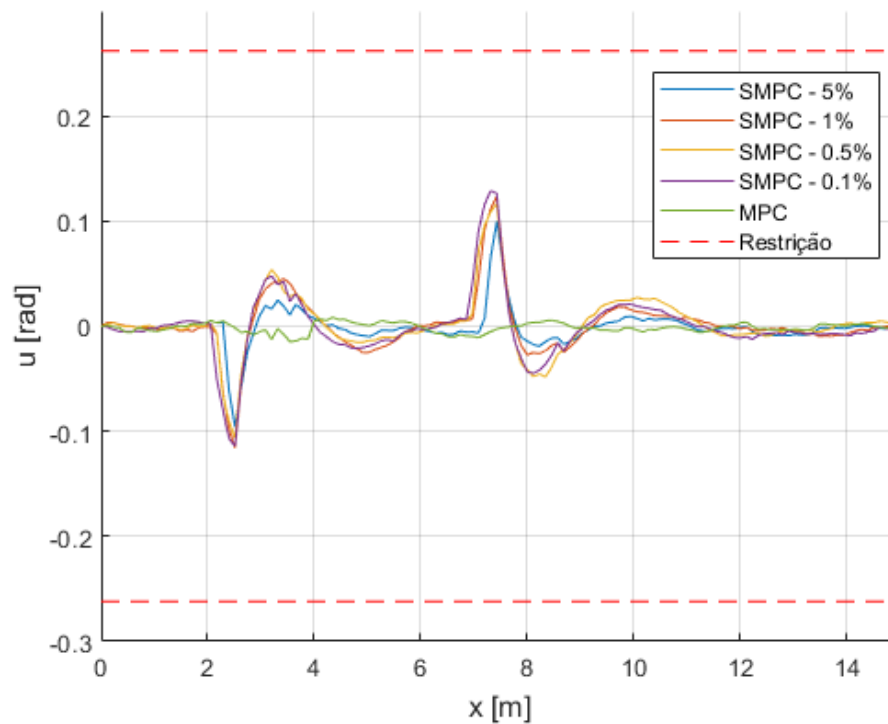
com diferentes porcentagens de aceitação de quebra de restrição, conforme mostrado na legenda, e no quinto ensaio para fins de comparação, mostra-se um controlador preditivo baseado em modelo sem a implementação estocástica.

A restrição de segurança no trajeto do veículo está indicada pela linha tracejada no formato de trapézio, esta é inserida no projeto do controlador e pode ser customizada conforme a interação com o ambiente, por exemplo no caso da inviabilidade de uma pista, pedestres, ou até mesmo um outro veículo.

Através da análise da Figura 21, observa-se que quanto menor a porcentagem da aceitação da quebra de restrição, maior será a margem de segurança entre o obstáculo e o veículo, levando em conta que a qualquer momento pode-se haver um distúrbio no sinal de controle acarretando em uma mudança de trajetória. No caso da aplicação do controle preditivo clássico, observa-se que o controlador não apresenta nenhuma forma de conservadorismo com relação ao obstáculo, sendo possível uma quebra de restrição no caso de um distúrbio mais severo.

Na Figura 22 observa-se os sinais de controle para o primeiro ensaio descrito anteriormente, onde nos resultados do SMPC, conforme esperado percebe-se uma diferença de amplitude no sinal de controle, através da trajetória mais agressiva mostrada na figura anterior. O controlador preditivo clássico apresenta uma pequena amplitude no sinal de controle, tendo em vista que não ocorrem grandes desvios de trajetória. Através da linha tracejada, demonstra-se a restrição no sinal de controle, que é definida como uma restrição física, devido a limitação do sistema de

Figura 22 – Resultado de simulação - ensaio com condutor - sinal de controle



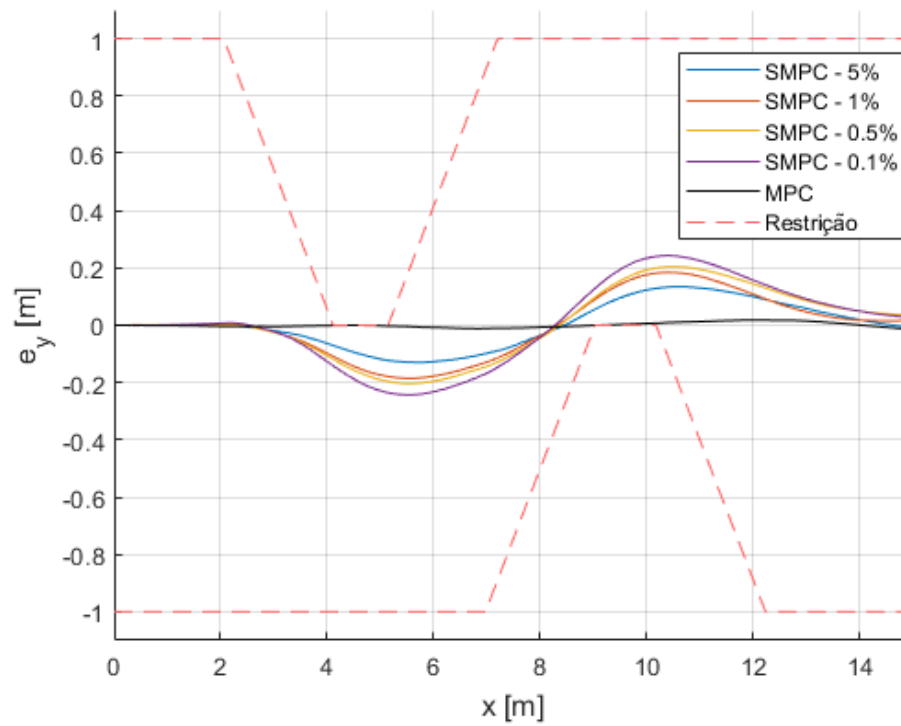
Fonte: Próprio Autor

direção do carro em miniatura.

Na Figura 23 apresenta-se o resultado de simulação do segundo ensaio, onde retira-se a ação do condutor, com objetivo de demonstrar que é possível tal modificação, tendo em vista que o resultado prático será obtido desta maneira. O comportamento apresentado é similar ao primeiro ensaio, apenas com um pequeno aumento na excursão da trajetória.

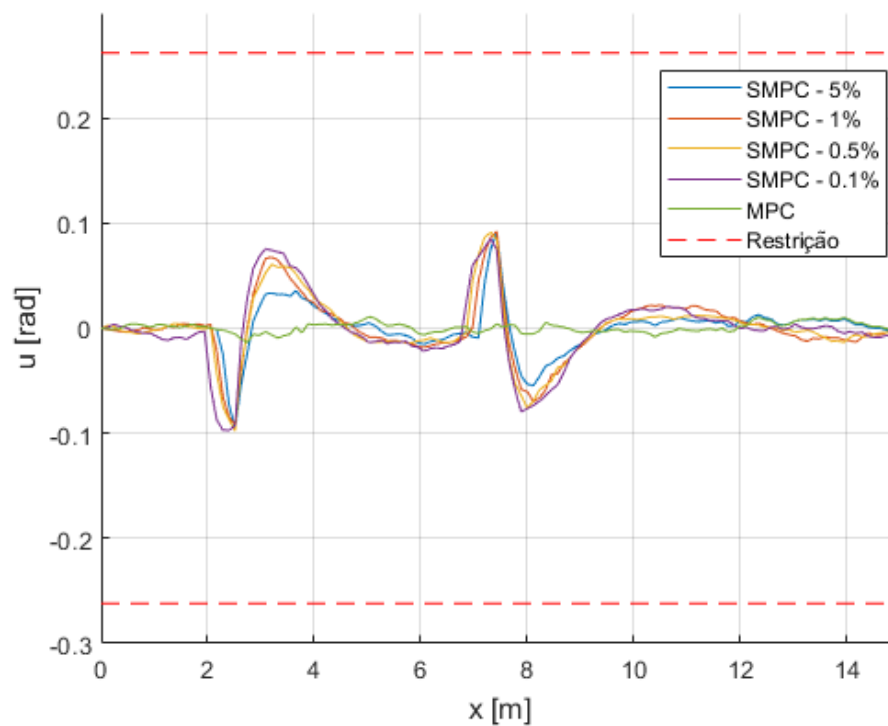
Por fim, na Figura 24 apresenta-se os sinais de controle obtidos no segundo ensaio.

Figura 23 – Resultado de simulação - ensaio sem condutor - sinal de saída



Fonte: Próprio Autor

Figura 24 – Resultado de simulação - ensaio sem condutor - sinal de Controle



Fonte: Próprio Autor

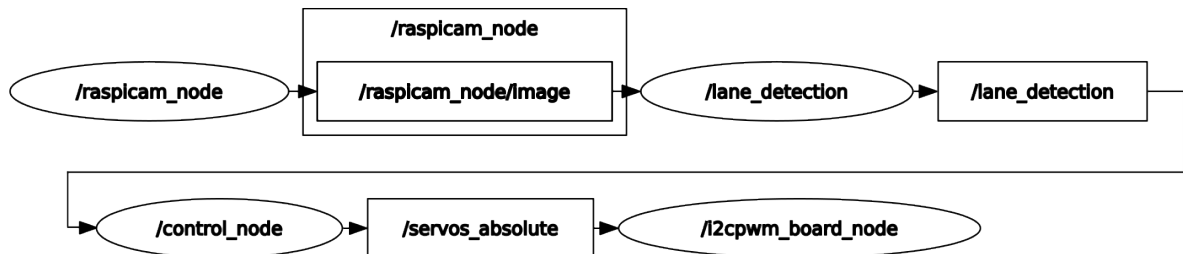
4.2 PRÁTICO

Nesta parte apresenta-se alguns resultados práticos do sistema embarcado adquiridos ao decorrer dos ensaios.

4.2.1 Estrutura de software

Como já citado anteriormente, este trabalho utiliza o framework ROS, *Robot Operating System*, o sistema desenvolvido consiste de quatro nós, como pode ser visto de maneira simplificada na Figura 25.

Figura 25 – Estrutura ROS simplificada



Fonte: Próprio Autor

O primeiro nó nomeado "raspicam_node", tem como objetivo apenas realizar a aquisição da imagem através da câmera e enviá-la ao segundo nó nomeado "lane_detection". Neste segundo nó é realizado todo o processamento da imagem conforme descrito nos capítulos anteriores, tendo como resultado as variáveis de estado necessárias para alimentação do terceiro nó nomeado "control_node".

O terceiro nó é onde se encontra toda a estrutura de controle preditivo e realimentação do modelo, responsável pelo cálculo da ação de controle e envio ao quarto e último nó nomeado "i2cpwm_board_node", responsável apenas pela aplicação do sinal de controle aos motores.

Para embarcação do sistema proposto, primeiramente realizaram-se testes através da instalação de um *Ubuntu Server* no *Raspberry Pi* com a finalidade da utilização da versão mais recente do ROS2, porém enfrentou-se diversas dificuldades com relação ao acesso da câmera e dos módulos I2C de controle dos motores. Com base nisso devido a praticidade e material disponível, optou-se por utilizar uma imagem pré gravada para o *Raspberry Pi* da versão Kinetic do ROS, disponibilizada pela *Ubiquity Robotics*.

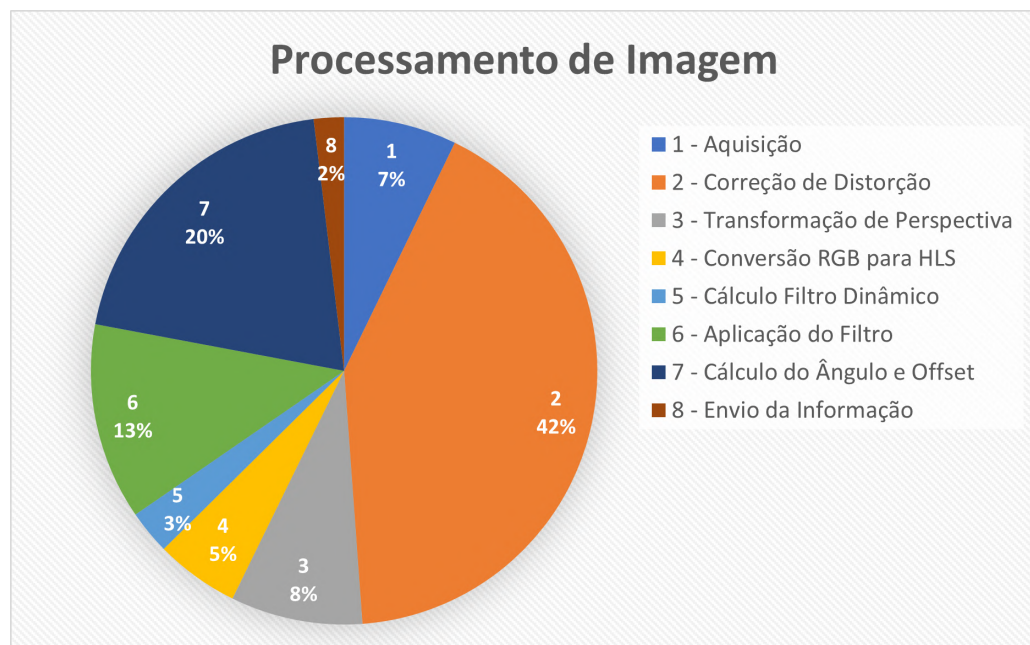
A estrutura final do software embarcado sem simplificações, contendo todos os nós e tópicos pode ser visualizada no Anexo A e está disponibilizada em (GITHUB, 2022).

4.2.2 Custo computacional e tempo de amostragem

Como já esperado durante o desenvolvimento do sistema, constatou-se que o nó de processamento de imagem é a maior limitação com relação ao tempo de processamento, resultando

na limitação do tempo de amostragem do controlador. Para melhor desempenho do sistema embarcado através da diminuição do tempo de processamento, decidiu-se a simplificação da pista através de uma linha central e por consequência a eliminação da etapa das janelas deslizantes descrita anteriormente. Na Figura 26 apresenta-se um estudo do tempo de processamento de cada etapa do módulo de processamento de imagem embarcado do sistema, onde o tempo total de processamento é de aproximadamente 99,2ms. Pode-se notar que as etapas de maior custo computacional são a correção da distorção, através das matrizes descritivas da câmera, a criação da máscara resultante, através da aplicação do filtro dinâmico e por fim o cálculo do ângulo e *offset* do veículo através dos polinômios descritivos da pista.

Figura 26 – Custo computacional do processamento de imagem



Fonte: Próprio Autor

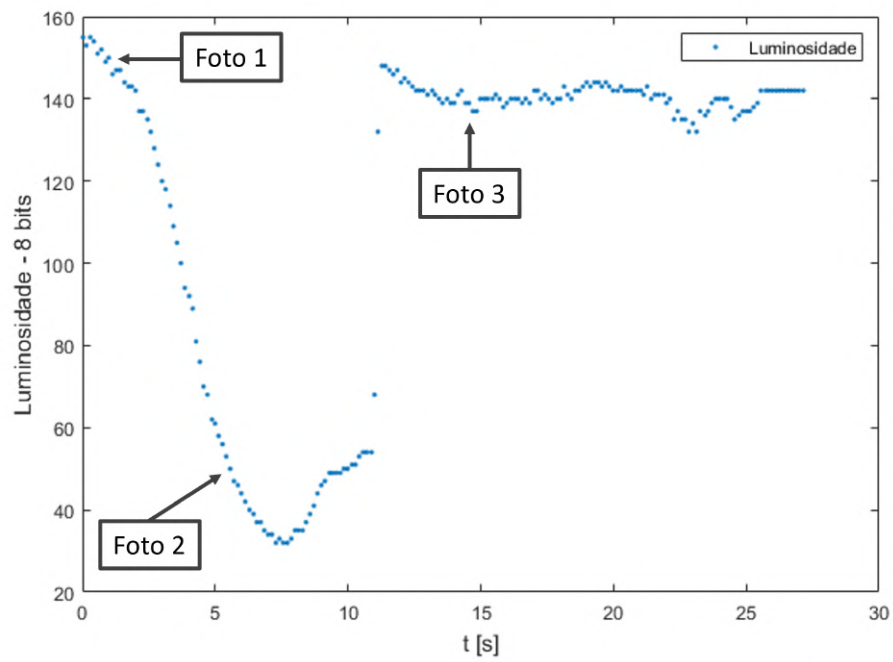
4.2.3 Aplicação do Filtro HLS Dinâmico

Durante os testes para aquisição dos resultados verificou-se uma grande variação com relação a luminosidade dos locais de uso, este desafio é crítico quando se trata de processamento de imagem, porém, verificou-se que com a aplicação do filtro dinâmico o problema é resolvido.

O gráfico da Figura 27 mostra o valor calculado do filtro dinâmico no canal de luminosidade em um único ensaio experimental, sendo que durante o ensaio provoca-se uma variação de amplitude de 45% causada pelo ambiente escuro e posteriormente pelo acendimento das luzes do local, conforme pode ser visto na Figura 28.

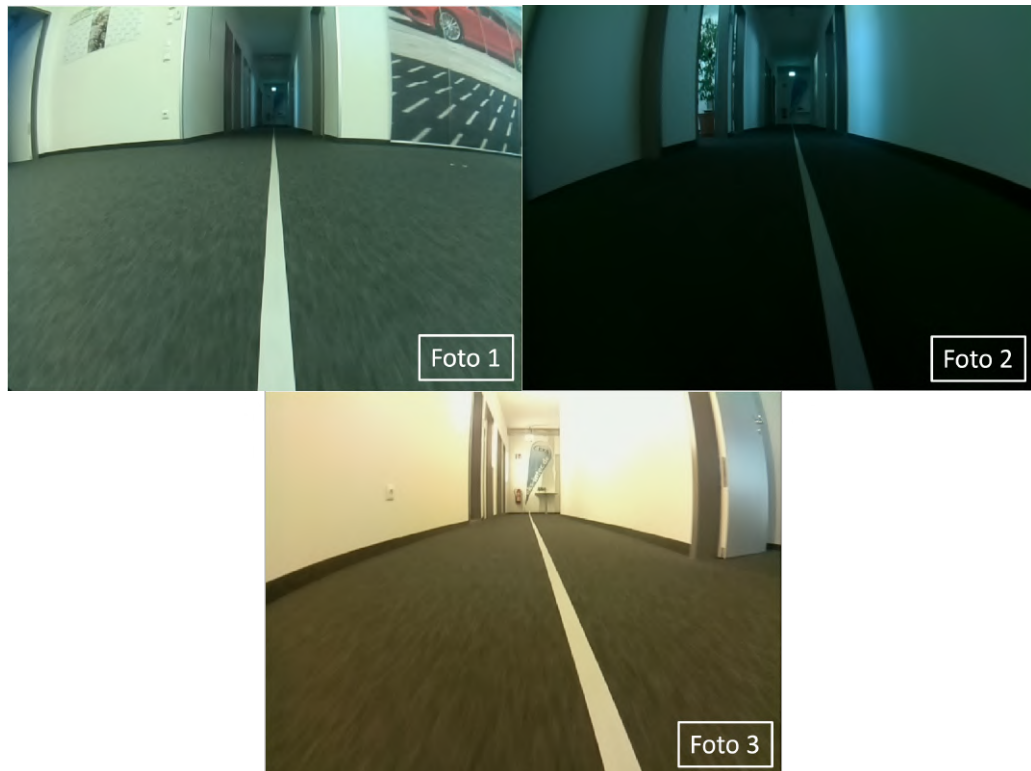
Tendo em vista o custo computacional baixo do cálculo do filtro, e o resultado gerado nas figuras anteriores, conclui-se que esta ferramenta se mostra eficaz, fazendo com que o processamento de imagem se torne mais robusto e adaptável.

Figura 27 – Filtro HLS dinâmico - gráfico



Fonte: Próprio Autor

Figura 28 – Filtro HLS dinâmico - fotos



Fonte: Próprio Autor

4.2.4 Resultado Prático

Para realização do resultado prático, criou-se um percurso de 25 metros em linha reta, conforme mostrado na Figura 29. De maneira virtual criaram-se os vetores de restrição de segurança, com três obstáculos em formato de triângulo. Para simulação do comportamento do motorista criou-se um comando através do teclado do computador, porém devido a limitação de dirigibilidade, optou-se por remover o comportamento do motorista, fazendo com que o sinal de controle seja gerado inteiramente pelo controlador, conforme mostrado no segundo ensaio das simulações da Seção 4.1 deste trabalho.

Figura 29 – Circuito de teste prático

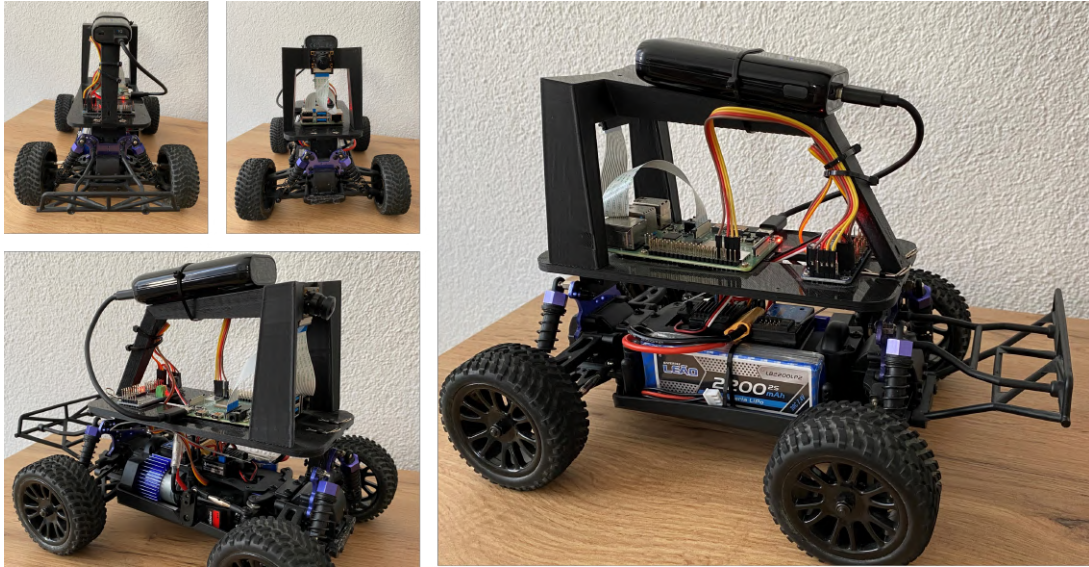


Fonte: Próprio Autor

O veículo final utilizado para os resultados práticos deste trabalho conforme descrito na Seção 3.1, pode ser visualizado na Figura 30.

Para realização do ensaio conectou-se um notebook com sistema operacional Ubuntu, através de comunicação SSH com o *Raspberry Pi* embarcado no veículo em miniatura. Através da conexão foi possível dar os comandos de início do ensaio e ativar a gravação dos sinais no formato "*bagfile*" que é suportado pelo *framework* ROS. Após a realização do ensaio e gravação dos sinais, foi possível utilizar a ferramenta "*plotjuggler*" para plotar gráficos e salvar os dados em formato csv, que posteriormente puderam ser importados em qualquer ferramenta de análise, como por exemplo Excel e Matlab. Para visualização das imagens da câmera, utilizou-se a aplicação "*rqt_image_view*", onde através dos tópicos publicados do sistema ROS, foi possível

Figura 30 – Carro em miniatura desenvolvido

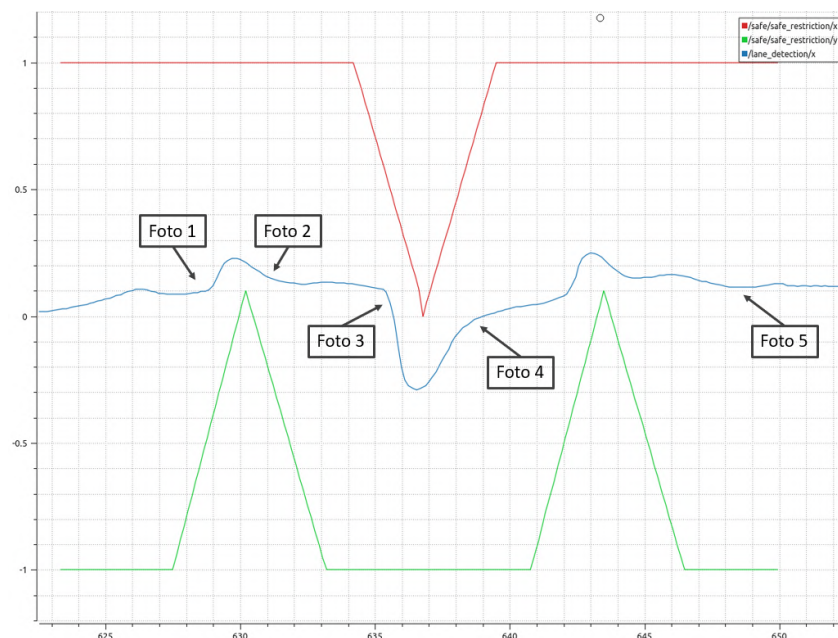


Fonte: Próprio Autor

analisar a imagem original da câmera e a imagem após o tratamento com a detecção da pista.

Para demonstração da plataforma criada, na Figura 31 mostra-se um ensaio com a marcação de cinco fotos, descritas pelas Figuras 32, 33, 34, 35 e 36, tiradas em momentos diferentes, para que se possa ter uma análise da dinâmica do sistema.

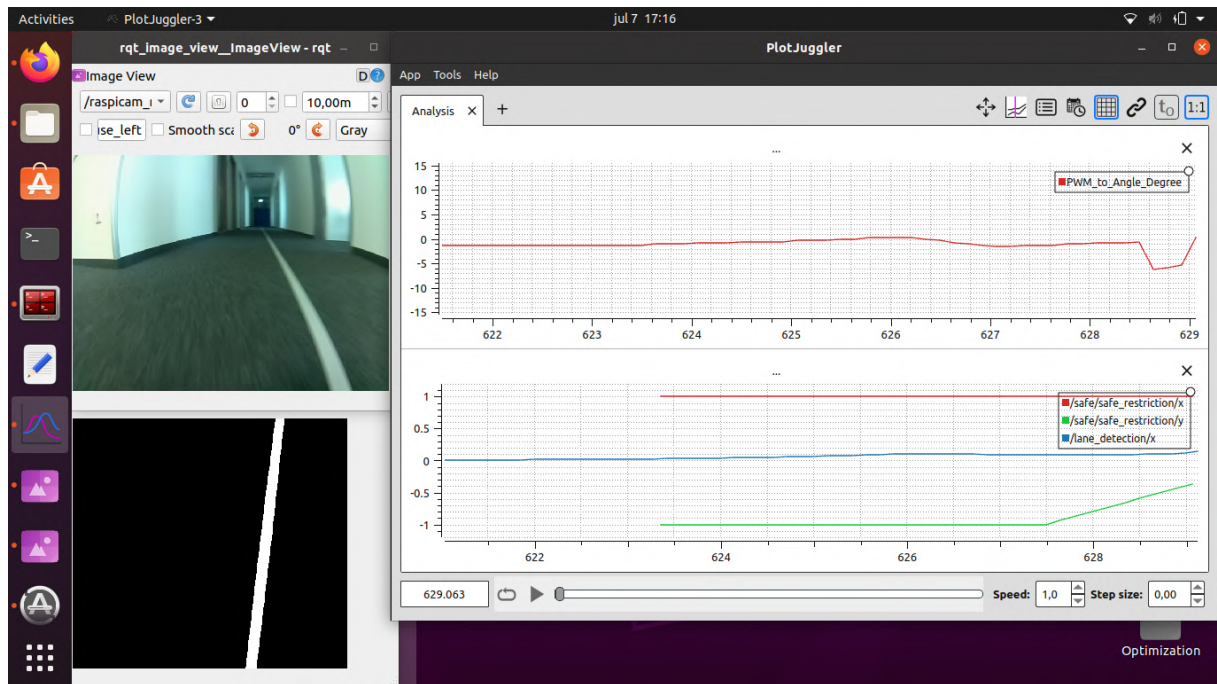
Figura 31 – Ensaio demonstrativo



Fonte: Próprio Autor

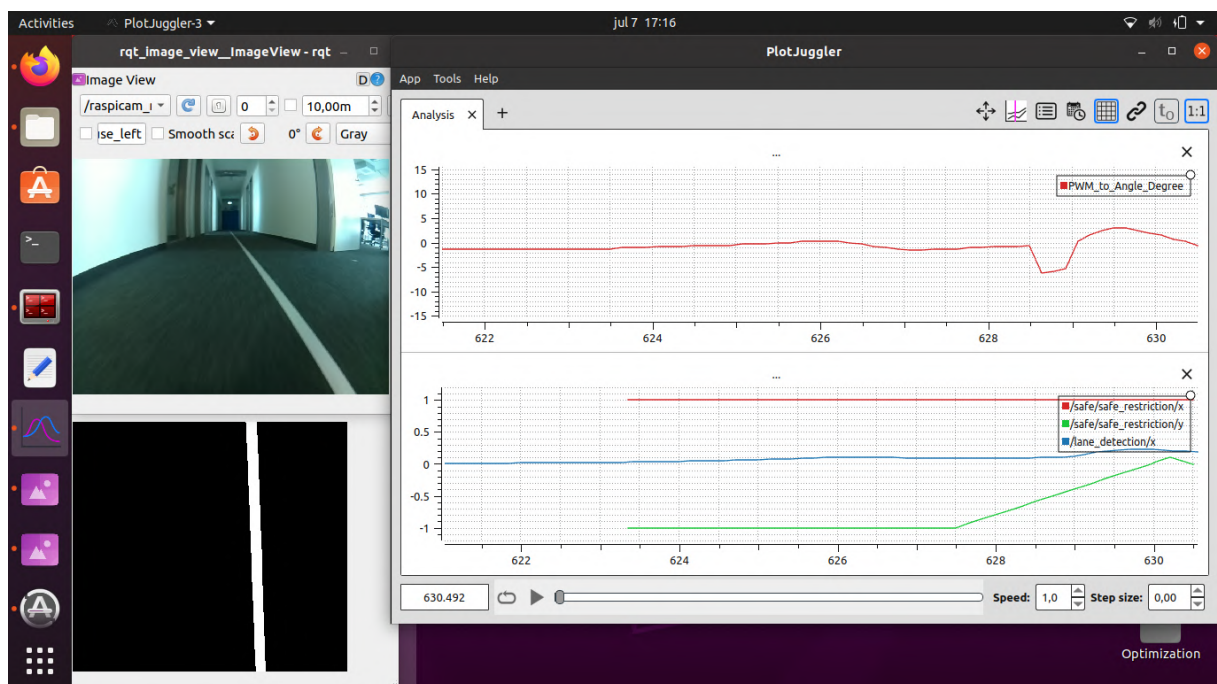
Para análise das fotos do ensaio da plataforma experimental, observa-se que no canto esquerdo superior apresenta-se a imagem original da câmera e no canto esquerdo inferior é possível visualizar a imagem resultante de detecção da pista. No canto direito da imagem é

Figura 32 – Ensaio demonstrativo - foto 1



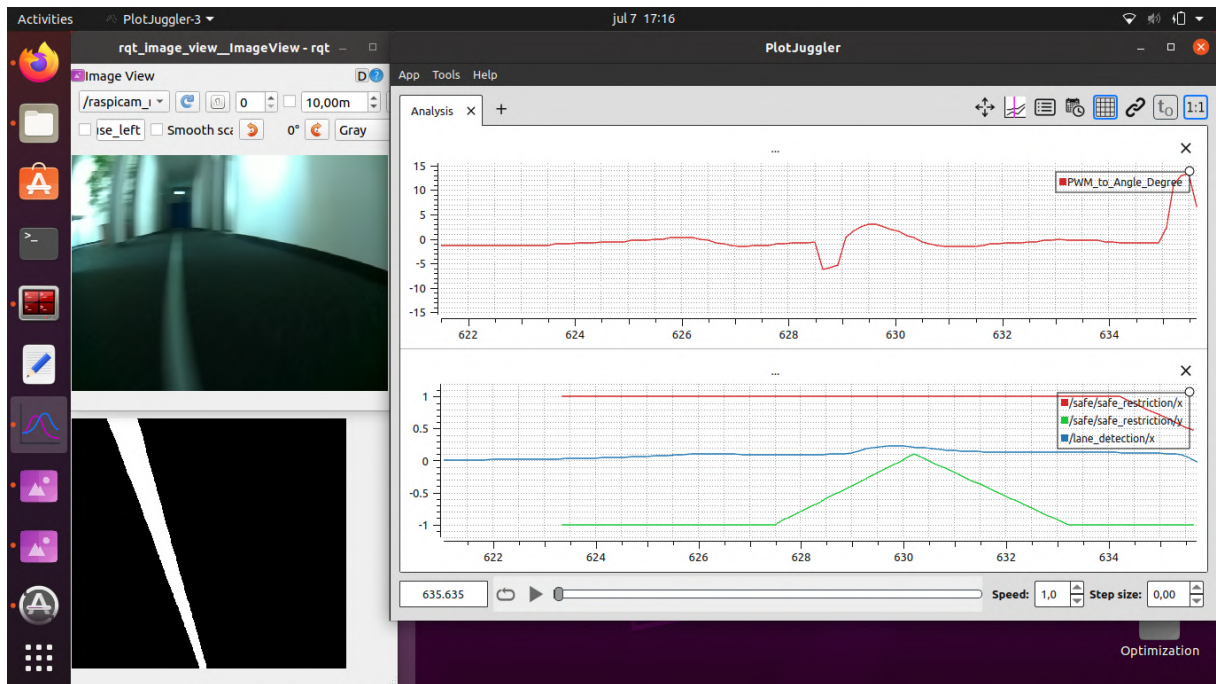
Fonte: Próprio Autor

Figura 33 – Ensaio demonstrativo - foto 2



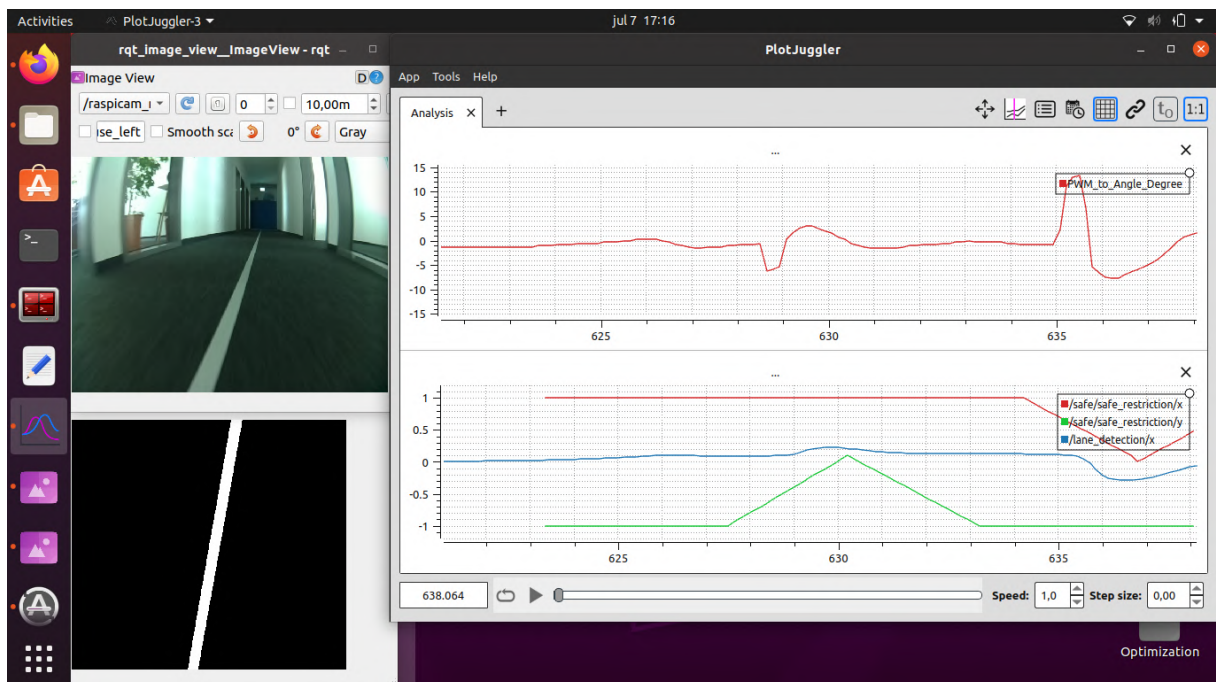
Fonte: Próprio Autor

Figura 34 – Ensaio demonstrativo - foto 3



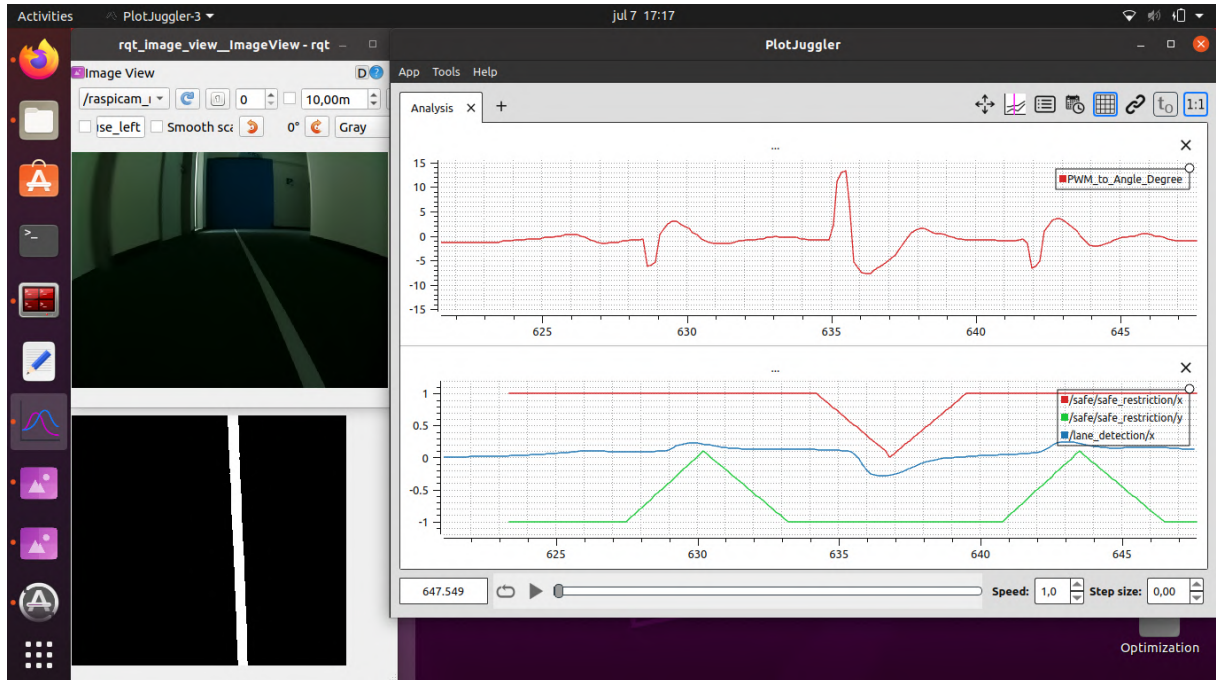
Fonte: Próprio Autor

Figura 35 – Ensaio demonstrativo - foto 4



Fonte: Próprio Autor

Figura 36 – Ensaio demonstrativo - foto 5



Fonte: Próprio Autor

possível visualizar dois gráficos, onde o gráfico superior descreve o sinal de controle em graus e o gráfico inferior, o mesmo apresentado na Figura 31, mostra em azul o trajeto do veículo descrito pelo *offset* com relação ao centro da linha, e em verde e vermelho as restrições de segurança criadas de maneira virtual.

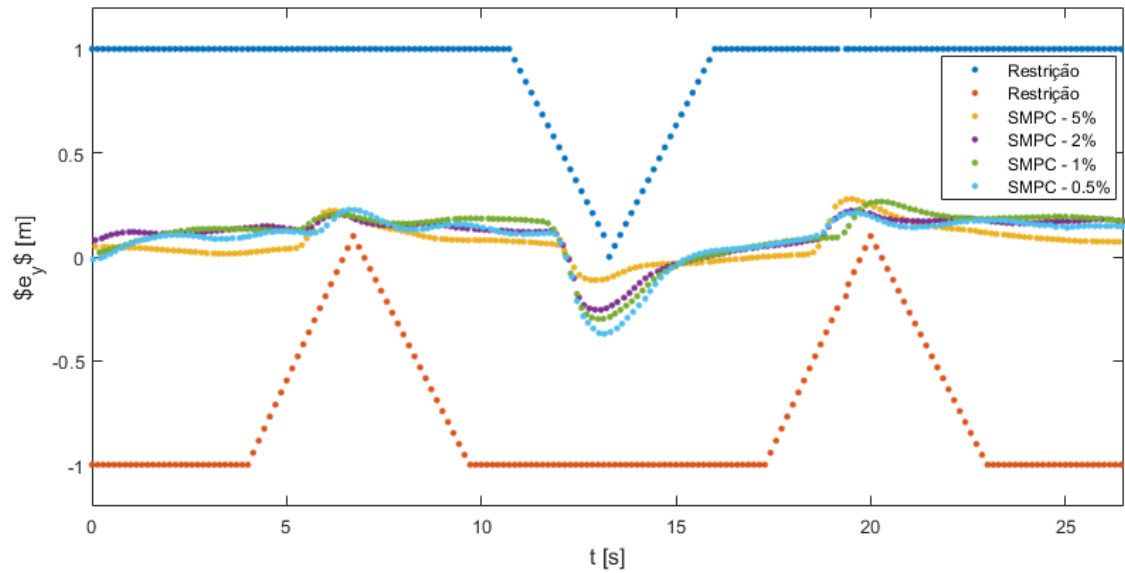
Neste Ensaio é possível verificar que obtêm-se um atraso com relação a imagem real e o resultado da imagem processada, isto se deve ao custo computacional do processamento de imagem que leva em média 100ms, como mostrado na Seção 4.2.2.

Para fins de análise da performance do controlador preditivo estocástico baseado em modelo, realiza-se um ensaio prático similar ao ensaio mostrado nas simulações, onde apresenta-se de forma simultânea o resultado de quatro ensaios, onde desenvolve-se o controlador preditivo estocástico baseado em modelo com diferentes porcentagens de aceitação de quebra de restrição.

O resultado da trajetória do veículo descrito pelo *offset* com relação ao centro da linha pode ser visto na Figura 37, através da análise do obstáculo superior, exibido na cor azul, pode-se verificar o comportamento esperado do sistema visto também nas simulações, onde com a diminuição da porcentagem aceitável de quebra das restrições o controlador aumenta a margem de segurança entre o obstáculo e o veículo, tornando assim menor a probabilidade de quebrar a restrição com a inserção de um sinal estocástico no sinal de controle.

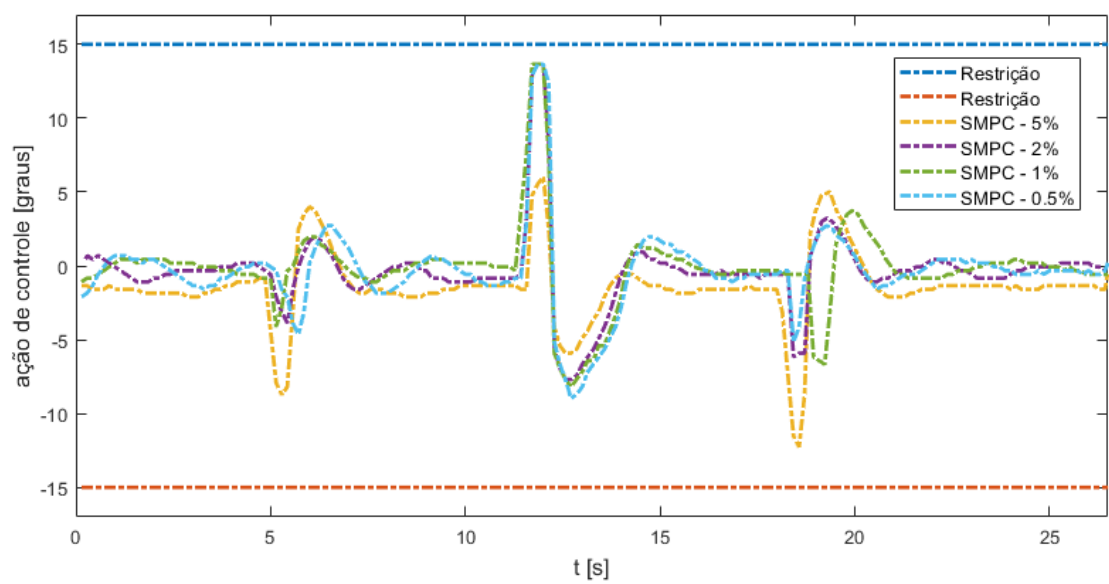
A ação de controle pode ser visualizada através da Figura 38, onde nota-se um aumento na amplitude das ações de controle conforme a diminuição da porcentagem de aceitação de quebra da restrição.

Figura 37 – Resultado do ensaio prático - sinal da trajetória



Fonte: Próprio Autor

Figura 38 – Resultado do ensaio prático - sinal de controle



Fonte: Próprio Autor

5 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho teve como principal objetivo, a aplicação prática de um Controlador Preditivo Estocástico Baseado em Modelo em um sistema de assistente de direção em carro em escala reduzida. Para tal aplicação desenvolveu-se a pesquisa e comissionamento de um protótipo em escala reduzida, baseado na praticidade e confiabilidade do sistema, contando com softwares de código aberto capazes de promover maior acessibilidade e compartilhamento de soluções com a comunidade.

No decorrer do trabalho enfrentaram-se diversos desafios técnicos de conhecimento para que a solução prática pudesse ser realizada, como manuseio de sistemas Linux, aplicação de sistemas ROS, linguagem de programação Python, técnicas de processamento de imagem, entre outros.

O protótipo resultante deste trabalho apresenta-se como uma boa ferramenta de testes para diferentes estratégias de controle, incluindo a possibilidade de estudos com inteligência artificial que estão em ascensão nas aplicações automotivas atualmente.

Através do exemplo de assistente de direção fez-se possível o estudo de controladores estocásticos através da adição de elementos não determinísticos no modelo do sistema. O controlador preditivo estocástico baseado em modelos, apresentou o seu valor no aumento da segurança do sistema. através da sua adição de conservadorismo ajustado através de sua porcentagem de quebra da restrição, baseando-se no valor crítico da distribuição normal do sinal estocástico.

Com relação a trabalhos futuros, é válida a remoção da simplificação com relação ao trajeto, onde através do modelo é possível variar a curvatura da pista e testar o sistema em condições mais extremas, além do aumento e variação de velocidade trazendo mais complexidade ao modelo.

Com relação ao carro em escala reduzida, acredita-se que é de suma importância a adição de um dispositivo de medição de velocidade, como um *encoder* acoplado a engrenagem do motor para que se possa medir e controlar tal grandeza.

Por fim, por todos estes pontos acima e pelos resultados apresentados, acredita-se que a proposta do trabalho apresentada através de seus objetivos foi cumprida, apresentando um estudo prático de controladores preditivos estocásticos e sistemas embarcados.

REFERÊNCIAS

- AMAZON. **amazon.com**. 2022. Disponível em: <<https://www.amazon.com/>>. Acesso em: 29 jun. 2022. Citado 6 vezes nas páginas 26, 27, 28, 29, 30 e 31.
- ATOMOCLAST. **Computer Vision/Perception: Structure From Motion**. 2018. Disponível em: <<https://realitybytes.blog/2018/05/10/computer-vision-perception-structure-from-motion/>>. Acesso em: 29 jun. 2022. Citado na página 33.
- AUTOCRYPT. **Camera, Radar and LiDAR: A Comparison of the Three Types of Sensors and Their Limitations**. 2021. Disponível em: <<https://autocrypt.io/camera-radar-lidar-comparison-three-types-of-sensors/>>. Acesso em: 07 jul. 2022. Citado na página 26.
- CAMACHO, Eduardo F.; ALBA, Carlo Bordons. **Model Predictive Control**. [S.l.]: Springer-Verlag London, 2007. Citado 2 vezes nas páginas 18 e 20.
- DONKEY CAR. **donkeycar.com | An opensource DIY self driving platform for small scale cars**. 2020. Disponível em: <<https://www.donkeycar.com/>>. Acesso em: 29 jun. 2022. Citado 2 vezes nas páginas 26 e 27.
- FARINA, Marcello; GIULIONI, Luca; SCATTOLINI, Riccardo. Stochastic linear model predictive control with chance constraints – a review. **Journal of Process Control**, v. 44, p. 53 – 67, 2016. ISSN 0959-1524. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0959152416300130>>. Citado na página 22.
- GITHUB. **Where the world builds software**. 2022. Disponível em: <<https://github.com/guigakn/stochastic-driver-assistance>>. Acesso em: 26 jul. 2022. Citado 2 vezes nas páginas 33 e 46.
- Gray, A. et al. Stochastic predictive control for semi-autonomous vehicles with an uncertain driver model. In: **16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)**. [S.l.: s.n.], 2013. p. 2329–2334. Citado na página 38.
- GVR. **Industrial Automation And Control Systems Market Size, Share Trends Analysis Report By Component (Industrial Robots, Control Valves), By Control System (DCS, PLC, SCADA), By Vertical, By Region, And Segment Forecasts, 2021 - 2028**. 2019. Disponível em: <<https://www.grandviewresearch.com>>. Acesso em: 1 jul. 2020. Citado na página 17.
- JHANG, Jyun-Hao; LIAN, Feng-Li. An autonomous parking system of optimally integrating bidirectional rapidly-exploring random trees* and parking-oriented model predictive control. **IEEE Access**, v. 8, p. 163502–163523, 2020. Citado na página 17.
- JHANG, Jyun-Hao; LIAN, Feng-Li. An autonomous parking system of optimally integrating bidirectional rapidly-exploring random trees* and parking-oriented model predictive control. **IEEE Access**, v. 8, p. 163502–163523, 2020. Citado na página 23.
- KAWASAKI, N.; KIENCKE, U. Standard platform for sensor fusion on advanced driver assistance system using bayesian network. In: **IEEE Intelligent Vehicles Symposium, 2004**. [S.l.: s.n.], 2004. p. 250–255. Citado na página 26.

LAURENSE, Vincent A.; GERDES, J. Christian. Long-horizon vehicle motion planning and control through serially cascaded model complexity. **IEEE Transactions on Control Systems Technology**, v. 30, n. 1, p. 166–179, 2022. Citado na página 17.

LEE, Jin-woo. **Model based predictive control for automated lane centering/changing control systems**. 2012. Disponível em: <<https://patents.google.com/patent/US8190330B2/en>>. Acesso em: 25 jun. 2020. Citado na página 17.

LEFÈVRE, Stéphanie; CARVALHO, Ashwin; BORRELLI, Francesco. A learning-based framework for velocity control in autonomous driving. **IEEE Transactions on Automation Science and Engineering**, v. 13, n. 1, p. 32–42, 2016. Citado na página 17.

MathWorks. **Developing Algorithms for ADAS Systems with MATLAB and Simulink**. 2020. Disponível em: <<https://bit.ly/3SkIeHa>>. Acesso em: 25 jun. 2020. Citado na página 18.

MERCADO LIVRE. **Mercado Livre**. 2022. Disponível em: <<https://www.mercadolivre.com.br/>>. Acesso em: 29 jun. 2022. Citado 2 vezes nas páginas 30 e 31.

MESBAH, A. Stochastic model predictive control: An overview and perspectives for future research. **IEEE Control Systems Magazine**, v. 36, n. 6, p. 30–44, Dec 2016. ISSN 1941-000X. Citado 2 vezes nas páginas 18 e 21.

RAJESH, Rajamani. **Vehicle Dynamics and Control**. [S.l.]: Springer, 2006. ISBN 9780387263960. Citado 3 vezes nas páginas 23, 24 e 25.

ROS. **ROS.org | Powering the world's robots**. 2019. Disponível em: <<http://www.ros.org/>>. Acesso em: 20 mar. 2019. Citado 2 vezes nas páginas 30 e 32.

THINGIVERSE. **Donkey Chassis and Adapters - Current**. 2017. Disponível em: <<https://www.thingiverse.com/thing:2566276>>. Acesso em: 26 jul. 2022. Citado na página 30.

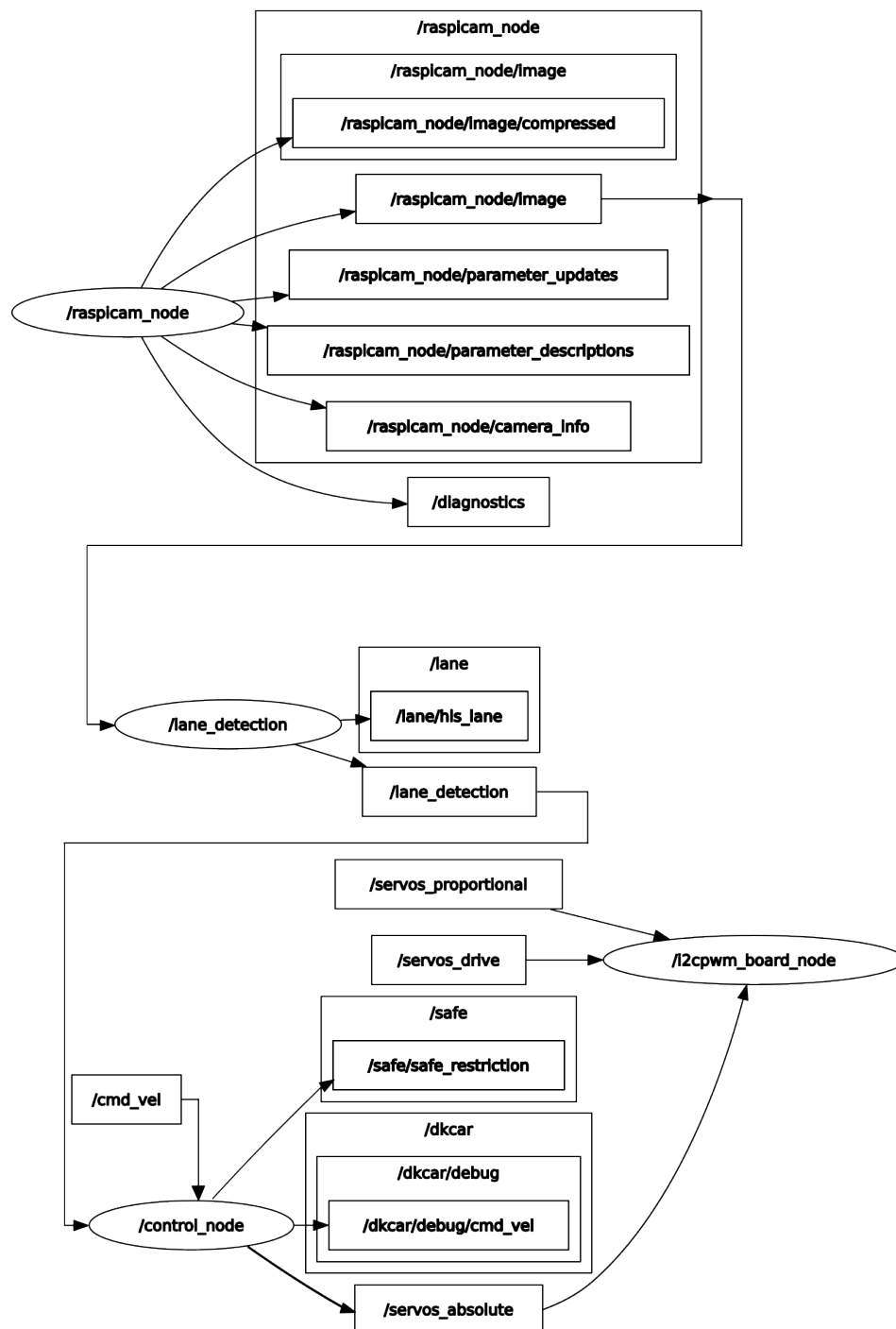
TUCKER, M.; HEENAN, A.; BUCHANAN, A. Real time embedded sensor fusion for driver assistance. In: **Proceedings. 2005 IEEE Intelligent Transportation Systems, 2005**. [S.l.: s.n.], 2005. p. 596–601. Citado na página 26.

VTPI. **Autonomous Vehicle Implementation Predictions**. 2022. Disponível em: <<https://www.vtpi.org/avip.pdf>>. Acesso em: 29 jun. 2022. Citado na página 17.

YAMAUCHI, Yohei; SAITO, Mitsuyuki; ONO, Takahiko. Adaptive identification method of vehicle modeling according to the fluctuation of road and running situation in autonomous driving. In: **2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)**. [S.l.: s.n.], 2019. p. 1255–1260. Citado na página 23.

ZHANG, Z. A flexible new technique for camera calibration. **IEEE Transactions on pattern analysis and machine intelligence**, v. 22, n. 11, p. 1330–1334, Nov 2000. ISSN 1939-3539. Citado na página 32.

ANEXO A – ESTRUTURA ROS COMPLETA



Fonte: Próprio Autor