



UNIVERSIDADE FEDERAL DA BAHIA

DISSERTAÇÃO DE MESTRADO

**Estimadores Visuais para o Seguimento de Caminhos Baseado
em NMPC.**

Iago José Pattas Bastos Franco

Programa de Pós-Graduação em Engenharia Elétrica

Salvador
2020

IAGO JOSÉ PATTAS BASTOS FRANCO

**ESTIMADORES VISUAIS PARA O SEGUIMENTO DE
CAMINHOS BASEADO EM NMPC.**

Esta Dissertação de Mestrado foi apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia, como requisito parcial para obtenção do grau de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. André Gustavo Scolari Conceição

Co-orientador: Prof. Dr. Tiago Trindade Ribeiro

Salvador
2020

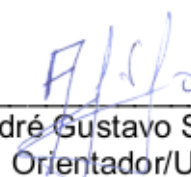
Iago José Pattas Bastos Franco

"Estimadores Visuais para o Seguimento de Caminhos Baseado em NMPC".

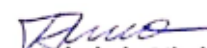
Dissertação apresentada à Universidade Federal da Bahia, como parte das exigências do Programa de Pós-Graduação em Engenharia Elétrica, para a obtenção do título de Mestre.

APROVADA em: 10 de Agosto de 2020.

BANCA EXAMINADORA



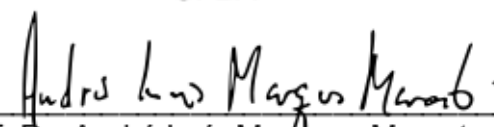
Prof. Dr. André Gustavo Scolari Conceição
Orientador/UFBA



Prof. Dr. Tiago Trindade Ribeiro
UFBA



Prof. Dr. Paulo César Machado de Abreu Farias
UFBA



Prof. Dr. André Luís Marques Marcato
UFJF

AGRADECIMENTOS

Aos meus pais, irmã e toda a família, pelo amor, incentivo e apoio.

À Universidade, por me proporcionar um ambiente amigável e um corpo docente sensacional. Ao meu orientador André Scolari pela orientação, apoio e confiança. Ao meu co-orientador Tiago Ribeiro pela dedicação, paciência e compreensão. Aos meus colegas do mestrado, Rodrigo, Márcio e Afonso pela amizade e empenho nos trabalhos e disciplinas. Aos companheiros do Laboratório de Robótica pelas orientações e discussões edificantes.

Ao SEPIN/MCTI e ao Programa de Pesquisa e Inovação Horizonte 2020 da União Europeia que, por meio do Contrato de Doação Nº 777096, forneceram financiamento para este projeto.

Aos meus colegas do SENAI, e em especial ao meu orientador Tiago Cahyba, agradeço pela amizade, compreensão e paciência ao longo do último ano. À Ravindra e ao pessoal do espaço Bella Ananda pelas práticas e lições ensinadas.

À todos que, direta ou indiretamente, contribuíram para essa caminhada, fica aqui meu agradecimento de coração, muito obrigado.

RESUMO

Este trabalho tem como objetivo principal estudar e desenvolver sistemas para detecção de caminhos e extração de parâmetros diretamente do plano da imagem, através de técnicas de visão computacional, para alimentar controladores para seguimento de caminhos visuais baseados em NMPC. Detectar um caminho para realizar o controle baseado em imagem requer um processamento eficiente que consiga extrair parâmetros de forma precisa mesmo na presença de inconvenientes no ambiente de navegação, como: falhas no caminho, pontos de luz ou sombra, má condição de luminosidade, ruídos na imagem capturada e pisos de coloração não homogênea. Tais problemas comprometem a eficácia do controlador pois afetam a maneira como os parâmetros são extraídos.

Para alcançar tais objetivos, o espaço de cores HSV é utilizado para realizar a segmentação frente às diferenças de luminosidade, contudo, para extrair os parâmetros, são propostos 3 métodos. Num primeiro método, utiliza-se uma extração baseada em regiões conectadas (*blobs*), onde a partir do maior *blob* os parâmetros do caminho são estimados. Num segundo método, visa-se melhorar a robustez do sistema visual, visto que no primeiro método não houve uma função que representasse o caminho a ser seguido, o que causa imprecisão nos cálculos dos parâmetros. Para isso, é inserido o Método dos Mínimos Quadrados como etapa de estimação do caminho. Num terceiro método, visa-se melhorar a sensibilidade na estimação do caminho feita no segundo método e consequentemente os problemas trazidos, para isso, é introduzido uma técnica robusta de estimação de modelos, o método RANSAC.

Para avaliar a performance destes métodos, os resultados são apresentados em duas etapas que corresponderam às fases de desenvolvimento. Na primeira fase foram validados e estudados o primeiro e segundo método. Na segunda fase, foram validadas as 3 estratégias. A partir dos resultados experimentais, a superioridade do sistema baseado em RANSAC em comparação aos outros dois, é demonstrada, principalmente em relação a problemas no ambiente de navegação, como: trajetos defeituosos, ruídos e más condições de iluminação.

Palavras-chave: Visão Computacional, NMPC, HSV, Seguimento de Caminhos Visuais, Método dos Mínimos Quadrados, RANSAC.

ABSTRACT

The main goal of this work is to study and develop systems for path detection and parameters extraction directly from the image plane, using computer vision techniques, to feed controllers to perform NMPC-based Visual Path Following. Detecting a path to perform image-based control requires efficient processing that can extract parameters precisely even in the presence of inconveniences in the navigation environment, such as path failures, spots of lights or shadows, poor lighting conditions, noise in the image captured, and floors with non-homogeneous colors. Such problems compromise the controller's effectiveness because they affect the way the parameters are extracted.

To achieve these goals, the HSV color space is used to perform segmentation in front of differences in lighting conditions, however, to extract the parameters, 3 methods are proposed. In a first method, an extraction based on connected regions (*blobs*) is used, wherefrom the largest *blob* the path parameters are estimated. In a second method, the aim is to improve the robustness of the visual system, since in the first method no function represented the path to be followed, which causes inaccuracy in the parameter calculations. For this, the Least Squares Method is inserted as a step for path estimation. In a third method, the aim is to improve the sensitivity in estimating the path by the second method and consequently the problems brought from it, for this, a robust model estimation technique, the RANSAC method, is introduced.

To evaluate the performance of these methods, the results are presented in two stages that corresponded to the development phases. In the first phase, the first and second methods were validated and studied. In the second phase, the 3 strategies were validated. From the experimental results, the superiority of the RANSAC-based system in comparison to the other two is demonstrated, especially concerning problems in the navigation environment, such as defective routes, noise, and poor lighting conditions.

Keywords: Computer Vision, NMPC, HSV, Visual Path Following, Least Square Method, RANSAC.

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Objetivos	2
1.1.1 Objetivos Específicos	3
1.1.2 Resultados Esperados	3
1.2 Motivação	3
1.3 Estrutura da dissertação	4
Capítulo 2—Formalização do Problema	5
Capítulo 3—Sistema de Detecção Visual	12
3.1 Espaço de Cores	12
3.1.1 RGB	12
3.1.2 HSV	14
3.2 Ajuste de Curvas	16
3.2.1 Método dos Mínimos Quadrados	16
3.2.2 RANSAC	17
3.3 Métodos Propostos	21
Capítulo 4—Estudo de Caso	25
4.1 O Robô Husky e a Plataforma ROS	26
4.2 Cenários de Teste	27
4.3 Pré-Configuração dos Experimentos	27
Capítulo 5—Resultados Experimentais	30
5.1 Performance no Primeiro Cenário	30
5.1.1 Segmentação do Caminho	30
5.1.2 Representação da Função Caminho	34
5.1.3 Resultados Experimentais	36
5.2 Performance no Segundo Cenário	40
5.2.1 Segmentação do Caminho	40
5.2.2 Representação da Função Caminho	43
5.2.3 Resultados Experimentais	44

Capítulo 6—Conclusão	51
6.1 Trabalhos Publicados	52

LISTA DE FIGURAS

2.1	Seguimento de Caminho Visual Operando Normalmente	6
2.2	Alguns problemas encontrados no modelo de seguimento de caminhos visuais.	8
2.3	Problemas encontrados na abordagem original proposta por (Ribeiro; Conceição, 2019)	9
2.4	Segmentação utilizando Otsu nos cenários analisados.	10
2.5	Cálculo de curvatura utilizado.	11
3.1	Cubo RGB (Su, 2016)	13
3.2	Sensibilidade Espectral x Comprimento de Onda. Espectro de cores obtidos a partir da combinações das cores primárias <i>vermelho</i> = 700nm, <i>verde</i> = 546.1nm e <i>azul</i> = 435.8nm. (Szeliski, 2011)	14
3.3	Cone de Cores HSV	15
3.4	Ajuste Através do Método dos Mínimos Quadrados (Fischler; Bolles, 1981)	18
3.5	Exemplo da estimação de uma função afim e o que o método considera como <i>Inliers</i> e <i>Outliers</i> . Adaptado de (Heydarzadeh et al., 2017).	19
3.6	Fluxograma do método RANSAC para estimação de uma função quadrática.	20
3.7	Segmentação utilizando espaço de cores HSV da Fig. 2.4(c).	22
3.8	Estimação dos caminhos (MMQ vs RANSAC) da Fig. 2.4(c).	23
3.9	Heurística utilizada e Acoplamento entre θ_r e Z	24
4.1	Materiais Utilizados	25
4.2	Dimensões do Robô Husky	26
4.3	Ambientes de Teste.	29
5.1	Conversão RGB-HSV em imagem do 1 ^a Cenário.	31
5.2	Região com alguns pontos acromáticos da Figura 5.1(a), onde os valores dos pixels RGB são mostrados na cor que os correspondem.	31
5.3	Separação dos canais HSV das imagens da Figura 5.1(b).	32
5.4	Imagem do 1 ^o cenário expostas ao ambiente com baixa iluminação.	32
5.5	Separação dos canais HSV das imagens da Figura 5.4	33
5.6	Imagem da Fig.5.4(b) convertida de volta a RGB, com o adicional de brilho para destacar a cor amarela detectada pelo canal H_c	33
5.7	Segmentação da Figura 5.1(a) na calibração do Método 1.	34
5.8	Segmentações na calibração do Método 2.	35
5.9	Análise do Brilho(V_c) e Saturação(S_c) nas segmentações das imagens no ambiente com baixa iluminação da Figura 5.4(b)	35

5.10	Funções que representam o caminho no 1 ^o Cenário frente às diferentes condições de iluminação.	36
5.11	Caminho Reto. Nesta curva, foram gerados os seguintes coeficientes: $a_p = 3.14928e^{-8}$, $b_p = 1.93633e^{-6}$ e $c_p = 93.5$	36
5.12	Comparação dos Métodos 1 e 2 no Primeiro Cenário.	37
5.13	Algumas das cenas vistas no cenário de teste.	38
5.14	Fontes de Luz Artificial no Ambiente com pouca iluminação.	39
5.15	Segmentação com Fonte de Luz Próxima ao Robô no Método 1. Nota-se que a curvatura calculada (pequeno círculo azul) é completamente distorcida com relação ao caminho real.	40
5.16	Conversão RGB-HSV em imagem do 2 ^a Cenário.	40
5.17	Instabilidade no espaço de cores HSV perto de pontos acromáticos.	41
5.18	Separação dos canais HSV das imagens da Figura 5.16(b).	41
5.19	Conversão RGB-HSV em <i>frame</i> do 2 ^a Cenário sob baixas condições de luminosidade.	42
5.20	Separação dos canais HSV das imagens da Figura 5.19(c).	42
5.21	Segmentações na calibração para o 2 ^o cenário.	43
5.22	Comparação Mínimos Quadrados vs RANSAC em ambiente com boa iluminação no cenário 2.	44
5.23	Estimação dos caminhos (MMQ vs RANSAC) em condições de baixa luminosidade no cenário 2.	44
5.24	Comparação das abordagens em ambiente com boa iluminação.	45
5.25	Alguns dos inconvenientes encontrados pelos três Métodos.	46
5.26	Comparação dos Métodos - Navegação ao longo do cenário com baixa luminosidade.	48
5.27	Comparação das abordagens em ambiente com má condição de iluminação.	49
5.28	Estimação dos caminhos (MMQ vs RANSAC) onde o Método 2 pára.	50

LISTA DE TABELAS

5.1	Análise Quantitativa	39
5.2	Análise Quantitativa até a primeira falha no caminho (<i>gap</i>).	47
5.3	Análise Quantitativa do Experimento Completo.	47

LISTA DE SIGLAS

CIE *Commission internationale de l'éclairage*

GPS *Global Position System*

HSI *Hue Saturation Intensity*

HSV *Hue Saturation Value*

IAE *Integral Absolute Error*

IMU *Inertial Measurement Unit*

LIDAR *Light Detection And Ranging*

MMQ *Método dos Mínimos Quadrados*

NMPC *Nonlinear Model Predictive Control*

RANSAC *Random Sample Consensus*

RGB *Red Green Blue*

ROS *Robot Operating System*

SVD *Single Value Decomposition*

TV *Total Variation*

UGV *Unmanned Ground Vehicle*

LISTA DE SÍMBOLOS MATEMÁTICOS

Γ	Caminho de referência
λ	Comprimento de Onda
$\mathbf{Q}$	Matriz de ponderação dos erros de estado
$\mathbf{R}$	Matriz de ponderação dos esforços de controle
$\mathbf{u}_e$	Entrada de Controle
$\mathbf{x}_e$	Vetor de estados
$\mathcal{U}$	Conjunto de entradas factíveis
$\mathcal{X}$	Conjunto de estados factíveis
$\ \cdot\ $	Norma Euclidiana
ω	Velocidade Angular
σ	Desvio máximo permitido para ser considerado <i>inlier</i>
θ_{fov}^v	Campo vertical do ângulo de visão vertical
θ_{cam}	Ângulo do eixo focal da câmera
θ_e	Erro de orientação
θ_r	Orientação de referência
$\{SF\}$	Sistema de coordenadas <i>Serret-Frenet</i>
$\{w\}$	Sistema de coordenadas do mundo
a_p, b_p, c_p	Coefficientes da função caminho
$c(s)$	Curvatura do Caminho
$F(\cdot)$	Função custo de estágio
H	Horizonte Visual
h_c	Altura da câmera

H_c, S_c, V_c Componentes do espaço de cores HSV

$J(\cdot)$ Função objetivo

k_θ Constante de calibração angular

k_h Constante de calibração do horizonte visual

k_z Constante de calibração linear

$L(\lambda)$ Espectro de Luz

$\max(\cdot)$ Função para a determinação do máximo elemento de um vetor

$\min$ Minimize (Problema)

$\min(\cdot)$ Função para a determinação do mínimo elemento de um vetor

N Número de iterações (RANSAC)

p Probabilidade de ser *inlier*

r Resíduo (erro)

s Tamanho do Caminho

$S_B(\lambda)$ Sensibilidade espectral do azul

$S_G(\lambda)$ Sensibilidade espectral do verde

$S_R(\lambda)$ Sensibilidade espectral do vermelho

T_c Horizonte de controle

T_p Horizonte de predição

T_s Período de amostragem

v Velocidade Linear

x_i^0 Coordenadas horizontal do pixel no centro da imagem

x_i Pixels horizontal no caminho

x_r, y_r, z_r Coordenadas do sistema de referência do robô

y_i^0 Coordenada vertical do pixel no centro da imagem

y_i Pixels vertical no caminho

Z Deslocamento Lateral

INTRODUÇÃO

Atualmente, as tarefas de detecção e navegação por caminhos visuais, usando técnicas de visão computacional, tem sido amplamente utilizada em diferentes áreas como: Industrial (Gorbunov et al., 2018), Transportes (Tianqi, 2017), Agricultura (Arakeri et al., 2017) e Automotivo (Lu et al., 2014; Mammeri; Boukerche; Lu, 2014). Tais técnicas, tem o objetivo de tornar as soluções menos sensíveis a erros e falhas advindas do ambiente inserido e do próprio sistema de captura visual.

Detectar linhas de faixa que são a referência de um caminho, não é uma tarefa trivial. Eventualmente, o sistema visual encontra alguns inconvenientes, no caminho de referência, em relação a: trechos faltantes (*gaps*), deformidades, baixo contraste, ruído ou luminosidade do ambiente (dia, noite, chuva, pouca iluminação, pontos de sombra ou luz).

É possível encontrar na literatura algumas técnicas de segmentação baseada em cores. Tais técnicas utilizam transformações baseadas em espaço de cores como HSV (*Hue Saturation Value*) e HSI (*Hue Saturation Intensity*), com intuito de aumentar a robustez da segmentação frente a diferenças de iluminação no ambiente (Mammeri; Boukerche; Tang, 2016; Tran et al., 2010). Estes métodos pressupõem que a faixa a ser extraída possui cor bem definida, sendo geralmente branca ou amarela.

É comum ainda encontrar técnicas que se utilizam do contraste entre a faixa e o *background*, sendo possível a aplicação de ferramentas de detecção baseadas em contorno (utilizando filtros de detecção por borda (Lee; Cho, 2009)).

Em diversos trabalhos são encontrados também, técnicas baseadas em transformação de perspectiva (Wang; Hong; Gong, 2018; Yan Jiang; Feng Gao; Guoyan Xu, 2010), que utilizam as imagens capturadas pelas câmeras para gerar novas imagens com uma perspectiva de uma visão de topo (*top view*). No geral, é possível observar que esta técnica é aplicada em vários trabalhos relacionados à detecção de faixas de trânsito em estradas e rodovias, por isso, seu uso está relacionado à característica da aplicação, como visto em (Wang; Dahnoun; Achim, 2012), que pressupõem que as faixas não mudam muito de posição, largura e cor no plano da imagem com o passar do tempo.

Numa etapa após a segmentação do caminho, o passo tomado por muitos estudos é a estimação de uma função que represente o caminho identificado. De acordo com (Lu et al., 2014), os autores tentam encontrar um equilíbrio para representar os caminhos de forma precisa sem que haja demasiado custo para estimá-los. Dentre as abordagens é possível deparar-se com modelos simplistas que utilizam transformada Hough (Borkar; Hayes; Smith, 2009; Jang; Baek; Park, 2014) e consideram o trajeto sempre como uma linha reta. Há também aqueles baseados em cenários de rodovias e estradas, que estimam as faixas de trânsito com modelos geométricos hiperbólicos (López et al., 2010; Du; Tan; Htet, 2015), modelos parabólicos (Huang et al., 2009; McCall; Trivedi, 2006), linear-parabólico (Jung; Kelber, 2005) e modelos semi-paramétricos como splines (Kim, 2008; Wang; Teoh; Shen, 2004).

Para os modelos ditos paramétricos (linhas, parábolas, hipérboles), muitos trabalhos utilizam, para estimar os parâmetros das funções, o Método dos Mínimos Quadrados (MMQ). Com o intuito de melhorar a estimação mesmo na presença de ruídos da imagem, alguns autores utilizam o método iterativo *Random Sample Consensus* (RANSAC) junto ao MMQ (Huang et al., 2009; Borkar; Hayes; Smith, 2009). O método alvo usado nesta dissertação baseia-se em modelos parabólicos, pois de acordo com (Lu et al., 2014), estes são considerados um meio termo entre precisão e custo.

Após a obtenção dos parâmetros dos caminhos de referência através de um dos métodos citados, vem se tornando cada vez mais necessário o uso de modelos representativos e controladores visuais com características robustas, pois estes, permitem que se aplique um controle computacionalmente eficiente feito a partir do plano da imagem. Tendo isso, emergem os controladores preditivos, que através do princípio do horizonte deslizante, tornam-se adequados para o problema de seguir caminhos arbitrários, mesmo que estes estejam sujeitos a restrições e imperfeições físicas. A versão não linear deste tipo de controlador foi usada para resolver tais desafios de seguimento de caminho (Faulwasser et al., 2009; Castano; Tan, 2019), e cuja implementação foi possível através do uso direto e indireto de sistemas de localização, que têm como desvantagem um alto custo computacional quando aplicado à grandes plataformas em ambientes dinâmicos e não estruturados.

Com isso, este trabalho pretende contribuir com o desenvolvimento de algoritmos para a detecção e extração de parâmetros que representam um trajeto a se seguir, com utilização voltada a veículos autônomos de pequeno e grande porte. Com estes algoritmos, é possível extrair informações úteis como, orientação, posição e curvatura do caminho, que serão utilizados para alimentar controladores preditivos, mais especificamente, o controlador proposto por (Ribeiro; Conceição, 2019). Estas abordagens visam aumentar o índice de robustez do trabalho de (Ribeiro; Conceição, 2019) frente às diversas condições de iluminação no ambiente e deformidades no caminho.

1.1 OBJETIVOS

O objetivo deste trabalho é estudar e desenvolver sistemas para detecção de caminhos visuais com a finalidade de estimar parâmetros e utilizá-los em controladores preditivos não lineares (NMPC).

1.1.1 Objetivos Específicos

Os objetivos específicos são:

- Desenvolver um sistema de detecção de caminhos visuais, com segmentação robusta a luminosidade, baseado no espaço de cores HSV;
- Estimar uma função matemática que represente o caminho segmentado no espaço 2D, utilizando técnicas baseadas no Método dos Mínimos Quadrados e RANSAC;
- Extrair parâmetros de controle a partir dos resultados obtidos pelo algoritmo de visão e estimação das funções matemáticas, através de uma heurística, para alimentar o controlador NMPC;
- Implementar as abordagens, utilizando a plataforma ROS (*Robot Operation System*);
- Validar os métodos propostos no robô terrestre Husky UGV (*Unmanned Ground Vehicle*) em condições boas e ruins de iluminação;
- Analisar os resultados dos métodos junto ao controlador usando métricas como IAE (*Integral Absolute Error*) e TV (*Total Variation*).

1.1.2 Resultados Esperados

As abordagens implementadas pretendem sanar os seguintes inconvenientes:

- Navegação em pisos de coloração não homogênea e pisos com alto índice de reflexão de luz;
- Navegação ao longo de caminhos em ambientes com condições adversas de luminosidade;
- Falhas no trajeto de referência;
- Sensibilidade ao ruído e imprecisão nos cálculos do caminho e curvatura visual;

1.2 MOTIVAÇÃO

Com a ascensão da indústria 4.0 e da automatização de carros e robôs, a tarefa de detectar um caminho e segui-lo, torna-se essencial para diversas aplicações que utilizam sensores visuais. Transportar cargas e passageiros de um lugar para outro, e locomover-se ordenadamente num ambiente fechado, são atividades nas quais esse sistema visual se encaixa. Para realizá-las, as aplicações devem ser capazes de apresentar certa resistência a inconvenientes encontrados no ambiente de navegação, por exemplo: um veículo que se locomove a partir de uma referência visual, numa indústria, deve ser capaz de detectar o caminho mesmo quando este apresenta falhas na pintura (causadas pelo tempo) e possíveis

quedas de energia que comprometam a luminosidade local, ou ainda, quando o próprio sistema de iluminação do robô/automóvel falhar em um ambiente externo.

Em alguns trabalhos na literatura (Cherubini; Chaumette; Oriolo, 2008; de Lima; Victorino, 2014), foram propostas soluções para realizar o seguimento de caminho visual. Estas estratégias exigem caracterização prévia do caminho a ser seguido, tais como: restrições na curvatura, velocidade de navegação constante e pré-disposição do trajeto a ser seguido no plano da imagem. Entretanto, nestes casos, tais abordagens não tratam de forma explícita os limites impostos pelo sistema visual. Neste contexto, (Ribeiro; Conceição, 2019) aplicou controladores preditivos não lineares (NMPC) ao problema de seguimento de caminhos, todavia o mecanismo de extração de características visuais não lidava com imperfeições no caminho, falhas, e condições anormais de luminosidade.

Diante deste cenário, tornou-se necessário a adoção de uma abordagem mais robusta para detecção e rastreamento dos caminhos visuais. Somando-se a estes inconvenientes e tendo em vista o aumento do número de aplicações voltadas à processamento de imagens e a intensa procura de trabalhos no IEEE (atualmente liderando o *Top Search* com buscas¹ em torno de 370.000), desenvolver soluções para contornar os mais diversos problemas, torna-se um requerimento para suprir as necessidades nos campos da indústria, transporte e agricultura.

1.3 ESTRUTURA DA DISSERTAÇÃO

A dissertação está estruturada da seguinte maneira:

- O capítulo 2 aborda a descrição formal do problema que se quer resolver;
- O capítulo 3 descreve o sistema visual e portanto os métodos propostos;
- O capítulo 4 descreve os materiais e os ambientes de teste utilizados e a pré-configuração dos experimentos;
- O capítulo 5 apresenta os resultados experimentais e as discussões pertinentes;
- As conclusões e considerações finais são feitas no capítulo 6.

¹Dado relacionado à buscas sobre “*Image Processing*”.
<https://ieeexplore.ieee.org/Xplore/home.jsp>. Acessado em 05 de julho de 2020.

Fonte:

FORMALIZAÇÃO DO PROBLEMA

O problema de seguimento de caminhos visuais, como proposto em (Ribeiro; Conceição, 2019), possui seus elementos principais conforme ilustrado na Fig. 2.1. Nesse modelo, para um perfil de velocidade linear pré-especificado v , os estados (idênticos às saídas, no presente caso) são características visuais extraídas de um sistema de visão computacional a cada interação, e a entrada é definida como o erro entre a velocidade angular (ω) do robô e de um veículo virtual posicionado no horizonte visual. Tal modelo é definido da seguinte maneira:

$$\mathbf{u}_e = \omega - c(s) \frac{(v + \omega Z)}{\cos \theta_r}; \quad (2.1)$$

$$\mathbf{x}_e = \begin{bmatrix} \dot{Z} \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \omega H + (\omega Z + v) \tan \theta_r \\ \mathbf{u}_e \end{bmatrix}, \quad (2.2)$$

com:

- $\mathbf{u}_e$: Entrada de Controle;
- s : Tamanho do Caminho;
- $c(s)$: Curvatura do Caminho;
- Z : Deslocamento Lateral (variável de estado);
- θ_r : Orientação de referência (variável de estado);
- H : Horizonte Visual.

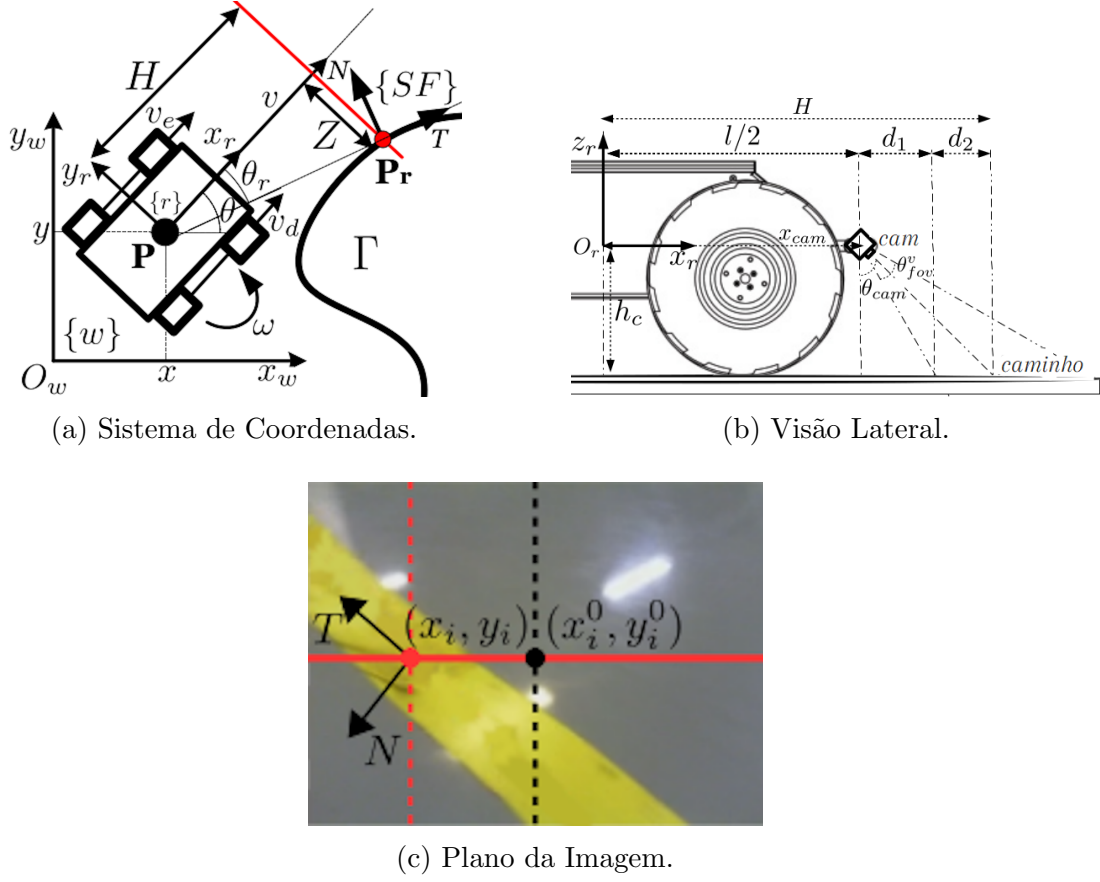


Figura 2.1: Seguimento de Caminho Visual Operando Normalmente

Originalmente, os estados Z e θ_r são calculados diretamente do plano da imagem da seguinte maneira:

$$Z = k_z(x_i - x_i^0); \quad (2.3)$$

$$\theta_r = k_\theta \arctan\left(\frac{x_i - x_i^0}{y_i}\right); \quad (2.4)$$

$$y_i = y_i^0, \quad (2.5)$$

com:

- k_z e k_θ : Constantes de calibração linear e angular;
- x_i^0 e y_i^0 : Coordenadas horizontal e vertical do pixel no centro da imagem;
- x_i e y_i : Pixels horizontal e vertical no caminho.

O horizonte visual é calculado como segue:

$$H = \frac{l}{2} + d_1 + d_2; \quad (2.6)$$

$$d_1 = h_c \tan\left(\theta_{cam} - \frac{\theta_{fov}^v}{2}\right); \quad (2.7)$$

$$d_2 = k_h y_i, \quad (2.8)$$

com:

- h_c : Altura da câmera;
- θ_{cam} : Ângulo do eixo focal da câmera;
- θ_{fov}^v : Campo vertical do ângulo de visão vertical;
- k_h : Constante de calibração do horizonte visual.

Para que este modelo seja representativo, considera-se que o robô está posicionado em frente a um caminho de forma longitudinal ao campo de visão da câmera. Nesse caso, a partir das relações trigonométricas básicas, haverá um acoplamento forte e explícito entre os estados Z e θ_r . No entanto, em algumas situações reais, há possíveis distúrbios no caminho visual, de tal forma que este acoplamento é perdido, gerando altos erros de modelagem. A Fig. 2.2 mostra alguns exemplos considerando caminhos rotacionados e com falhas (*gaps*). Na Fig. 2.2(a) há um caso em que o caminho aparece rotacionado no campo visual, gerando medidas de Z dissociadas de θ_r . Como visto na imagem da Figura 2.2(c), o caminho rotacionado tende a produzir valores de Z próximos a 0, enquanto o caminho real é transversal ($\theta_r = \frac{\pi}{2}$ rad). Da mesma forma, uma falha no caminho, como ilustrado na Fig. 2.2(b), torna o modelo não representativo. Como pode ser visto na imagem da Fig. 2.2(c), Z tende a um valor elevado, fora do plano visual, perdendo o acoplamento com as medidas de θ_r .

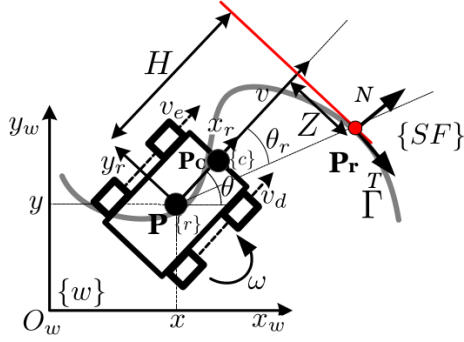
Nas situações ocorridas nos dois cenários da Fig. 2.2, o modelo definido em (2.1) e (2.2) não pode ser usado diretamente em controladores baseados em modelo.

Considerando o parâmetro de curvatura, essas imperfeições no caminho são ainda mais prejudiciais, pois $c(s)$ deve satisfazer:

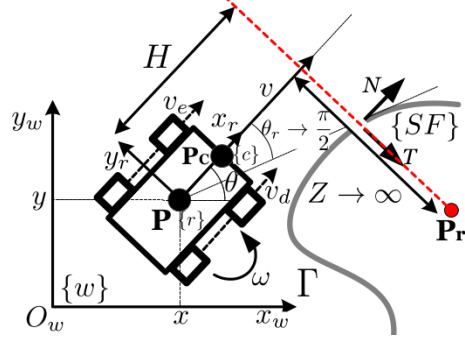
$$\dot{c} = \frac{dc}{ds} \frac{(v + \omega Z)}{\cos \theta_r}. \quad (2.9)$$

A partir dessa condição, é possível notar que, para o sucesso das implementações de controladores baseados em modelo, é necessária a aquisição de medidas precisas de curvatura visual, uma vez que é estabelecida uma relação entre os comportamentos temporal e espacial para a função de curvatura. Por razões de eficiência computacional, esses valores devem ser obtidos em tempo de execução, diretamente do plano da imagem.

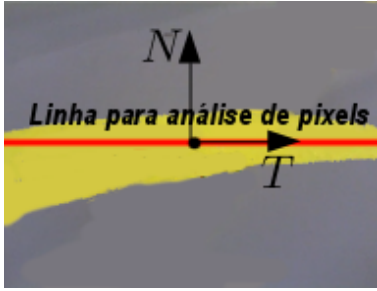
Com os parâmetros extraídos (Z , θ_r e $c(s)$), é possível alimentar o preditor do controlador NMPC. Considera-se uma abordagem NMPC nominal de tempo contínuo que visa minimizar a seguinte função:



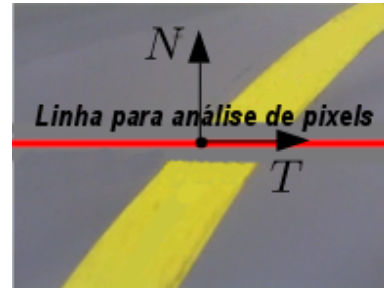
(a) Caminho rotacionado no sistema de coordenadas do mundo.



(b) Caminhos com falha no sistema de coordenadas do mundo.



(c) Caminho rotacionado no plano da imagem.



(d) Caminho com falha no plano da imagem.

Figura 2.2: Alguns problemas encontrados no modelo de seguimento de caminhos visuais.

$$J_{min} = \min_{\mathbf{u}_e} \int_t^{t+T_p} F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) d\tau, \quad (2.10)$$

$$\text{Sujeito a: } \dot{\mathbf{x}}_e(\tau) = f(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)), \quad (2.11)$$

$$\mathbf{u}_e(\tau) \in \mathcal{U}, \forall \tau \in [t, t + T_c], \quad (2.12)$$

$$\mathbf{x}_e(\tau) \in \mathcal{X}, \forall \tau \in [t, t + T_p], \quad (2.13)$$

com o custo de F dado por:

$$F(\mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) = \mathbf{x}_e^T \mathbf{Q} \mathbf{x}_e + \mathbf{u}_e^T \mathbf{R} \mathbf{u}_e, \quad (2.14)$$

onde:

T_p é o horizonte de predição;

T_c é o horizonte de controle; Com $T_c \leq T_p$;

$\mathcal{U}$ é o conjunto de entradas factíveis;

$\mathcal{X}$ é o conjunto de estados factíveis;

$\mathbf{Q}$ e $\mathbf{R}$ são matrizes definidas positivas usadas para ponderar os efeitos da minimização sobre os estados e a entrada.

Como o objetivo do presente trabalho é manipular informações visuais para alimentar o controlador NMPC de maneira eficiente, visando reduzir a complexidade computacional, não são consideradas abordagens adicionais para garantir factibilidade e estabilidade, diferentemente das propostas de (Ribeiro; Conceição, 2019).

Apesar de todos os benefícios proporcionados por este controlador NMPC, é possível notar que, devido a natureza da estratégia de detecção do caminho adotada, são verificados alguns problemas durante a navegação robô. Originalmente, o processamento da imagem se resumia a uma análise dos pixels feita em apenas uma linha na imagem, e que dependendo da velocidade de navegação e da taxa de quadros por segundo, o caminho era perdido, não havendo solução para o problema. Este inconveniente é ilustrado nas Figuras 2.3(a) e 2.3(b).



(a) Análise antes da Descontinuidade.



(b) Análise após Descontinuidade.

Figura 2.3: Problemas encontrados na abordagem original proposta por (Ribeiro; Conceição, 2019)

Como solução parcial, (Ribeiro et al., 2018), foi levada em consideração uma estratégia, baseada em *blobs* utilizando o método de Otsu (Corke, 2011) aplicados em imagens em escala de cinza, Figs. 2.4(a) e 2.4(b). No entanto, apesar das vantagens obtidas em desempenho e esforço computacional, o método aplicado foi limitado quanto ao ambiente de navegação, pois este deveria ser mais controlado, no sentido de haver um

cenário com contrastes bem definidos e cores homogêneas. Sem esses requerimentos, o método não consegue detectar corretamente o caminho.

Um exemplo em que tal abordagem não se aplica é ilustrado nas Figs. 2.4(c) e 2.4(d), que possuem pisos de cores não homogêneas separadas por rejuntas (de agora em diante chamado de cenário 1). Outra desvantagem é observada em superfícies refletoras (de agora em diante chamado de cenário 2). Neste tipo de cenário é notado uma sensibilidade aos reflexos vistos em algumas partes do caminho que, após etapa de segmentação, podem ser entendidos como parte do trajeto de referência, Figs. 2.4(e) e 2.4(f).



(a) Cenário com contrastes bem definidos e cores homogêneas.



(b) Segmentação com Otsu nesse cenário foi adequada.



(c) *Frame* de um piso com cores não homogêneas (cenário 1).



(d) Segmentação não representativa para este cenário.



(e) *Frame* de um piso com alto índice de reflexão (cenário 2).



(f) Pontos de Luz interferindo no caminho identificado.

Figura 2.4: Segmentação utilizando Otsu nos cenários analisados.

Outros problemas da proposta original foram:

- Ausência de uma função representativa do caminho: não houve a estimação de uma função que representasse o trajeto que se queria seguir. Foram utilizados como parâmetro de posição e orientação as características extraídas com relação ao centróide do maior *blob* detectado pelo etapa de segmentação do algoritmo;
- Ausência de um função curvatura explícita: como não houve uma estimação de uma função caminho, a função curvatura foi estimada utilizando uma aproximação por coordenadas de um círculo, ver Fig. 2.5, que gerava inconsistências nos valores à depender da deformidade do trecho que se queria seguir ou se alguma trepidação acometesse o robô. Para ajudar a mitigar esta situação, o sensor visual foi utilizado usado muito próximo ao caminho, fazendo com que este permanecesse aproximadamente como uma linha reta no plano da imagem e para evitar que qualquer trepidação no robô exercesse forte influência nos valores de curvatura gerados.



Figura 2.5: Cálculo de curvatura utilizado.

A sensibilidade dos parâmetros visuais junto aos métodos de extração de caminhos utilizados por estes trabalhos, tornam-se um desafio quando a aplicação é voltada à plataformas de médio e grande porte que tendem a reforçar perturbações mecânicas, como o Husky UGV (Clearpath Robotics, 2020), robô utilizado para testar os algoritmos desenvolvidos neste estudo.

SISTEMA DE DETECÇÃO VISUAL

Neste capítulo serão abordados os fundamentos teóricos dos sistemas de detecção visual utilizados, sendo apresentado, na seção 3.1, os conceitos de espaços de cores e a utilização específica de dois modelos: *Red Green Blue* (RGB) e *Hue Saturation Value* (HSV). As técnicas para a estimação de funções matemáticas usadas para representar um caminho/trajeto a se seguir, como o Método dos Mínimos Quadrados e o RANSAC, são apresentadas na seção 3.2. Após a fundamentação teórica das técnicas usadas no sistema de visão, na seção 3.3, são propostos os métodos deste trabalho para sanar os problemas vistos no capítulo 2.

3.1 ESPAÇO DE CORES

Um espaço de cores é uma organização específica de cores, ou ainda, pode-se dizer que o espaço de cores fornece um método racional que especifica, manipula, organiza e mostra a representação digital dos espectros das cores (Plataniotis; Venetsanopoulos, 2000). Um modelo de cores bem escolhido preserva as informações e provê melhor percepção para operação visual necessária. Dentre os espaços de cores, são encontrados: RGB, HSV, CMY, XYZ, L^*a^*b , Escala de Cinza, dentre outros. No presente trabalho, os 2 espaços de cores discutidos são: RGB e HSV.

3.1.1 RGB

De maneira geral um modelo RGB, acrônimo para *Red* (vermelho) *Green* (verde) *Blue* (azul), é um espaço que utiliza a combinação de três cores primárias: vermelho, verde e azul e que pode ser representado geometricamente, de forma normalizada, por um cubo, como mostra a Fig. 3.1.

Similar ao esquema que ocorre com os cones e bastonetes, células sensíveis a luz nos olhos humanos, o espaço RGB tenta representar a percepção tri-cromática da luz que nosso sistema visual possui, já que temos 3 diferentes tipos de cones que respondem a 3 diferentes porções do espectro das cores.

Para testar e quantificar a teoria tri-cromática de percepção, nos anos de 1930, segundo (Szeliski, 2011), a *Commission Internationale d'Eclairage* (CIE) padronizou a representação RGB (CIE 1931 RGB) executando experimentos com as cores primárias. Essa representação do espectro pode ser vista na Figura 3.2.

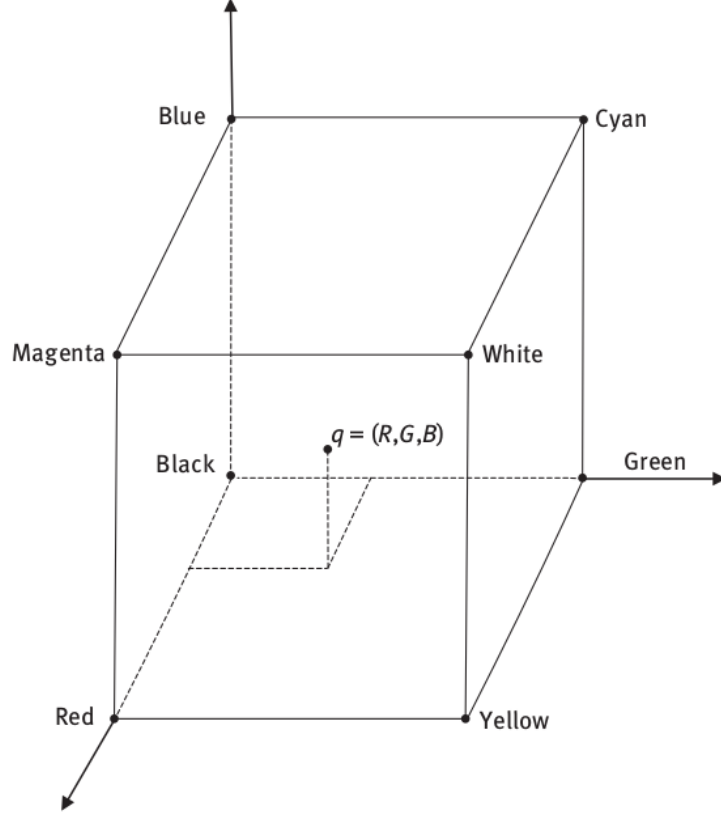


Figura 3.1: Cubo RGB (Su, 2016)

A aquisição de imagens por meio das câmeras digitais é feita, em grande maioria, sob utilização do modelo de cores RGB, e como explica (Szeliski, 2011) essa grande utilização está historicamente relacionada à quantidade de fósforo disponível que continham nos televisores de tubos (também RGB), e que na prática é possível descrever a luz de acordo com a função de resposta espectral dos sensores vermelho, verde e azul presentes nesses aparelhos de capturas fotográficas, como demonstrado nas equações abaixo:

$$\mathbf{R} = \int L(\lambda) S_R(\lambda) d\lambda; \quad (3.1)$$

$$\mathbf{G} = \int L(\lambda) S_G(\lambda) d\lambda; \quad (3.2)$$

$$\mathbf{B} = \int L(\lambda) S_B(\lambda) d\lambda, \quad (3.3)$$

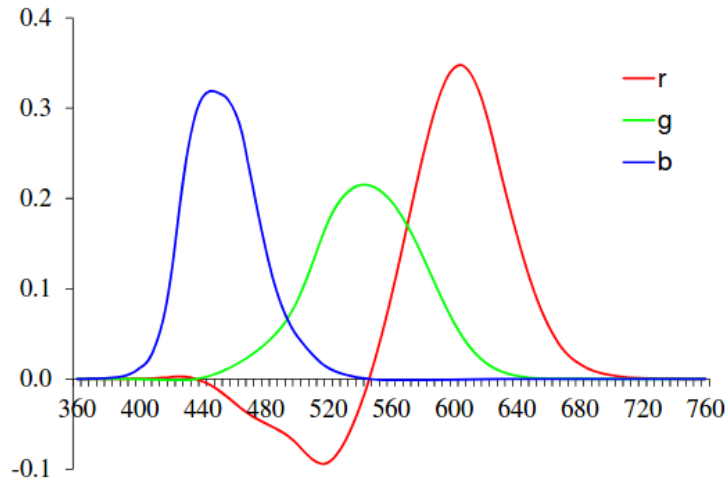


Figura 3.2: Sensibilidade Espectral x Comprimento de Onda. Espectro de cores obtidos a partir da combinações das cores primárias *vermelho* = $700nm$, *verde* = $546.1nm$ e *azul* = $435.8nm$. (Szeliski, 2011)

onde λ e $L(\lambda)$, nas eqs. (3.1), (3.2) e (3.3), são comprimentos de onda e o espectro de luz observados em um determinado pixel respectivamente, e $S_R(\lambda)$, $S_G(\lambda)$, $S_B(\lambda)$, são as sensibilidades espectrais correspondentes, na devida ordem, vermelho, verde e azul.

Uma importante característica deste espaço é cores, como explica (Plataniotis; Venetsanopoulos, 2000), é que ele é orientado a dispositivo, ou seja, está associado com a entrada, processamento e saída de sinais de um dispositivo (*hardware*). Esses modelos são amplamente utilizados em aplicações que exigem consistência na representação das cores, como em monitores e câmeras fotográficas (Sonka; Hlavac; Boyle, 2007)

3.1.2 HSV

O modelo HSV, acrônimo para *Hue* (Matiz) *Saturation* (Saturação) e *Value* (Valor), é um sistema orientado à percepção de cores no espaço. Segundo (Vadivel; Sural; Majumdar, 2005), as características desse sistema se assemelham a percepção de cores na visão humana, pois, fazendo-se uma analogia, os canais *Hue*, *Saturation* e *Value* que representam as imagens, descrevem num pixel:

- A pureza da cor (matiz), que é percebida pela excitação dos cones quando a luz que incide nos olhos possui iluminação suficiente e contém apenas cores monocromáticas;
- A saturação, que nos dá a percepção de como a pureza da cor está diluída sobre a luz branca;
- O valor (brilho), que corresponde a iluminação percebida na imagem formada através das células de bastonetes.

Contudo, no mundo real, a imagem percebida pelo sistema visual humano é composta

de diversos comprimentos de onda, o que gera diferentes níveis de excitações e que a depender do ambiente inserido pode-se resultar na perda de informações de cores.

A identificação da cor de um objeto utilizando o espaço HSV é facilmente percebida mesmo com um certo nível de mudanças no ambiente, por exemplo, todos os tons de amarelo independentemente se estão mais claro, mais escuro ou menos saturado por conta das condições de iluminação, estarão sempre dentro de uma faixa específica de matiz, contudo, ainda segundo (Vadivel; Sural; Majumdar, 2005) valores muito baixos de brilho e/ou saturação podem descaracterizar a cor que se almeja identificar. Um nível de alto de saturação formará uma imagem com cores mais vívidas, caso contrário cores mais neutras, pois quanto menor este nível, mais tons de cinza ele contém. O brilho ou valor, apresenta o quanto de luz foi refletido pelo objeto, podendo estar claro ou escuro, neste caso portanto, quanto menor a quantidade de valor/brilho mais escura se torna a imagem, o valor zero representa o preto qualquer que seja o valor de matiz ou saturação.

Geometricamente o espaço HSV pode ser representado por um cone, Fig. 3.3, onde é possível analisar e observar as características citadas anteriormente. As cores (matiz) estão quantificadas variando num valor entre $[0^\circ - 360^\circ]$, a saturação entre $[0 - 255]$ (onde 0 é a cor cinza e 255 a cor pura) e o valor/brilho entre $[0 - 255]$ (onde 0 é cor preta e 255 a cor com mais brilho).

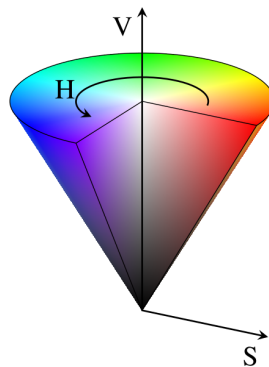


Figura 3.3: Cone de Cores HSV

Uma característica interessante com relação ao espaço de cores HSV, segundo (Platanotis; Venetsanopoulos, 2000), é que ele é orientado a usuário, ou seja, é como se fosse o caminho entre o humano que opera as imagens e o *hardware* usado para manipulá-las. Estes modelos permitem ao usuário descrever e aproximar o que ele percebe ao através da cor.

Outra característica importante desse espaço de cores, segundo (Forsyth; Ponce, 2011), é a não linearidade do espaço devido ao componente matiz (de natureza angular), que diferentemente de um espaço linear como RGB (cubo), captura a intuição humana sobre a topologia das cores. Por exemplo, é intuitivo que a matiz forme um círculo, no sentido que, dado uma cor qualquer, como o vermelho, a matiz vai mudando passando através do laranja, amarelo, verde e então daí para o ciano, azul, roxo, voltando assim novamente ao vermelho.

De forma resumida, abaixo é possível observar as características, vantagens e desvan-

tagens desses dois modelos:

- RGB
 - Linear;
 - Orientado a Dispositivo - Não requer transformações para a exibição das informações nos monitores;
 - Vasta utilização, devido a questões históricas relacionadas à quantidade de fósforo;
 - Dificuldade em identificar e segmentar imagens baseando-se nas cores através dos 3 canais (R, G e B).
- HSV
 - Não Linear - Componente Hue (matiz) de natureza angular;
 - Orientado a Usuário - Similar a como os seres humanos percebem a cor;
 - Bom para aplicações voltadas à processamento de imagens;
 - Singularidade e sensibilidade próximo aos pontos de singularidade do componente Hue. Isso acontece quando a Saturação é 0 ou próxima a 0, ou seja, quando os pixels tendem a acromaticidade;

3.2 AJUSTE DE CURVAS

O ajuste de curvas é um processo de construção de curvas, com o objetivo de representar e tentar prever os dados de acordo com sua disposição no espaço. O ajuste está ligado à duas principais técnicas matemáticas: interpolação e regressão. Ambos os métodos são utilizados para tentar traçar uma relação entre variáveis independentes e variáveis dependentes. A diferença básica entre elas é que dado uma quantidade n de pontos, a interpolação achará uma curva que passe por todos os pontos tendo no final um polinômio de grau $n-1$. Já com a regressão, é possível estimar uma curva de grau menor ou igual à $n-1$.

Nesta dissertação foi estudado também um método estatístico iterativo para estimar parâmetros de uma função mesmo na presença *outliers* (discrepâncias), sendo este chamado de RANSAC (*Random Sample Consensus*)

3.2.1 Método dos Mínimos Quadrados

Como é apresentado em (Boyd; Vandenberghe, 2018), o Método dos Mínimos Quadrados, é uma forma de encontrar soluções aproximadas para sistemas de equações lineares do tipo $Ax = b$, onde A é uma matriz de tamanho $m \times n$ e b um vetor de tamanho $m > n$, ou seja, sobredeterminado, que significa que existem mais equações (m) do que variáveis a se determinar (n). Para achar uma solução para o sistema, é necessário encontrar as melhores variáveis através da minimização da soma dos quadrados dos erros nas equações, sendo que estas equações só possuem solução se b for uma combinação linear das colunas de A .

Como encontrar uma solução para o vetor x de tamanho n , dado o vetor b , é por vezes impossível, o que se faz é calcular o melhor vetor x que minimize a norma do resíduo $r = \|Ax - b\|$, ou de forma semelhante, um vetor que minimize a norma do resíduo ao quadrado, como na equação abaixo:

$$\min \|Ax - b\|^2, \quad (3.4)$$

onde $\|Ax - b\|^2$, pode ser desmembrada como segue:

$$\|Ax - b\|^2 = \|r\|^2 = r_1^2 + r_2^2 + \dots + r_m^2. \quad (3.5)$$

É possível ainda, segundo (Boyd; Vandenberghe, 2018), obter-se uma solução aproximada para o problema dos mínimos quadrados, esta aparece quando dado um vetor $\hat{x}$, ele satisfaz a equação 3.6, para todo x .

$$\|A\hat{x} - b\|^2 \leq \|Ax - b\|^2. \quad (3.6)$$

Esta solução é chamada de solução aproximada do Método dos Mínimos Quadrados.

Como o objetivo deste método é a minimização dos resíduos, a função $f(\hat{x}) = \|A\hat{x} - b\|^2$ deve satisfazer a seguinte condição:

$$\nabla f(\hat{x}) = \frac{\partial f(\hat{x})}{\partial x_i} = 0, \quad i = 1, 2, \dots, n. \quad (3.7)$$

Aplicando as devidas regras de derivação matemática em (3.7), obtém-se, na forma matricial:

$$\nabla f(\hat{x}) = 2A^T(A\hat{x} - b) = 0. \quad (3.8)$$

E por fim, manipulando (3.8), chega-se a solução de $\hat{x}$ com:

$$\hat{x} = (A^T A)^{-1} A^T b. \quad (3.9)$$

O termo $(A^T A)^{-1} A^T$, na equação 3.9 é chamado de pseudo-inversa de A . A matriz pseudo-inversa de A pode ser calculada através de diversos métodos de decomposição, como Equações Normais (eq. 3.9), SVD (*Single Value Decomposition*) e a Fatoração QR (Lee, 2012).

3.2.2 RANSAC

O algoritmo iterativo RANSAC (*Random Sample Consensus*) foi uma técnica proposta por Fischler and Bolles (Fischler; Bolles, 1981) com o intuito de estimar parâmetros de um modelo mesmo na presença de *outliers*. Esta ferramenta é bastante difundida na área de visão computacional devido a sua facilidade de implementação e robustez a ruídos nos dados (Derpanis, 2010). Segundo os autores que propuseram o algoritmo, técnicas clássicas como Método dos Mínimos Quadrados não possuem mecanismos internos para detectar e rejeitar erros grosseiros, que podem descaracterizar completamente o ajuste de um modelo, como visto na Figura 3.4. Este modelos são baseados em técnicas de

média, já que assumem que o máximo desvio esperado de qualquer dado em um modelo é uma função direta do conjunto de dados, esperando-se então que sempre haja um número suficiente de dados que suavizarão qualquer erro grosseiro (*outlier*), o que pode não sempre ocorrer.

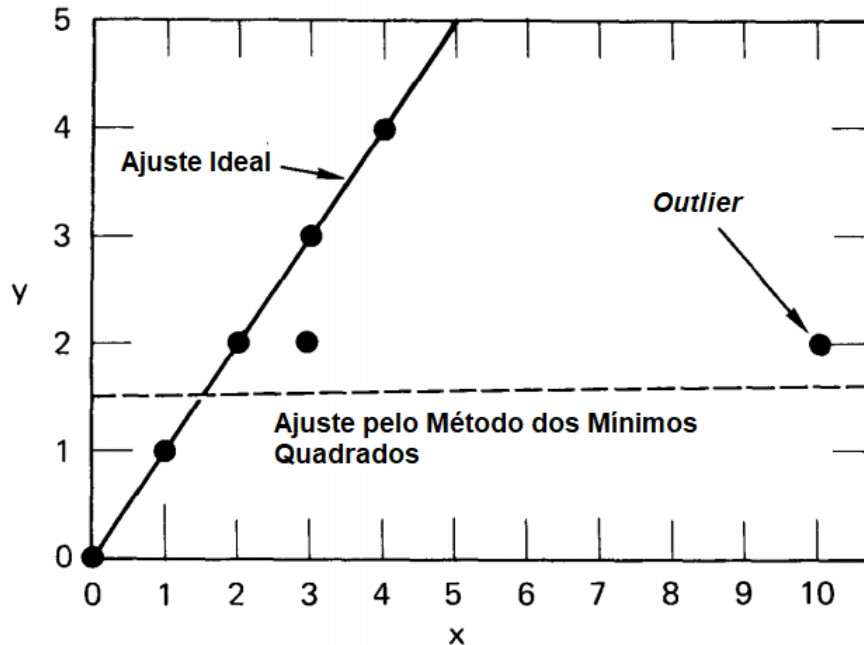


Figura 3.4: Ajuste Através do Método dos Mínimos Quadrados (Fischler; Bolles, 1981)

Com o método RANSAC, é possível ajustar curvas mesmo com mais de 50% de *outliers* nos dados. Tal estratégia foi originalmente pensada em duas etapas, sendo executadas de forma iterativa (Zuliani, 2014):

- **Hipótese:** Um conjunto mínimo de amostras dos dados é escolhido de forma aleatória (*Random Sample*) para estimar o modelo. O número de amostras mínimas necessárias dependerá da cardinalidade do problema. Por exemplo: suponha que se quer estimar uma função afim, então o número mínimo de dados (pontos) necessários para estimar o modelo são 2. Se a função a ser estimada fosse do tipo quadrática o número mínimo de dados seriam 3. Este se difere dos mínimos quadrados pois não leva em consideração todos os pontos.
- **Teste:** No segundo passo o método RANSAC testa se todos os pontos/elementos do conjunto de dados pertencem ao modelo estimado no passo anterior (*Consensus*). Este teste é feito de acordo com um valor predeterminado de desvio máximo permitido para o modelo (σ) que serve para comparar se o ponto/dado é *inlier*, ou seja, pertence ao modelo, ou *outlier*, não pertence ao modelo. Na Figura 3.5 é possível ver como o método considerou o que é *inlier* e *outlier* baseado no valor σ .

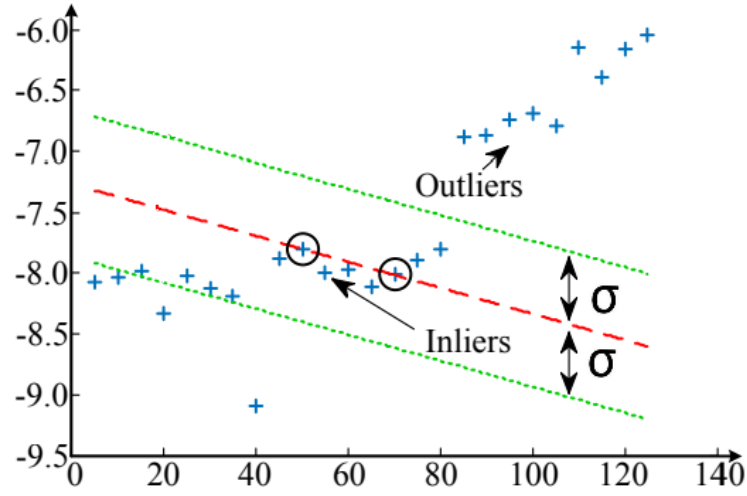


Figura 3.5: Exemplo da estimação de uma função afim e o que o método considera como *Inliers* e *Outliers*. Adaptado de (Heydarzadeh et al., 2017).

A quantidade de vezes que este método iterativo é executada é escolhida suficientemente grande para garantir que a probabilidade p de pelo menos uma amostra aleatória dos dados, esteja livre de *outliers*. É possível obter o número de iterações a partir da seguinte equação (Derpanis, 2010):

$$N = \frac{\log(1 - p)}{\log(1 - w^k)}, \quad (3.10)$$

onde k é o número de pontos necessários para estimar o modelo e w é a probabilidade de escolher um ponto *inlier* dado o conjunto de dados, ou seja:

$$w = \frac{\text{número de inliers}}{\text{número total de dados do conjunto}} \quad (3.11)$$

Ainda, para polir a estimativa do método RANSAC, é possível ajustar a solução com base no conjunto *inlier* (conjunto dos pontos válidos para o ajuste), para isso, um ajuste de curvas é aplicado ao final, após a estimativa do RANSAC (Cai et al., 2018). Para melhor visualização, na Figura 3.6, é possível observar um fluxograma dos passos realizados por este método. Neste fluxograma, são mostrados os passos executados para estimar uma função quadrática. Como visto no fluxograma, para a execução correta do método, é necessário iterar N vezes. Essas N iterações, na medida em que há um aumento de *inliers* e *outliers*, tornam-se mais custosas, podendo até comprometer o desempenho da aplicação. O estudo aprofundado nas questões de complexidade e custo computacional não foram alvos deste trabalho, contudo, mais detalhes sobre esses assuntos podem ser vistos em (Raguram; Frahm; Pollefeys, 2008; Zuliani, 2014).

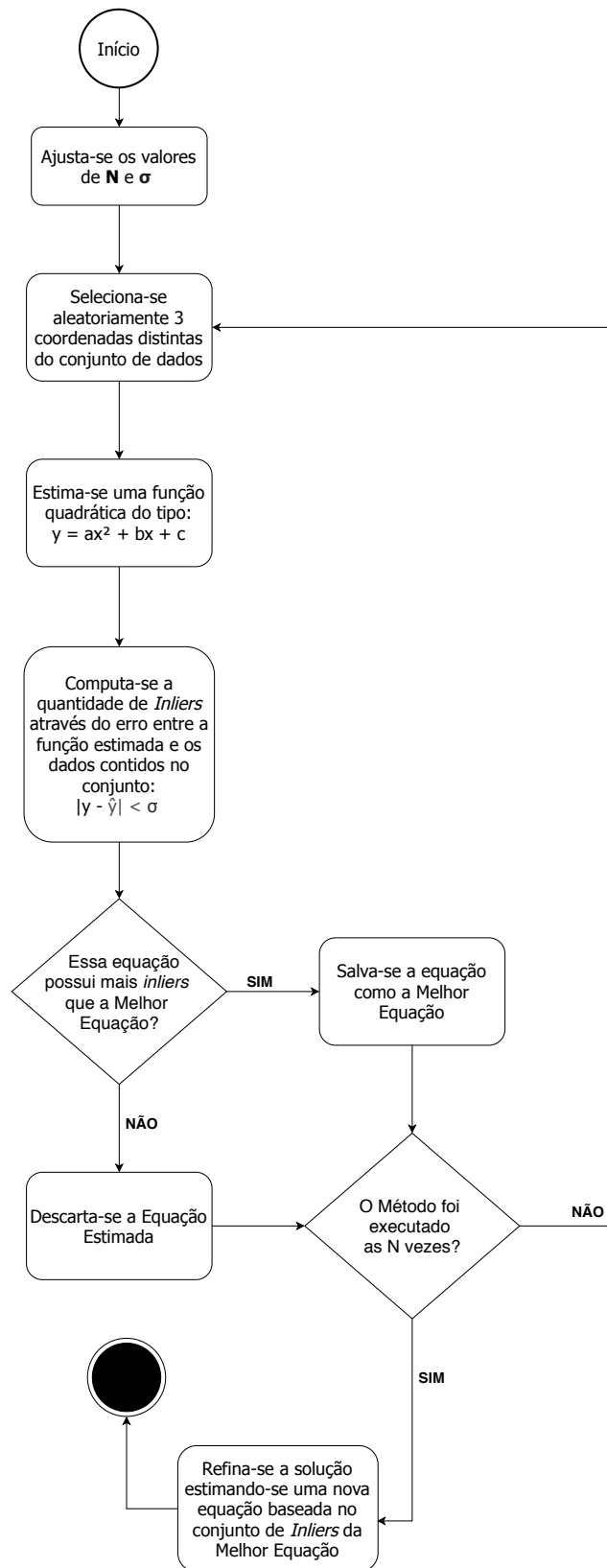


Figura 3.6: Fluxograma do método RANSAC para estimação de uma função quadrática.

3.3 MÉTODOS PROPOSTOS

Após formalizado o problema, e fundamentado as técnicas, este trabalho, como já explicitado no capítulo de Introdução, propõe-se a estudar e desenvolver sistemas para detecção e extração de parâmetros para a alimentação de controladores usados em esquemas de controle baseados em imagem. Para isso, foram analisados 3 diferentes métodos:

- Método 1: Solução baseada em *blobs* adotada em (Ribeiro et al., 2018), descrita anteriormente no capítulo 2, sendo agora melhorada utilizando segmentação através do espaço de cores HSV ao invés do método de Otsu, devido aos problemas ilustrados nas Figuras. 2.4(c) e 2.4(d).
- Método 2: Solução desenvolvida nesse trabalho que utiliza espaço de cores HSV para segmentação do caminho e estimação de uma função utilizando o MMQ para representá-lo.
- Método 3: Solução desenvolvida nesse trabalho voltada a melhorar do Método 2, no qual foi trocado o MMQ pelo método RANSAC. Este método visa melhorar a sensibilidade gerada pelo MMQ na estimação da função caminho e portanto contornar os problemas encontrados com a adoção do Método 2, principalmente aqueles relacionados a ambientes com condições adversas de luminosidade.

O 1º passo tomado por esta proposta, usado para os 3 Métodos, consiste na **segmentação utilizando o espaço de cores HSV**. De maneira resumida, o espaço HSV é útil à aplicação pois fornece informação de como as cores são percebidas, tornando mais fácil a extração do caminho (representado aqui por uma fita amarela) mesmo em pisos de coloração não homogênea e em condições adversas de luminosidade, tarefa que é não trivial quando se utiliza o espaço RGB (*Red Green Blue*), formato em que as imagens são capturadas. Para realizar a segmentação do caminho é necessário então converter as imagens do espaço RGB para HSV. Para isso, no Algoritmo 1, é mostrado como essa transformação é realizada.

Após essa conversão, para extrair o caminho, segmenta-se a imagem a partir de valores específicos para a cor que se almeja detectar. Tais valores variam dentro de um intervalo tanto para a Matiz (*Hue*), que situa-se entre $[0^\circ - 360^\circ]$, quanto para Saturação (*Saturation*) e Brilho/Valor (*Value*) que situam-se entre $[0 - 255]$. Um exemplo dessa conversão de espaço de cores e segmentação podem ser vistos na Figura 3.7.

Detalhes mais aprofundados sobre o espaço de cores RGB e HSV, onde eles são comumente usados e as suas vantagens e desvantagens podem ser melhores observados nas seção 3.1 deste capítulo.

Após conversão da imagem RGB em HSV, o resultado é uma máscara binária (preto e branco), Fig. 3.7(b), em que pixels brancos correspondem ao caminho no plano da imagem. Após extraído o caminho na imagem, o 2º passo tomado por esta proposta é a **estimação de uma função caminho** para contornar todos os problemas vistos no capítulo 2. O grau da parábola escolhida neste caso foi 2, pois abaixo disso a estimação não representa bem o caminho e acima disso torna-se complexo computacionalmente. Então,

Algoritmo 1: Conversão RGB-HSV

Dados: canais R, G e B.

Resultado: canais Hue, Saturation e Value.

$Value = V = \max(R, G, B);$

$\beta = \min(R, G, B);$

se $V \neq 0$ **então**

 | $Saturation = \frac{V-\beta}{V};$

senão

 | $Saturation = 0;$

se $V = R$ **então**

 | $Hue = \frac{60 \times (G-B)}{V-\beta};$

senão se $V = G$ **então**

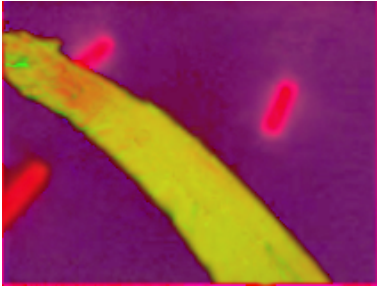
 | $Hue = 120 + \frac{60 \times (B-R)}{V-\beta};$

senão

 | $Hue = 240 + \frac{60 \times (R-G)}{V-\beta};$

se $Hue < 0$ **então**

 | $Hue = Hue + 360;$



(a) Frame convertido de RGB para HSV.



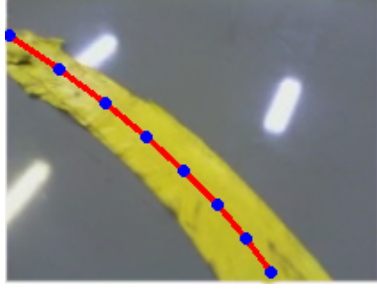
(b) Segmentação para cor amarela (Máscara Binária).

Figura 3.7: Segmentação utilizando espaço de cores HSV da Fig. 2.4(c).

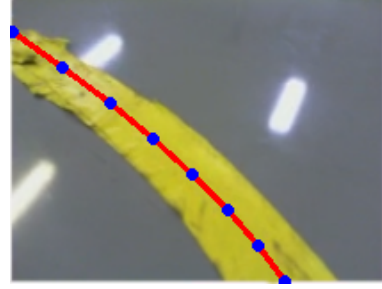
a partir dos pontos gerados pela máscara binária, estima-se que uma função quadrática para representar o caminho que se quer seguir:

$$x = a_p y^2 + b_p y + c_p. \quad (3.12)$$

Para estimar os coeficientes dessa função quadrática, foram utilizadas duas técnicas: o método dos Mínimos Quadrados (Método 2), Fig. 3.8(a) e a técnica RANSAC (Método 3), Fig. 3.8(b). Para as condições de iluminação mostradas nessas imagens, os métodos estimam da mesma forma a função caminho, no entanto, quando testado em ambientes com más condições de iluminação e ruídos, o Método 3 apresenta algumas vantagens que serão melhores discutidas no capítulo referente aos resultados.



(a) Método 2 - MMQ



(b) Método 3 - RANSAC.

Figura 3.8: Estimação dos caminhos (MMQ vs RANSAC) da Fig. 2.4(c).

O Método dos Mínimos Quadrados é uma abordagem utilizada na análise de regressão para aproximar a solução de sistemas sobredeterminados (onde há mais equações do que incógnitas), minimizando a soma dos quadrados dos resíduos (erros) nas equações.

O RANSAC é um método iterativo usado para estimar parâmetros de um modelo matemático a partir de um conjunto de dados observados mesmo que este contenha valores *outliers* (discrepantes). Tal método, diferentemente do MMQ, possui mecanismos para detectar e rejeitar erros grosseiros do conjunto de dados. Sendo assim, esta técnica torna a estimação do caminho menos sensível a ruídos na imagem. Detalhes mais aprofundados sobre ambos os métodos e como eles são utilizados nesta aplicação podem ser encontrados na seção 3.2 deste capítulo.

A partir do caminho gerado, representado pela função quadrática, a função de curvatura é estimada da seguinte forma:

$$\frac{1}{c} = \frac{(1 + (\frac{dx}{dy})^2)^{\frac{3}{2}}}{\left| \frac{d^2x}{dy^2} \right|}. \quad (3.13)$$

Para destacar a importância de estimação dos coeficientes da função quadrática para os cálculos de curvatura, substituí-se (3.12) em (3.13), obtendo-se:

$$c = \frac{|2a_p|}{(1 + (2a_p y + b_p)^2)^{\frac{3}{2}}}. \quad (3.14)$$

Com essa representação de caminho e uma função de curvatura analítica, foi escolhida uma heurística para apoiar mudanças futuras no modelo (como variações temporais no horizonte visual) e também fazer melhor uso das características de controle preditivo, como o horizonte de predição. Portanto, com base na altura do caminho visto no plano da imagem, M pontos principais foram escolhidos para representar a função estimada (nesta aplicação, $M = 8$), como visto na Figura. 3.9. Em cada um desses pontos, é possível extrair informações de pose e curvatura, no entanto, o modelo para seguimento de caminhos visuais baseado em NMPC usado aqui, requer apenas extração referente a um único ponto/coordenada. Portanto, para manter a semelhança com o Método 1, o ponto escolhido nesse caso foi a 5ª coordenada (de baixo para cima), Figura. 3.9.

O parâmetro de orientação (θ_r) é calculado através do ângulo entre essa coordenada

e o eixo y em relação ao centro da câmera. O espaçamento lateral Z é a distância entre o centro da imagem e o caminho estimado, após a calibração adequada.

Com esta forma de extrair os parâmetros, é mantido o forte acoplamento entre as características visuais (Z e θ_r), aumentando assim a representatividade do modelo para seguimento de caminhos visuais mesmo sob as condições adversas ilustradas na Fig. 2.2(b).

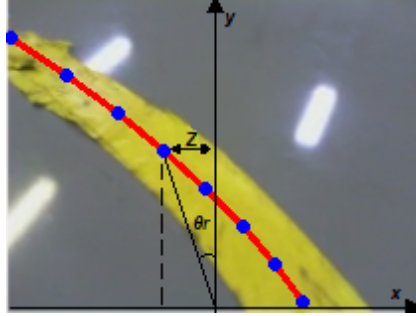


Figura 3.9: Heurística utilizada e Acoplamento entre θ_r e Z .

Após extraídas as características necessárias, é possível fornecer as entradas apropriadas ao preditor do algoritmo NMPC, resolvendo os problemas apresentados no capítulos anteriores. Apesar do controlador utilizado neste trabalho ser um NMPC, os métodos e a heurística proposta não se limitam apenas ao seu uso, outros tipos de controladores, também estão aptos a utilizá-los, pois estas estratégias são voltadas à processamento e extração de informações úteis da imagem.

ESTUDO DE CASO

Os equipamentos utilizados para avaliação dos sistemas propostos foram:

- Robô Husky UGV, Fig. 4.1(a);
- Câmera Microsoft HDCAM 3000, Fig. 4.1(b);
- Computador Intel Core-i5 4^a geração com 8 GB de RAM rodando sistema Operacional Linux Ubuntu 16.04 LTS;
- Plataforma ROS (*Robotic Operational System*) na versão Kinetic (Stanford Artificial Intelligence Laboratory et al., 2016), onde foram desenvolvidos códigos em C++ e python;
- Bibliotecas auxiliares em C++ como OpenCV (Bradski, 2000) e Eigen (Guennebaud; Jacob et al., 2010).



(a) Robô Móvel Husky UGV



(b) Câmera Microsoft HDCAM 3000

Figura 4.1: Materiais Utilizados

4.1 O ROBÔ HUSKY E A PLATAFORMA ROS

O robô Husky UGV (*Unmanned Ground Vehicle*) fabricado pela empresa Cleopath Robotics é um veículo de médio porte, voltado a aplicações em ambientes terrestres *indoor* e *outdoor*. Sua construção robusta e o trem de força de alto torque foram feitos para atender as necessidades dos pesquisadores e engenheiros. O robô possui peso de 50Kg, capacidade de carga em até 75Kg, velocidade máxima de até 1.0 m/s, e em versões mais completas, podem vir com sensores como: câmera, GPS, LIDAR e IMUs. As especificações de altura, largura e comprimento podem ser vistas nas Figuras 4.2(a) e 4.2(b).

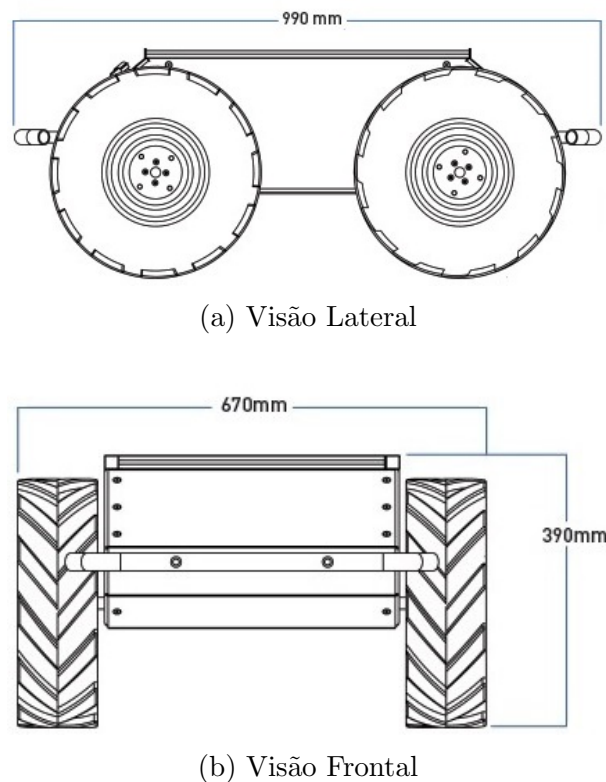


Figura 4.2: Dimensões do Robô Husky

O Husky é totalmente suportado pelo ROS, que é um sistema operacional de código aberto com a utilização voltada a robôs. Ele fornece os serviços que são comumente encontrados em sistemas operacionais, como:

- Abstração de *hardware*;
- Controle de dispositivo de baixo nível;
- Transmissão de mensagens entre processos;
- Gerenciamento de pacotes.

A interface criada para o Husky permite que mensagens contendo informações de velocidade linear e angular do centro de massa (*geometry_msgs/Twist*) sejam enviadas ao robô através de um tópico específico (*husky_velocity_controller/cmd_vel*). Com isso, é possível fazer o Husky movimentar-se no ambiente. Já para a leitura de informações como odometria, basta inscrever-se em um tópico específico (*husky_velocity_controller/odom*) e assim obter as informações requeridas advindas dos encoders do Husky.

Os tópicos supracitados foram essenciais para execução da aplicação e para coleta de dados respectivamente.

4.2 CENÁRIOS DE TESTE

Os cenários utilizados para teste foram 2, sendo que cada um foi utilizado durante uma etapa do desenvolvimento dos Métodos propostos.

O cenário 1, Fig. 4.3(a), foi utilizado para validar os testes nos Métodos 1 e 2. Nesse cenário, há uma fita amarela disposta de forma quase elíptica num chão que possui pisos de coloração não homogênea separados por rejuntas. Nesse cenário, há também falhas e caminhos transversais propositalmente adicionados para testar os sistemas. Durante a época de desenvolvimento, para esse cenário, somente os Métodos 1 e 2 estavam disponíveis, já que este ambiente era mais controlado (quando comparado ao cenário 2) e as estratégias testadas foram suficientes para o propósito. Estes testes serviram como base para um teste mais extremo (cenário 2), onde surgiu a necessidade de um método (Método 3) que fosse mais robusto frente às baixas luminosidades.

O cenário 2, Fig. 4.3(b), foi utilizado para validar os testes em todos os Métodos, e nesse cenário há uma fita amarela disposta de forma sinuosa num chão de epoxy industrial no qual possui coloração cinza homogênea e alto índice de reflexão de luz. Nesse cenário, há também falhas, caminhos transversais e ruídos propositalmente adicionados para testar os algoritmos.

4.3 PRÉ-CONFIGURAÇÃO DOS EXPERIMENTOS

Tomando como referência o sistema de coordenadas ilustrado pela Figura 2.1, os testes foram realizados com a pose da câmera na seguinte configuração:

- $x_r = 0,5\text{m}$;
- $y_r = 0\text{m}$;
- $z_r = -0,1\text{m}$;
- $\theta_{cam} = \frac{\pi}{4}\text{rad}$.

O controlador NMPC, mantendo a nomenclatura apresentada em (Ribeiro; Conceição, 2019), considera:

- $T_p = T_c = 3T_s$;
- $T_s = 0,2\text{s}$;

- $v = 0,2 \text{ m/s}$.

Os parâmetros de sintonia do controlador NMPC:

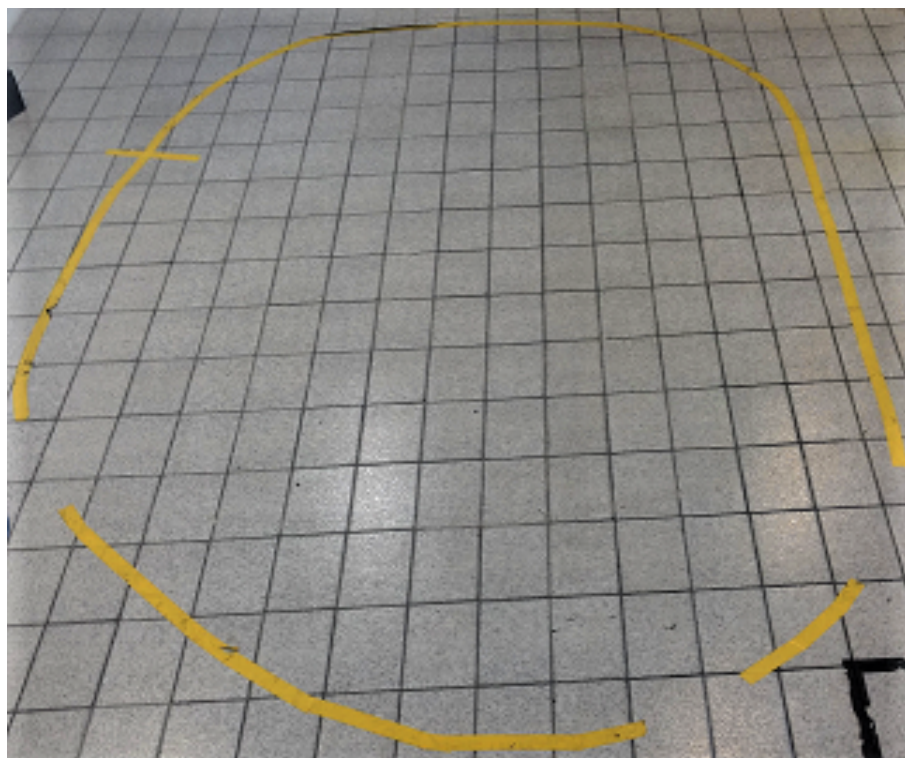
- $Q_i = \text{diagonal}(0, 1; 0, 1)$;
- $R_i = 0,01$.

As restrições elementares do controlador NMPC foram definidas como:

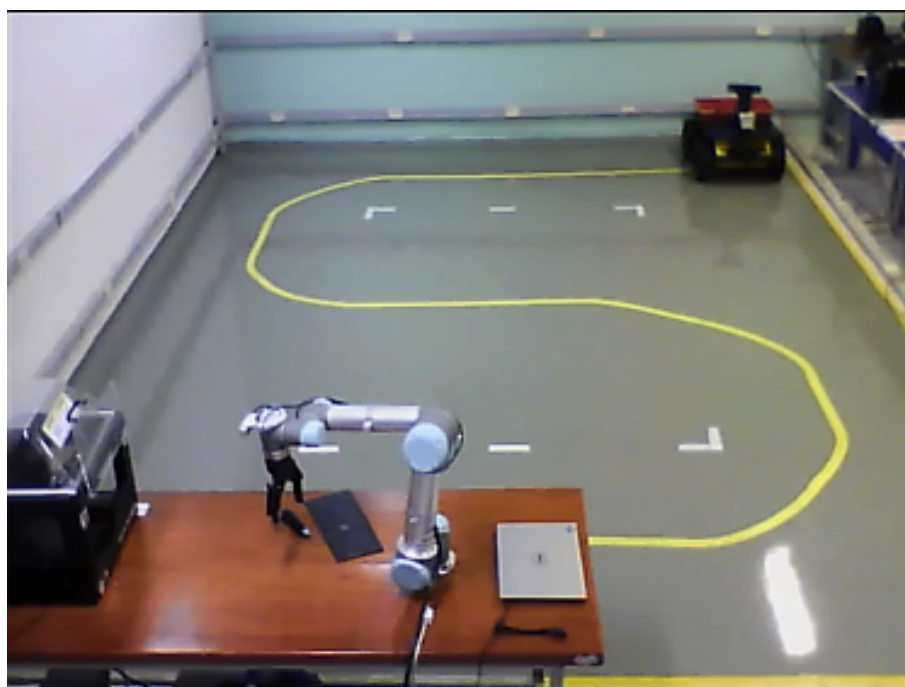
- $|u_1| \leq 0,2 \text{ m/s}$;
- $|z| \leq 0,20 \text{ m}$;
- $|\theta_e| \leq 0,5 \text{ rad}$.

A resolução da imagem utilizada foi:

- 640x480 para o 1º cenário;
- 213x160 para o 2º cenário. A redução da resolução, neste caso, foi realizada para não comprometer a performance do método iterativo RANSAC pela quantidade do número de pixels que são levados em consideração nos cálculos de *inliers* e *outliers*, como visto no fim da seção 3.2.2.



(a) Primeiro Cenário.



(b) Segundo Cenário.

Figura 4.3: Ambientes de Teste.

RESULTADOS EXPERIMENTAIS

Neste capítulo, serão apresentados os resultados dos métodos propostos, os seus desempenhos, os esforços de controle gerados e as discussões pertinentes relacionadas às abordagens.

5.1 PERFORMANCE NO PRIMEIRO CENÁRIO

As discussões relativas a esse cenário são voltadas a:

- Análise do espaço de cores HSV;
- A influência da calibração do espaço HSV no ambiente testado;

5.1.1 Segmentação do Caminho

A primeira parte da etapa de segmentação do caminho tanto no Método 1 como Método 2, é a conversão e segmentação da imagem utilizando o espaço de cores HSV. Para não haver confusão entre o H de HSV e o H de horizonte visual (ver eq. 2.2), as referências individuais aos canais do espaço de cores será tomada como tal: H_c , S_c e V_c , onde c significa “canal”.

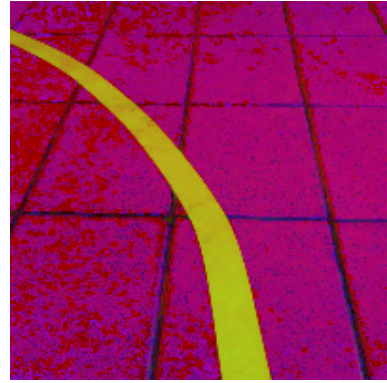
Na Figura 5.1 é possível ver a conversão do espaço de cores RGB em HSV de um *frame* do 1º cenário exposto a uma boa condição de luminosidade. Na imagem, é possível observar que há pontos no chão que possuem cor rosa mais escura que o resto. Neste caso específico, isso aconteceu devido ao problema de singularidade citado na seção 3.1.2, que ocorre no canal H_c . Na Figura 5.2, foi feita uma aproximação na Figura 5.1(a) para inspecionar tal problema. Nesta imagem, são observados os pixels acromáticos, ou seja, pixels sem cor definida, que são interpretados como valores idênticos nos canais R, G e B.

É possível observar mais claramente o problema quando os canais HSV são separados, Figura 5.3. Estudando os canais separadamente, é possível perceber que:

- No canal H_c , há instabilidade nos pixels acromáticos ou próximos a acromaticidade;



(a) Imagem em RGB.



(b) Imagem em HSV

Figura 5.1: Conversão RGB-HSV em imagem do 1^a Cenário.

157	154	149	168	162	157
156	152	148	168	162	157
156	153	149	168	162	157
107	116	119	107	111	98
108	111	114	105	108	98
104	112	116	104	108	97
135	146	160	158	149	136
130	144	158	156	147	134
134	147	159	157	148	135
140	153	159	137	166	168
140	153	159	137	162	168
142	153	159	137	163	168
144	162	145	147	126	153
144	162	145	147	126	153
144	162	145	147	126	153
161	158	153	161	140	114
148	153	151	159	141	118
97	144	154	160	140	118

Figura 5.2: Região com alguns pontos acromáticos da Figura 5.1(a), onde os valores dos pixels RGB são mostrados na cor que os correspondem.

- No canal S_c , há características bastante discriminativas, ou seja, consegue-se perceber bem o contraste entre o caminho e o piso.
- No canal V_c , por mais que exista (nessas condições de luminosidade) um certo contraste entre o piso e a faixa, esse contraste não é tão discriminativo quanto os dos canais H_c e S_c . Muito se deve à simplicidade no cálculo do brilho visto no Algoritmo 1.

Neste cenário com baixa iluminação, Figuras 5.4 e 5.5, que pode ser encontrado, por exemplo, após uma falha no sistema de iluminação, é possível observar como os canais e espaço de cores HSV se comportam. Neste caso, pode-se verificar:

- A imagem capturada pela câmera nestas condições de luminosidade gera bastante ruído. Como observado, quanto pior a condição de luminosidade mais ruído é acumulado;
- O canal H_c mostrou-se um canal discriminativo e com uma certa robustez à luminosidade, dado que com a falta de brilho ainda é possível notar o reconhecimento

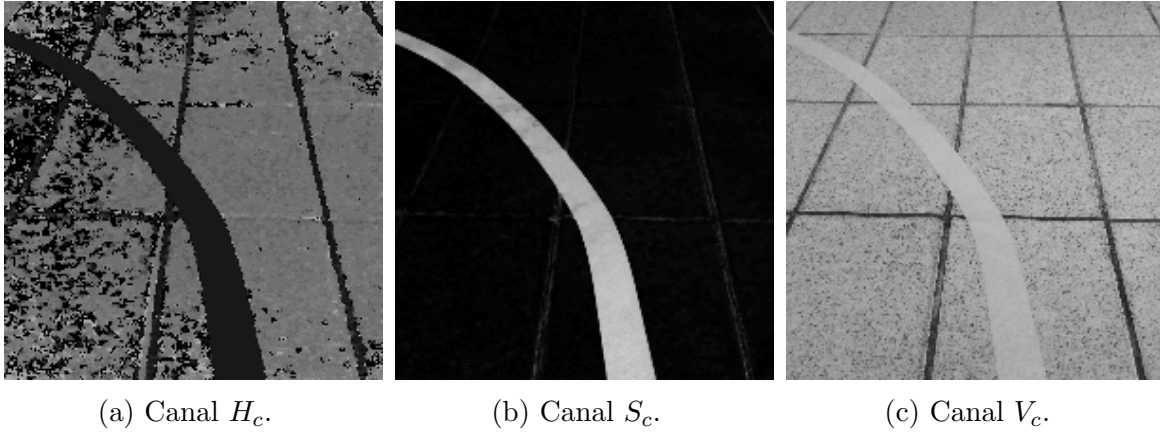


Figura 5.3: Separação dos canais HSV das imagens da Figura 5.1(b).

da cor amarela, ver Fig. 5.6. Porém, como já mencionado, esse canal é instável em pontos acromáticos.

- O canal S_c mostrou-se mais uma vez discriminativo para este cenário, já que mesmo limitado pela baixa luminosidade consegue discernir em saturação entre o caminho e o ambiente de navegação. Contudo, em casos de ainda mais escuridão, a informação desse canal começa a descaracterizar a identificação de cores.
- O canal V_c , mostrou-se um canal não muito confiável para uso em cenas noturnas. No geral, segmentar a imagem diretamente por esse canal se torna uma tarefa complexa devido à simplicidade no seu cálculo ($\max(R,G,B)$), contudo observando o cone de cores HSV (Fig. 3.3), é possível notar que este canal exerce uma forte influência sobre os outros dois que são os que de fato são utilizados para segmentar a cor desejada.

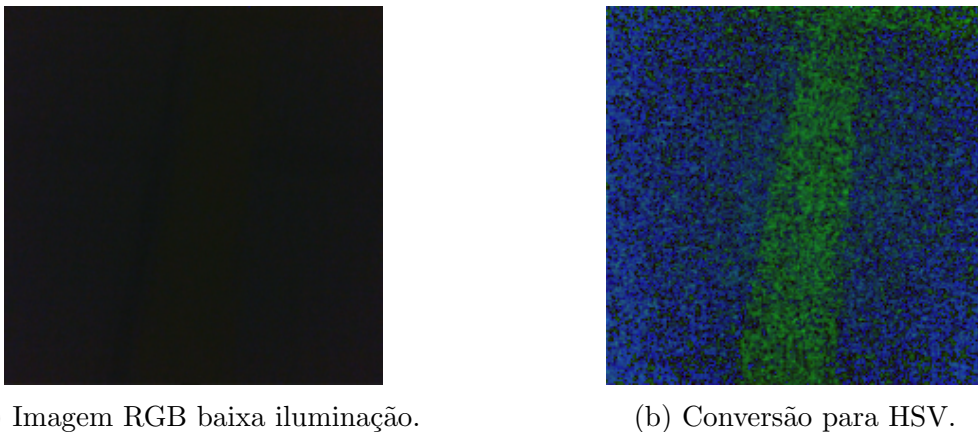
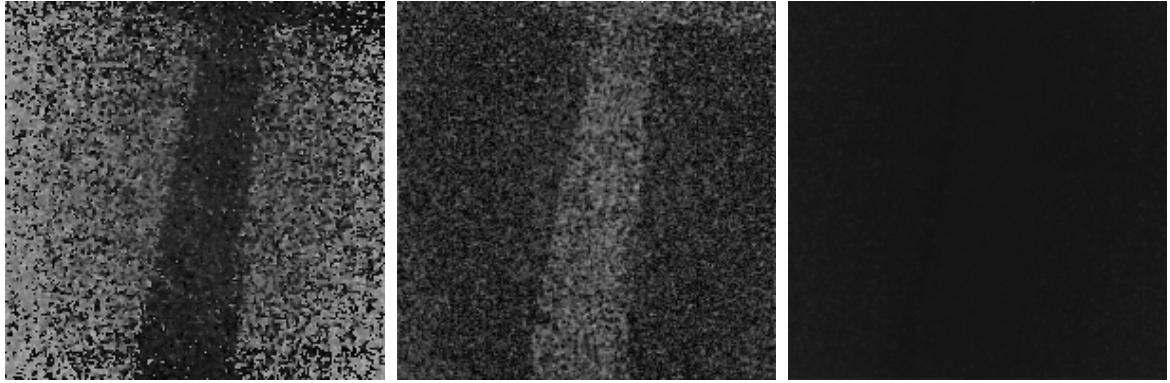


Figura 5.4: Imagem do 1º cenário expostas ao ambiente com baixa iluminação.



(a) Canal H_c da Figura 5.4(b). (b) Canal S_c da Figura 5.4(b). (c) Canal V_c da Figura 5.4(b).

Figura 5.5: Separação dos canais HSV das imagens da Figura 5.4

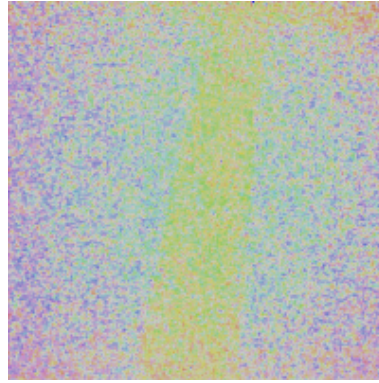


Figura 5.6: Imagem da Fig.5.4(b) convertida de volta a RGB, com o adicional de brilho para destacar a cor amarela detectada pelo canal H_c .

Para finalizar o processo de extração do caminho, as abordagens adotadas visaram a segmentação baseada numa faixa de aceitação tanto para H_c , quanto S_c e V_c . Por isso, calibraram-se os canais¹ para aceitarem faixas relacionadas ao caminho de cor amarela. Para este cenário, o Método 1 foi calibrado aceitando os seguintes canais:

$$H_c = [0 - 45]; S_c = [61 - 255]; V_c = [106 - 194].$$

Já o Método 2, foi calibrado da seguinte forma:

$$H_c = [20 - 40]; S_c = [140 - 255]; V_c = [20 - 255].$$

Para o Método 1, foi adotado uma calibração mais restrita para o canal V_c , já que a estratégia de blobs utilizada por este método não funciona bem quando exposta à imagens e segmentações ruidosas (geradas pelas baixas condições de luminosidade nos cenários

¹Os canais neste caso, utilizando a biblioteca OpenCV, são calibrados e representados em números correspondentes a 8 bits, ou seja, são representações que só aceitam valores entre 0 e 255.

noturnos), ou seja, com tal calibração tenta-se evitar estes tipos de imagens que são problemáticas na segmentação utilizada por este método. Por isso, para o cenário testado, o resultado da segmentação para o Método 1 é apresentado apenas para o ambiente com boas condições de iluminação, Fig. 5.7. Esta problemática das cenas com baixa luminosidade no Método 1 pode ser observada mais claramente no estudo do cenário 2 cujas calibrações foram equalizadas. Os resultados finais da segmentação geram uma máscara binária que corresponde ao caminho desejado.

Na Figura 5.8, são apresentados os resultados da segmentação do Método 2 para as diferentes condições de luminosidade. É interessante notar que, para a Figura 5.8(b), a luminosidade muito baixa deteriora os canais do espaço de cores HSV. Tal fato é visto também quando leva-se em consideração o cone de cores da Figura 3.3, onde é possível ver que quanto menor o brilho/valor, mais descaracterizado (acromático) fica o pixel, ou seja, os níveis máximos de saturação ficam limitados à medida que o brilho vai decrescendo. Isso implica em dizer que, mesmo que todo o canal V_c fosse aceito, para segmentação ($[0-255]$), o resultado final seria o mesmo (Fig. 5.9(a)), pois, não basta somente aceitar uma faixa maior deste, a componente de saturação S_c também foi comprometida e necessitaria ser ajustada para aceitar os baixos valores que foram limitados pela luminosidade. Contudo, se as faixas de aceitação do HSV fossem ajustadas para comportar mais valores, os ruídos vistos nas imagens com pouca luminosidade seriam levados também à segmentação final, Fig. 5.9(b).



Figura 5.7: Segmentação da Figura 5.1(a) na calibração do Método 1.

5.1.2 Representação da Função Caminho

Nesta subseção, será mostrado o resultado do Método dos Mínimos Quadrados (Método 2) sobre as máscaras binárias gerada na etapa de segmentação, Fig. 5.8. Como já mencionado, o Método 1 não gerou uma função que representasse o caminho a se seguir, o que foi feito foi apenas a extração de características baseadas no maior *blob* detectado, abordagem vista em (Ribeiro et al., 2018).

Sendo assim, na Fig. 5.10, são apresentadas as funções de caminho geradas sobre os *frames* anteriores, utilizando a máscara binária extraída. Com atenção especial à Fig. 5.10(b), onde o caminho foi estimado mesmo com os poucos pixels segmentados. Nes-

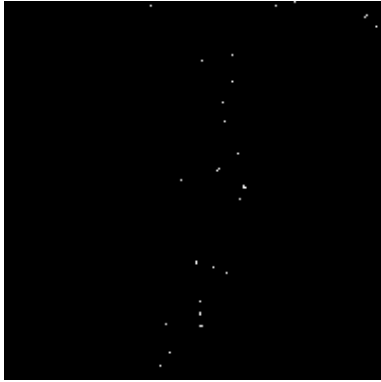
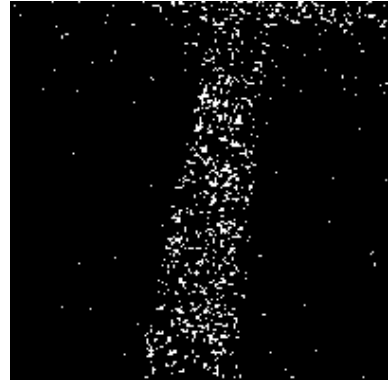


(a) Segmentação da Figura 5.1(b).

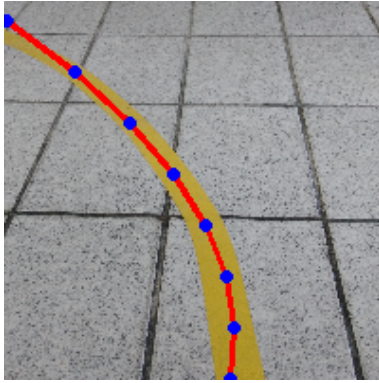


(b) Segmentação da Figura 5.4(b)

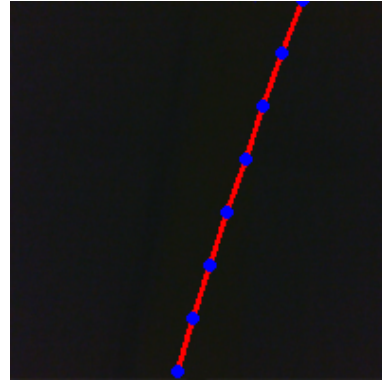
Figura 5.8: Segmentações na calibração do Método 2.

(a) Nova Faixa de Brilho: $V_c = [0 - 255]$.(b) Novas Faixas de Saturação e Brilho: $S_c = [100 - 255]$ e $V_c = [0 - 255]$.Figura 5.9: Análise do Brilho(V_c) e Saturação(S_c) nas segmentações das imagens no ambiente com baixa iluminação da Figura 5.4(b)

tas imagens, é interessante notar que as funções estimadas pelo Método dos Mínimos Quadrados conseguem representar bem o caminho que se quer seguir, inclusive caminhos retos conseguem ser representados de forma satisfatória, Figura 5.11. Contudo, a estimação destas funções pode tornar-se mais sensível na presença de ruídos e obstáculos, tais problemas serão discutidos no próximo cenário com a inclusão do método RANSAC.



(a) Boa Iluminação



(b) Baixa Iluminação.

Figura 5.10: Funções que representam o caminho no 1º Cenário frente às diferentes condições de iluminação.



Figura 5.11: Caminho Reto. Nesta curva, foram gerados os seguintes coeficientes: $a_p = 3.14928e^{-8}$, $b_p = 1.93633e^{-6}$ e $c_p = 93.5$.

5.1.3 Resultados Experimentais

A Fig. 5.12 ilustra os resultados obtidos². Na Fig. 5.12(a) são apresentas as posições instantâneas para as dois Métodos. Observa-se que apenas o Método 2 consegue seguir completamente o caminho, mesmo na presença de distúrbios e falhas propositalmente inseridas. Uma das razões para esse desempenho superior pode ser vista na Fig. 5.12(b), que ilustra melhor precisão na estimativa da curvatura visual.

As Figs. 5.12(c) e 5.12(d) fornecem os erros de seguimento do caminho para os Métodos 1 e 2, respectivamente. Observa-se que, com o Método 2, a perturbação incluída transversalmente ao caminho foi satisfatoriamente rejeitada e as falhas adicionadas, foram transpassadas, comprovando a eficácia da interpolação geométrica realizada. Por outro lado, o Método 1, apesar de regular a perturbação transversal, falha ao deparar-se com *gaps* no trajeto.

Os resultados obtidos sob a perspectiva dos controladores podem ser observados nas

²Esses resultados podem ser visualizados no vídeo disponível em: <https://youtu.be/CG6Yf7Xi-M0>, onde Abordagens Original e Nova referem-se aos Métodos 1 e 2 respectivamente.

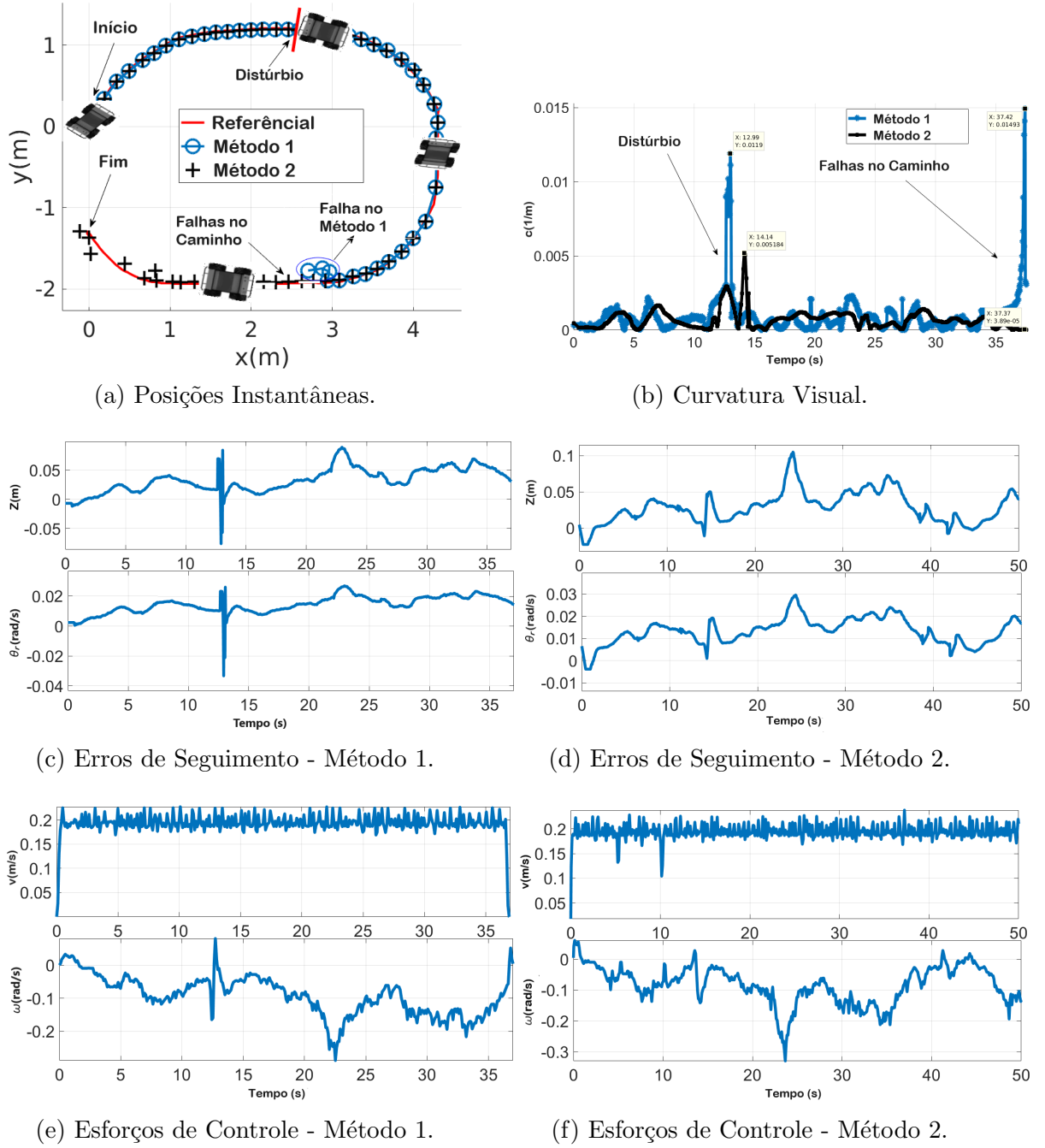
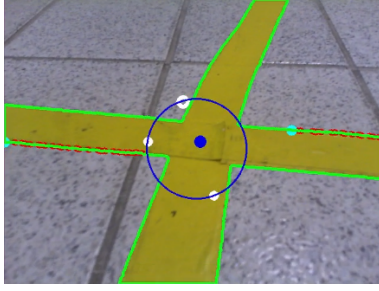


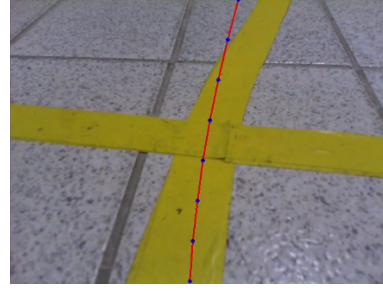
Figura 5.12: Comparação dos Métodos 1 e 2 no Primeiro Cenário.

Fig. 5.12(e) e 5.12(f), onde é possível verificar que, com o Método 1, há mais variações abruptas durante a regulação das perturbações, o que pode levar os atuadores ao desgaste mais facilmente.

A diferença de performance, dá-se pela diferença das estratégias. No Método 1, usa-se uma estratégia baseada em *blobs* (regiões conectadas) sem estimação de função caminho. O uso do maior blob nessa estratégia, por exemplo, impacta diretamente quando se



(a) Distúrbio - Método 1.



(b) Distúrbio - Método 2.

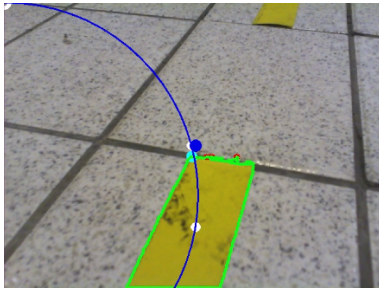
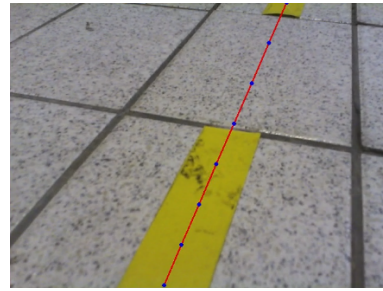
(c) Falhas (*gaps*) - Método 1.(d) Falhas (*gaps*) - Método 2.

Figura 5.13: Algumas das cenas vistas no cenário de teste.

tem caminhos que apresentam falhas (*gaps*), pois quando o algoritmo de segmentação se depara com esses inconvenientes, ele somente leva em consideração um único *blob*, que no caso é a região antes da falha, o que faz o algoritmo parar, por causas de problemas ao calcular os parâmetros de posição e curvatura. Por outro lado, o Método 2, por estimar uma função, consegue representar o caminho completo quando é possível enxergar as duas partes separadas pelo *gap*, ou seja, se houver um mínimo trecho de caminho visível após a falha, o algoritmo consegue estimar uma função para o trajeto a ser seguido. A Figura 5.13 ilustra alguns desses inconvenientes encontrados nos experimentos que reforçam esta análise. Ênfase particular é dada à análise da Fig. 5.13(a), onde se constata que, com o Método 1, a perturbação transversal gerou um círculo com um raio de curvatura muito pequeno, tendendo a gerar ações de controle elevadas, como já mencionado, podem ser prejudiciais às plataformas robóticas.

Por fim, os métodos são comparados através de duas métricas quantitativas, a Integral do Erro Absoluto³ (*IAE*), calculada por $\int_0^{T_{END}} |e(t)| dt$, e a Variação de Controle Total⁴ (*TV*), calculada por $\sum_{k=0}^{k_{END}} |u(k) - u(k-1)|$. Na Tabela 5.1 são apresentadas as métricas relacionadas aos erros (z e θ_r) e as ações de controle (v e ω) para o mesmo período em que as abordagens funcionaram. Nela percebe-se que, para os erros de caminho, o Método 2 é superior ao Método 1, comprovando a eficácia da interpolação geométrica. O leve aumento no índice ω_{TV} (o índice v_{TV} , apesar de menor, neste caso não é controlável),

³Este índice é muito utilizado para comparação de desempenho de estratégias distintas em experimentos semelhantes.

⁴Com este índice, objetiva-se avaliar o efeito dos ruídos nos sinais de controle.

é reflexo da melhor precisão nos cálculos dos parâmetros, que exige um pouco mais de esforço para seguir o caminho corretamente.

	IAE		TV	
	$Z_{iae}[m]$	$\theta_{iae}[rad]$	$v_{tv}[m/s]$	$\omega_{tv}[rad/s]$
Método 1	1.2177	0.6910	6.2216	4.1044
Método 2	1.1395	0.4924	5.9633	4.3441

Tabela 5.1: Análise Quantitativa

Para o ambiente com pouca iluminação⁵, o Método 1, como explicado anteriormente, não detecta um caminho válido. Já o Método 2, consegue reconhecer, estimar as funções caminho e navegar sobre o ambiente. No teste feito, o Método 2 consegue detectar o caminho tanto com um ponto distante de luz artificial (lanterna), Fig. 5.14(a), quanto sem fonte luminosa alguma, Fig. 5.10(b), já o Método 1, consegue detectar o caminho apenas quando um fonte de luz artificial é introduzida próxima ao robô, Fig. 5.14(b), e mesmo assim a detecção de um *blob* que represente o caminho ainda é difícil, às vezes detectado-se apenas uma pequena seção, o que compromete o cálculo da curvatura, Fig. 5.15.

Como o Método 1 não funciona em ambientes de baixa iluminação, pelos motivos discutidos, as métricas quantitativas não foram usadas para comparar as estratégias.



(a) Ponto de Luz Distante.



(b) Ponto de Luz Próximo.

Figura 5.14: Fontes de Luz Artificial no Ambiente com pouca iluminação.

⁵Estes resultados podem ser melhor visualizados no vídeo disponível em: <https://youtu.be/Hr-Bjfe3b1Y>, onde Abordagens Original e Nova referem-se aos Métodos 1 e 2 respectivamente.

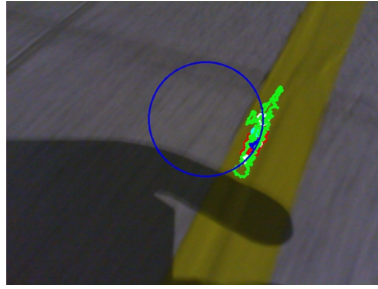


Figura 5.15: Segmentação com Fonte de Luz Próxima ao Robô no Método 1. Nota-se que a curvatura calculada (pequeno círculo azul) é completamente distorcida com relação ao caminho real.

5.2 PERFORMANCE NO SEGUNDO CENÁRIO

As discussões relativas a esse cenário são voltadas a:

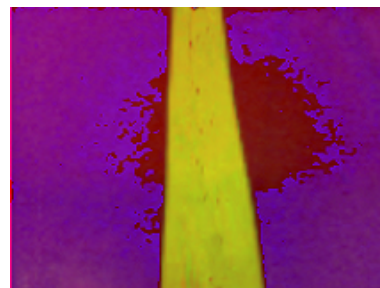
- Análise do espaço HSV;
- Introdução do Método 3 e comparação entre RANSAC e MMQ;
- Análise de desempenho dos métodos em condições de boa e extrema adversidade de iluminação.

5.2.1 Segmentação do Caminho

De maneira similar a seção anterior, a conversão para HSV da imagem no segundo cenário, em boas condições de luminosidade, segue representada nas Figura. 5.16. Nesta imagem, é possível observar um borrão avermelhado mais ou menos na altura do meio do caminho. Esse borrão, foi causado pela instabilidade do canal H_c próximo a pontos quase acromáticos, Fig. 5.17. Um detalhe importante sobre os pontos acromáticos é que eles apresentam-se sob cores neutras, como: branco, preto e cinza, e o caminho desse cenário é predominantemente em tom de cinza.



(a) Imagem em RGB.



(b) Imagem em HSV

Figura 5.16: Conversão RGB-HSV em imagem do 2^a Cenário.

105	104	103	102
102	101	101	101
99	100	101	101
104	103	102	102
101	101	101	100
98	101	101	100
103	101	100	101
100	100	99	99
99	101	101	100
102	100	99	99
99	98	97	97
98	101	100	100

15	10	5	3
105	104	103	102
15	8	0	0
104	103	102	102
15	5	3	5
15	0	0	0
103	101	101	101
10	3	5	5
8	150	135	165
102	101	100	100
10	8	8	8
8	140	140	140

(a) Pixels em RGB tendendo a acromatidade. (b) Pixels Instáveis, onde H_c está em azul, S_c em verde e V_c em vermelho.

Figura 5.17: Instabilidade no espaço de cores HSV perto de pontos acromáticos.

É possível observar mais claramente o problema quando os canais HSV são separados, Figura 5.18. Estudando os canais separadamente, é possível perceber algumas coisas como:

- No canal H_c , há instabilidade nos pontos acromáticos como já relatado anteriormente;
- No canal S_c , há características bastante discriminativas, ou seja, consegue-se perceber bem o contraste entre o caminho e o piso. Tal contraste, é enfatizado devido a cor cinza do caminho, que apresenta baixa saturação, que pode ser comprovado através do cálculo visto em Algoritmo 1;
- Para o canal V_c , observa-se o mesmo que foi descrito anteriormente (cenário 1) para o ambiente de boa iluminação.



(a) Canal H_c .

(b) Canal S_c .

(c) Canal V_c .

Figura 5.18: Separação dos canais HSV das imagens da Figura 5.16(b).

Neste cenário com iluminação mais baixa, Figuras 5.19 e 5.20, que assim como no cenário anterior, pode ser causado por uma falha no sistema de iluminação, é possível observar como os canais e espaço de cores HSV se comportam. Neste caso, pode-se relatar:

- A imagem capturada pela câmera nestas condições de luminosidade gera bastante ruído. Como observado, quanto mais escuro mais ruído é acumulado;
- Nestas condições de luminosidade, apenas o canal S_c mostrou-se moderadamente discriminativo. Como já mencionado, a escuridão favorece às baixas saturações, e além disso, este cenário apresenta como ambiente de navegação um caminho que possui piso em cor cinza, o que dificulta ainda mais a segmentação da imagem por causa da instabilidade no canal H_c que pode fazer com que um pixel fora do caminho principal seja considerado na segmentação final.
- Para o canal V_c , observa-se o mesmo que foi descrito anteriormente (cenário 1) para o ambiente de baixa iluminação.

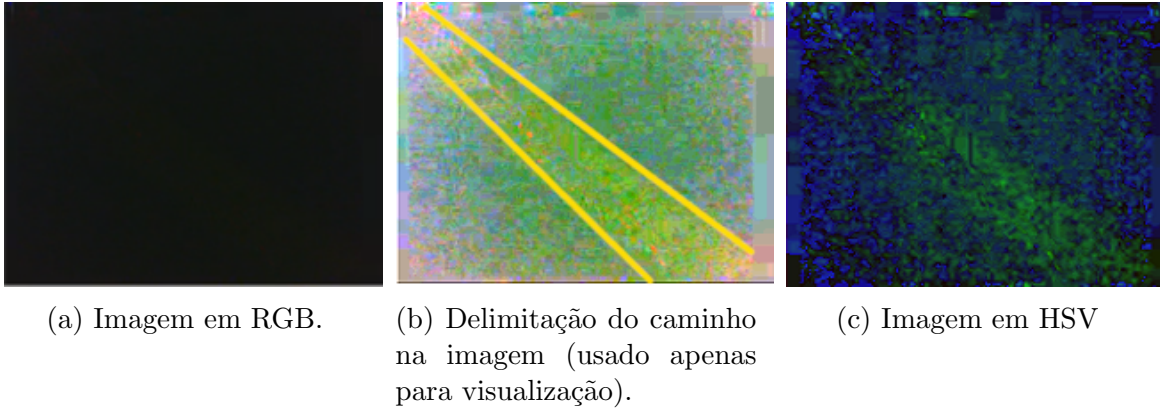


Figura 5.19: Conversão RGB-HSV em *frame* do 2^a Cenário sob baixas condições de luminosidade.

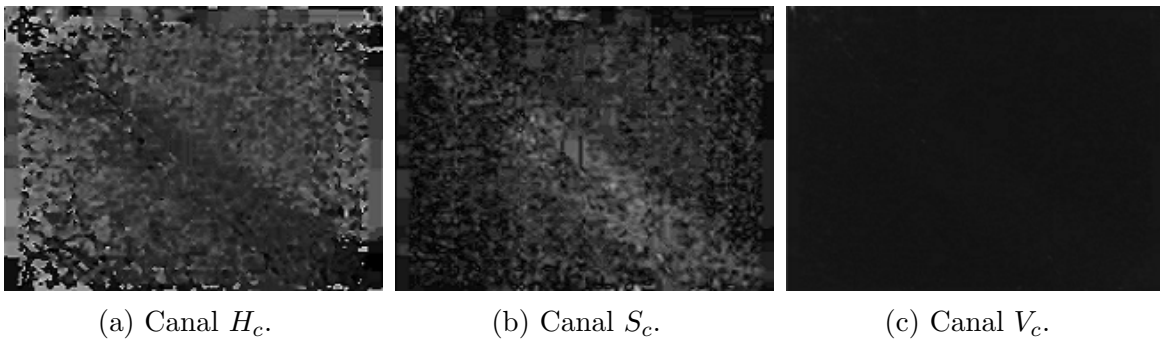
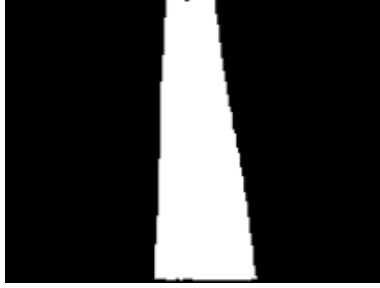


Figura 5.20: Separação dos canais HSV das imagens da Figura 5.19(c).

Para finalizar o processo de segmentação, calibrou-se os canais, para as 3 abordagens, com as seguintes faixas de valores:

$$H_c = [20 - 40]; S_c = [90 - 255]; V_c = [0 - 255].$$

As devidas segmentações podem ser vistas nas Figura 5.21.



(a) Segmentação da Figura 5.16(b).



(b) Segmentação da Figura 5.19(c)

Figura 5.21: Segmentações na calibração para o 2º cenário.

5.2.2 Representação da Função Caminho

Nesta subseção, serão mostrados os resultados do método RANSAC para estimação da função caminho, assim como uma comparação com o Método dos Mínimos Quadrados sobre as máscaras binárias geradas após a segmentação. Mais uma vez, a 1ª abordagem não gerou uma função que representasse o caminho a se seguir.

Sendo assim, na Figs. 5.22 e 5.23, são apresentadas as estimações das funções utilizando o Método dos Mínimos Quadrados versus o método RANSAC para os ambientes com boa e adversas condições de iluminação respectivamente.

Para os ambientes claros, os métodos estimam a curva de maneira similar assim como visto no capítulo 3, contudo, para caminhos expostos à baixa luminosidade o método RANSAC mostra-se superior.

Nas cenas capturadas no escuro, há uma queda na qualidade das imagens, além de aumento no ruído. É por isso que métodos como o RANSAC, que são mais robustos aos ruídos, se tornam uma opção melhor diante dessas condições de iluminação. Como visto na Fig. 5.19, a imagem RGB original vista a olho nu, não possui informações relevantes, sendo necessário manipulá-la através de variações empíricas no brilho e no contraste para perceber o caminho 5.19(b). Como pode ser visto nas Fig. 5.23, com o novo sistema visual baseado em RANSAC o caminho foi identificado de maneira mais satisfatória.

O RANSAC para estes cenários, foi calibrado com os seguintes parâmetros:

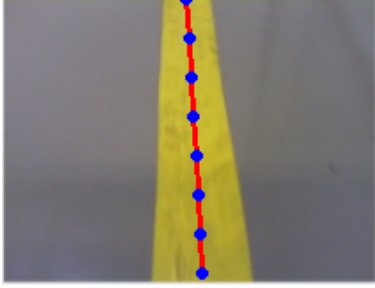
$$N = 55; \sigma = 36;$$

Com $N = 55$, pode-se encontrar o caminho mesmo com 57% de *outliers* (ou 43% de *inliers*), esse valor⁶ pode ser calculado através das equações 3.10 e 3.11. Para tentar segurar a perda de performance devido ao cálculo da RANSAC sobre todos os pontos válidos de uma imagem binária, reduziu-se⁷ o tamanho da imagem de entrada de 640x480

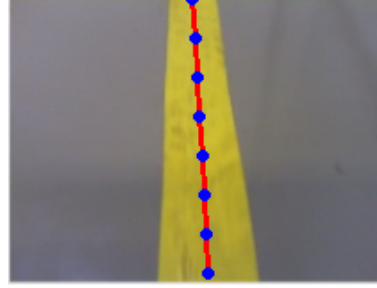
⁶Para calcular o valor de N , considerou-se $p = 0.99$.

⁷Para os experimentos, o tamanho das imagens em todas as abordagens foi reduzido de maneira igual.

para 213x160. Com este tamanho de imagem, pôde-se estimar o valor⁸ de $\sigma = 36$.

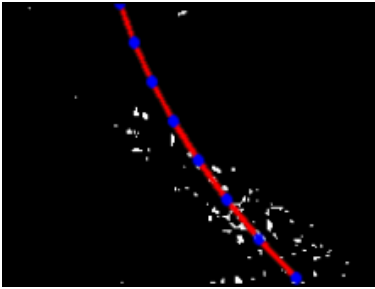


(a) Estimação pelo Método 2.

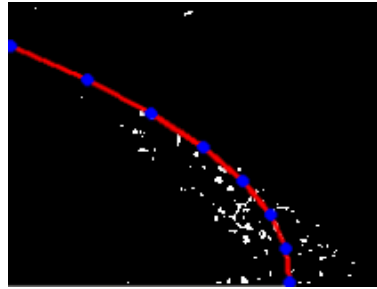


(b) Estimação pelo Método 3.

Figura 5.22: Comparação Mínimos Quadrados vs RANSAC em ambiente com boa iluminação no cenário 2.



(a) Estimação usando o Método 2.



(b) Estimação usando o Método 3.

Figura 5.23: Estimação dos caminhos (MMQ vs RANSAC) em condições de baixa luminosidade no cenário 2.

5.2.3 Resultados Experimentais

A Fig. 5.24 expressa os resultados coletados. A Fig. 5.24(a) mostra as posições instantâneas nos três métodos. Nota-se que apenas o Método 2 e o Método 3 puderam concluir o percurso, mesmo na presença de ruídos e falhas (*gaps*) intencionalmente adicionados. Uma das razões para esse desempenho superior foi a estimativa da função do caminho, que gerou melhores resultados de curvatura. O Método 3, neste caso, foi um pouco melhor que o Método 2, pois passou por distúrbios (caminhos transversais) e ruídos com menor variação. Esses resultados podem ser vistos na Fig. 5.24(b).

A Fig. 5.24(c) fornece o erro de caminho visual para todos os três métodos. Mais uma vez, observa-se que somente no Método 2 e no Método 3, a perturbação incluída transversalmente ao caminho foi adequadamente rejeitada e as falhas adicionadas a ele também foram transpassadas, confirmando a efetividade da interpolação geométrica realizada. O Método 1, pelo contrário, apesar de regular a perturbação transversal, não pôde continuar quando deparou-se com a falha no caminho.

⁸O valor de σ depende da posição e orientação da câmera montada no robô

Os resultados obtidos para os esforços de controle podem ser visualizados na Fig. 5.24(d), onde é possível verificar que todos os experimentos durante a navegação estão bem regulados e respeitam os limites físicos da velocidade linear e angular do Husky UGV.

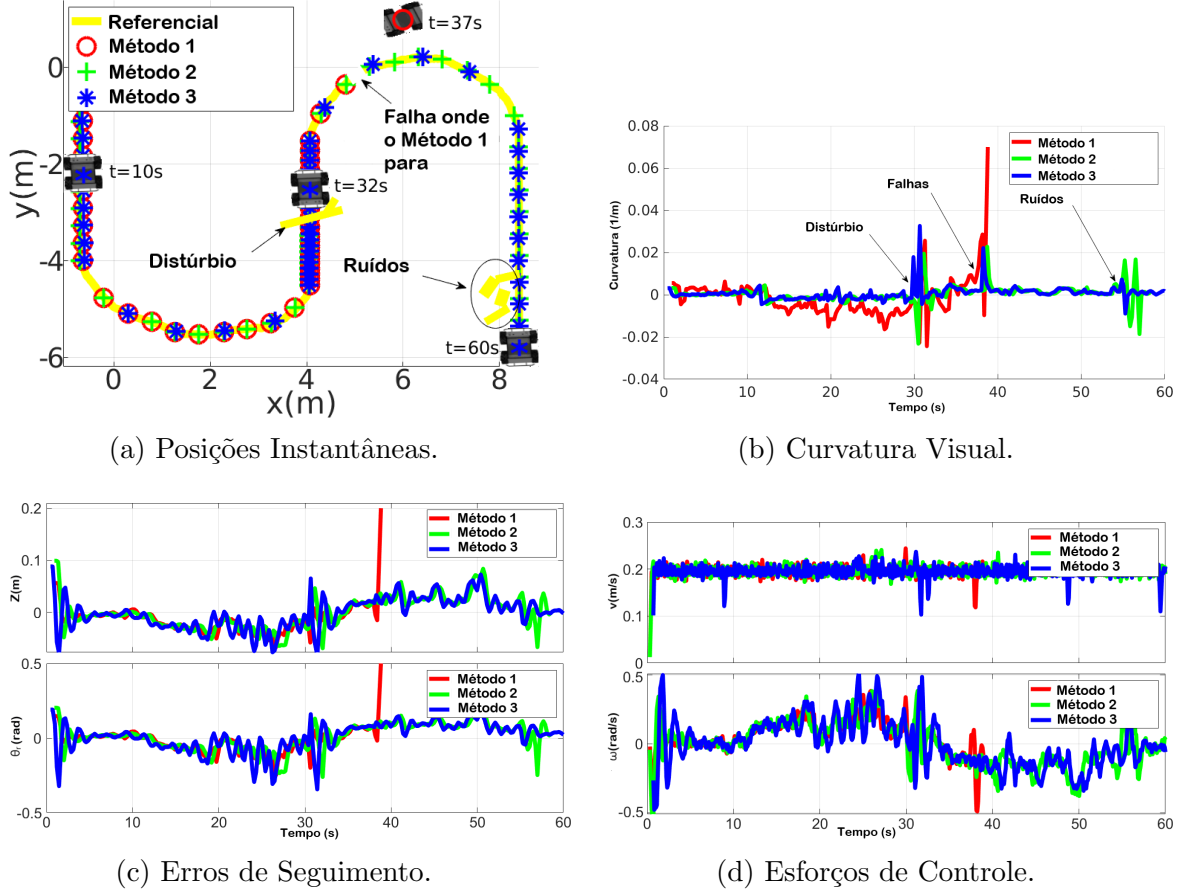


Figura 5.24: Comparação das abordagens em ambiente com boa iluminação.

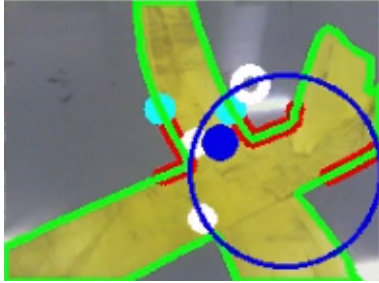
As técnicas utilizadas nas estratégias são as principais causas de diferenças de desempenho. O Método 1 usa uma técnica baseada em regiões conectadas (*blobs*) para a etapa de segmentação do caminho. Os parâmetros fornecidos ao controlador foram estimados a partir desta região conectada, sem o uso de uma função de caminho, o que causa perda de referência e termino da navegação quando há falhas no percurso.

Por outro lado, o Método 2 e o Método 3 estimam uma função quadrática que representa melhor o caminho a ser seguido e, portanto, fornece parâmetros mais confiáveis ao preditor. Na Fig. 5.25, é possível ver como eles lidam com cada desafio, com ênfase no Método 1 que gera um raio de curvatura muito baixo quando confrontado com caminhos transversais e ruídos (Figs. 5.25(a) e 5.25(g)⁹), e a superioridade dos outros dois métodos ao tentar estimar os caminhos com falhas (Figs. 5.25(e) e 5.25(f)). No Método 3, percebe-se que há ainda menos sensibilidade a esses problemas do que as outras estratégias (Figs.

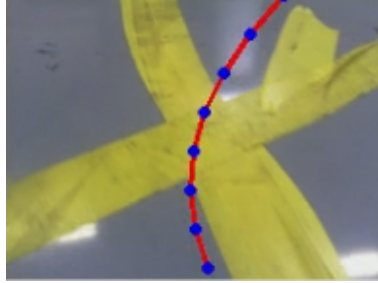
⁹Esse *frame* foi adquirido levando-se manualmente o robô até o ponto do caminho com mais ruído.

5.25(c) e 5.25(i)).

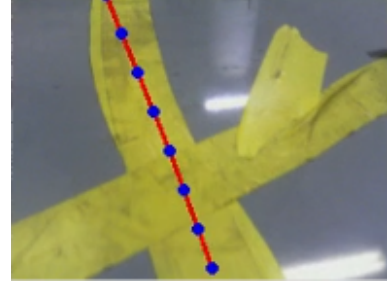
Assim como no cenário anterior, as métricas quantitativas da integral do erro absoluto (IAE) e a variação total do controle (TV) foram usadas para comparar os experimentos.



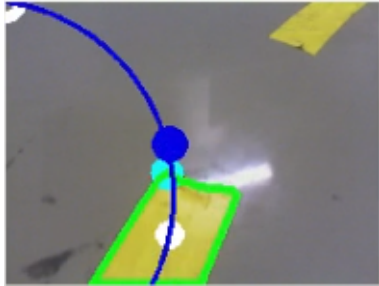
(a) Distúrbio - Método 1.



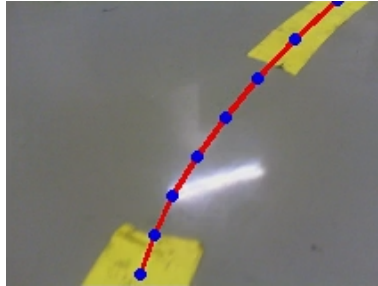
(b) Distúrbio - Método 2.



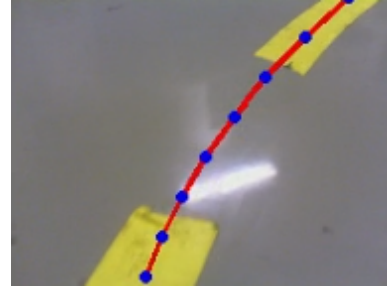
(c) Distúrbio - Método 3.



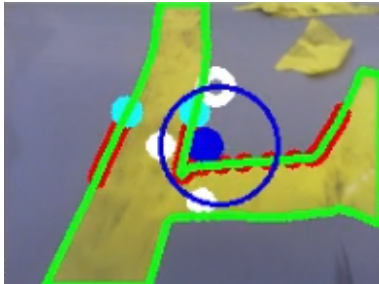
(d) Falhas - Método 1.



(e) Falhas - Método 2.



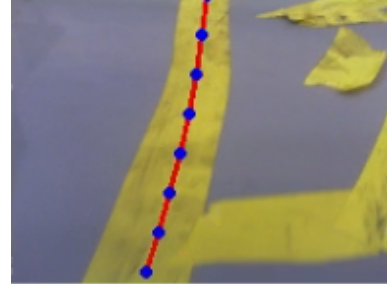
(f) Falhas - Método 3.



(g) Mais Distúrbio - Método 1.



(h) Mais Distúrbio - Método 2.



(i) Mais Distúrbio - Método 3.

Figura 5.25: Alguns dos inconvenientes encontrados pelos três Métodos.

Os erros (Z e θ_r) e as ações de controle (v e ω), para o período em que os três métodos funcionaram, geram os dados mostrados na Tabela 5.2. Analisando os índices relacionados à implementação do NMPC, percebe-se que, para os erros de caminho, o Método 2 e Método 3 são superiores ao Método 1.

A Tabela 5.3 mostra os índices do Método 2 e do Método 3 para todo o experimento, onde é possível ver que, para os erros de caminho, as métricas são praticamente as mesmas e para as ações de controle, como na Tabela 5.2, o RANSAC obteve valores mais altos. Os esforços de controle foram maiores para o Método 3 devido às melhores estimativas da função do caminho que requerem ações mais agressivas para segui-lo corretamente,

isso é mais claro ao analisar as Figs. 5.25(h) e 5.25(i), onde para o Método 2 o caminho estimado gera uma ação de controle que não reflete a realidade, ou seja, embora o caminho seja quase uma linha reta (que gera valores baixos de TV), está incorreto porque não reflete a posição real do caminho na imagem.

	IAE		TV	
	$Z_{iae}[m]$	$\theta_{iae}[rad]$	$v_{tv}[m/s]$	$\omega_{tv}[rad/s]$
Método 1	1.14	2.68	4.93	11.98
Método 2	0.81	2.15	4.82	12.43
Método 3	0.86	2.48	4.93	17.10

Tabela 5.2: Análise Quantitativa até a primeira falha no caminho (*gap*).

	IAE		TV	
	$Z_{iae}[m]$	$\theta_{iae}[rad]$	$v_{tv}[m/s]$	$\omega_{tv}[rad/s]$
Método 2	1.30	3.79	8.03	19.75
Método 3	1.23	3.78	8.50	25.80

Tabela 5.3: Análise Quantitativa do Experimento Completo.

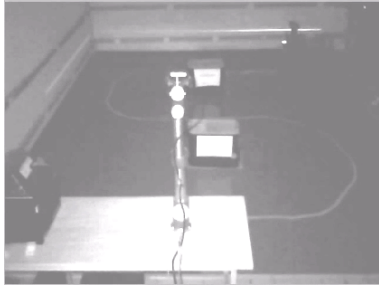
Por fim, uma análise desses métodos foi feita no mesmo ambiente, mas agora com más condições de iluminação, mais especificamente no escuro. Algumas imagens encontradas neste cenário podem ser vistas na Fig. 5.26, onde é possível observar, mais uma vez, a superioridade do Método 3 sobre os demais em termos de estimativa de trajetória. Com o Método 1, o caminho não é detectado e o robô não se move (Figs. 5.26(a), 5.26(b) e 5.26(c)). Com o Método 2, o caminho é perdido no primeiro trecho curvo (Figs. 5.26(d), 5.26(e) e 5.26(f)). Somente a abordagem baseada no RANSAC passa por trechos curvos (Figs. 5.26(g), 5.26(h) e 5.26(i)) e segue o caminho inteiramente, mesmo na presença das imperfeições no trajeto.

Fig. 5.27 fornece os resultados analíticos deste experimento. Nas Figs. 5.27(a) e 5.27(b), é visto que o Método 1 pára logo no início do experimento, gerando altos valores de curvatura. Na abordagem baseada nos Mínimos Quadrados (Método 2), as curvaturas estimadas, apesar de pouca variação, são causadas pela má representação na estimativa do caminho e, portanto, também pára ao encontrar a primeira seção da curva. A estratégia baseada no RANSAC (Método 3) foi a única que pôde concluir todo o percurso, pois estimou os caminhos corretamente com mais frequência do que a estratégia dos Mínimos Quadrados. Na Fig. 5.28 é possível notar a superioridade do Método 3 na estimativa da função de caminho (lidando com ruído) no *frame* imediatamente antes do Método 2 parar. Isso mostra, junto com o que é ilustrado na Fig. 5.26(f), que não há atividade do controlador, mesmo com os erros aumentando até o limite das restrições.

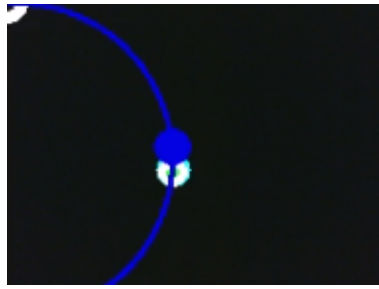
A Figura 5.27(c) mostra os erros de caminho. Observa-se que os erros gerados pelo Método 2, que apresentaram valores muito baixos, foram causados pela deturpação da função de caminho. Por outro lado, o Método 3 gerou mais erros, porém conseguiu percorrer todo o ambiente.

A Figura 5.27(d) mostra que ambos os métodos, na medida em que funcionaram, foram bem regulados, respeitando as capacidades físicas do Husky.

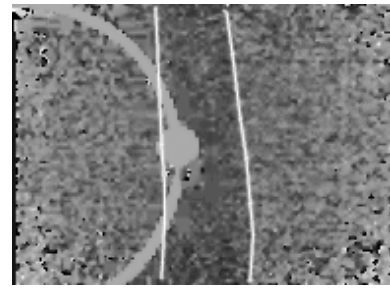
Em comparação com o ambiente mais iluminado, é necessário muito mais esforço dos controladores para o robô seguir o caminho desejado, devido à dificuldade de estimar uma função caminho pelos inconvenientes da má iluminação que acarreta aumento na quantidade de ruído nas imagens. No entanto, uma condição tão extrema de avaliação ilustra que o Método 3 melhora consideravelmente a robustez global do sistema visual.



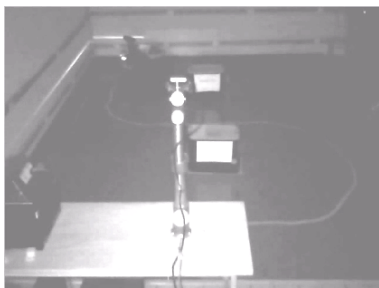
(a) Método 1 no início do trajeto.



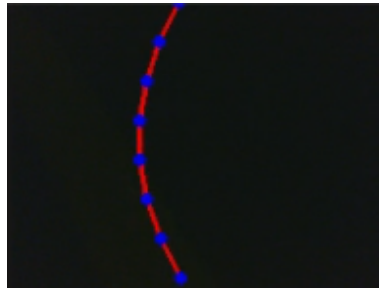
(b) Método 1 - Câmera do Robô.



(c) Método 1 - Componente H_c .



(d) Método 2 na primeira curva.



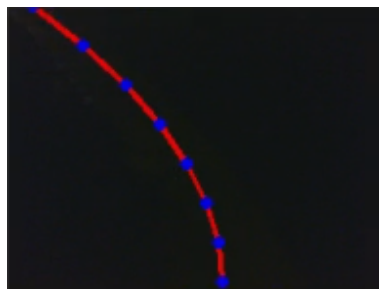
(e) Método 2 - Câmera do Robô.



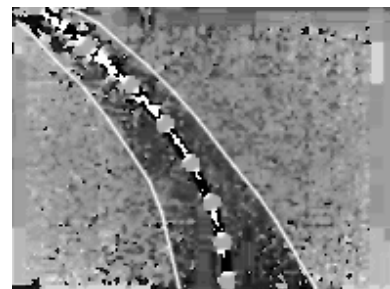
(f) Método 2 - Componente H_c .



(g) Método 3 na primeira curva.

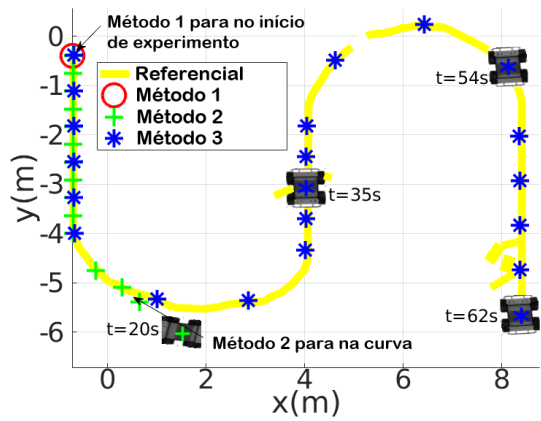


(h) Método 3 - Câmera do Robô.

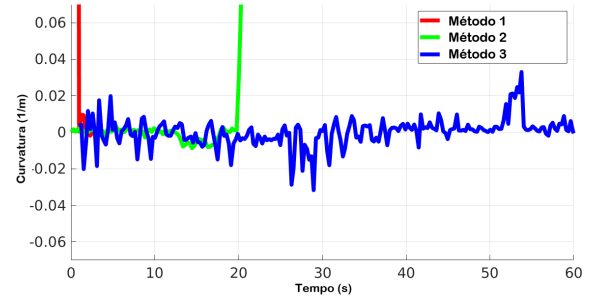


(i) Método 3 - Componente H_c .

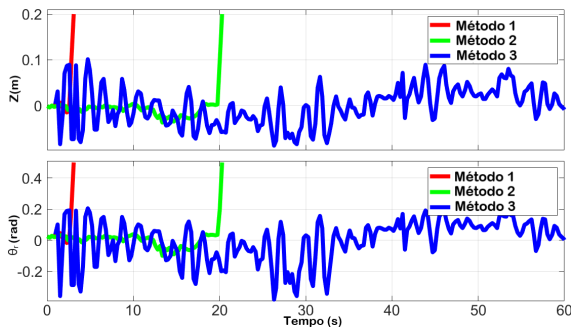
Figura 5.26: Comparação dos Métodos - Navegação ao longo do cenário com baixa luminosidade.



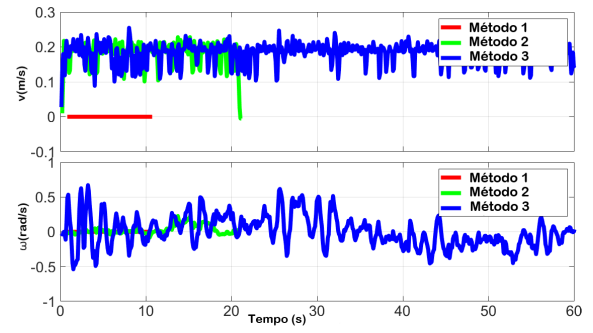
(a) Posições Instantâneas.



(b) Curvatura Visual.

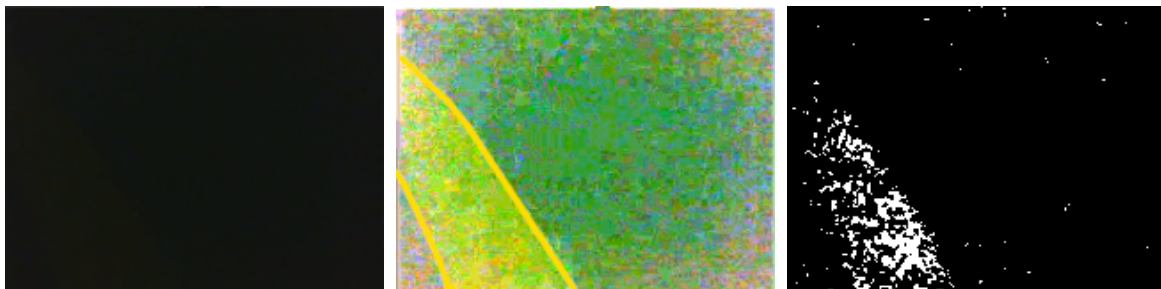


(c) Erros de Seguimento.



(d) Esforços de Controle.

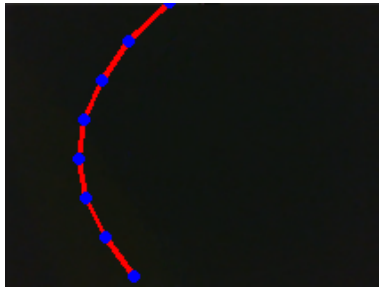
Figura 5.27: Comparação das abordagens em ambiente com má condição de iluminação.



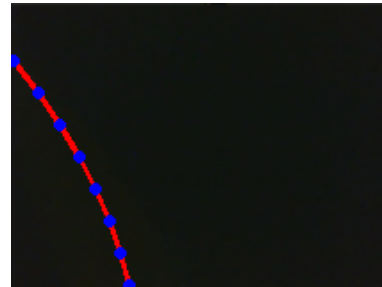
(a) Primeira curva do trajeto.

(b) Delimitação do caminho na imagem (usado apenas para visualização).

(c) Máscara Binária gerada pela segmentação.



(d) Estimação usando o Método 2.



(e) Estimação usando o Método 3.

Figura 5.28: Estimação dos caminhos (MMQ vs RANSAC) onde o Método 2 pára.

CONCLUSÃO

Este trabalho propôs um estudo e desenvolvimento de sistemas visuais para detecção e extração de caminhos diretamente do plano da imagem, através de técnicas de visão computacional para alimentar controladores para seguimento de caminhos visuais baseados em NMPC.

Com respeito ao espaço de cores HSV, percebe-se que o parâmetro de matiz (*Hue*) é robusto para detecção de cor, juntamente com a saturação, mesmo em ambientes mal iluminados. O parâmetro de valor/brilho apesar de não afetar diretamente na segmentação, influencia os valores de saturação e matiz. Outro ponto do espaço HSV diz respeito à calibração que, no escuro viu-se em uma espécie de *trade-off* entre caminho válido e ruído, ou seja, aceitando uma maior faixa para segmentação também aceita-se mais pixels ruidosos, contudo quanto menor a faixa menor o ruído aceito, porém também, menos chances de aceitar pixels válidos do caminho (o que pode granular a imagem). Isso, como se pôde ver, influencia no Método 1 pois este não lida bem com ruídos e no Método 2, por causa da sensibilidade não tratada pelo Método dos Mínimos Quadrados.

Comparando os três métodos, conclui-se que a abordagem usada no Método 1 para aquisição de características visuais e para o cálculo a curvatura geram inconsistências, o que compromete o desempenho da aplicação diante de distúrbios e falhas (*gaps*) no caminho. Essa abordagem, por adotar a técnica baseada em regiões conectadas, mostra dificuldade ao extrair o caminho quando exposto a más condições de iluminação, pois no escuro, na segmentação, o caminho é dividido em várias regiões conectadas e não em uma única, como esperado.

O método baseado nos Mínimos Quadrados (Método 2) resolve parcialmente os problemas gerados pelo Método 1. Nos experimentos, a adoção de uma função de caminho permitiu que a curvatura fosse estimada com mais precisão, no entanto, em cenas muito deterioradas pelo escuro, não foi possível estima-los corretamente devido à sensibilidade ao ruído, que é aumentada devido as condições adversas de iluminação.

A abordagem baseada no RANSAC (Método 3) provou ser superior às demais devido à capacidade de resistir melhor ao ruído e estimar melhor a função caminho, fornecendo parâmetros mais confiáveis ao controlador NMPC. Comparado com o Método 2, os valores

de curvatura gerados variaram menos frente a distúrbios e ruídos, assim como no ambiente escuro, a função foi melhor estimada.

O Método 3, pela sua superioridade, demonstra um potencial maior para aplicações autônomas, como: execução de tarefas coordenadas num armazém ou indústria, tarefas de inspeção e monitoramento, sistemas de carros autônomos e veículos aéreos.

Como trabalhos futuros, considera-se uma investigação detalhada da eficiência computacional do método RANSAC, avaliação de desempenho em ambientes externos, modificações do modelo do seguidor de caminhos visuais baseados em NMPC para a inclusão de horizonte visual variável no tempo e investigação de abordagens utilizando outros espaços de cores ou técnicas baseadas em aprendizado de máquina.

6.1 TRABALHOS PUBLICADOS

Este trabalho de mestrado gerou publicação em conferências como:

- SBAI 2019:

Andre Gustavo Scolari Conceicao and Tiago Trindade Ribeiro and Iago José Pattas Bastos Franco, 2020, Uma Nova Abordagem para Extração de Parâmetros do Modelo de Seguidores Visuais Baseado em NMPC aplicado ao Husky UGV. Em: Anais do 14^o Simpósio Brasileiro de Automação Inteligente. Campinas: Galoá. 2019.

- ICAR 2019:

I. J. P. B. Franco, T. T. Ribeiro and A. G. S. Conceição, “A Novel Approach for Parameter Extraction of an NMPC-based Visual Follower Model,” 2019 19th International Conference on Advanced Robotics (ICAR), Belo Horizonte, Brazil, 2019, pp. 117-122.

REFERÊNCIAS BIBLIOGRÁFICAS

- Arakeri, M. P.; Vijaya Kumar, B. P.; Barsaiya, S.; Sairam, H. V. Computer vision based robotic weed control system for precision agriculture. In: *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.: s.n.], 2017. p. 1201–1205.
- Borkar, A.; Hayes, M.; Smith, M. T. Robust lane detection and tracking with ransac and kalman filter. In: *2009 16th IEEE International Conference on Image Processing (ICIP)*. [S.l.: s.n.], 2009. p. 3261–3264. ISSN 2381-8549.
- Boyd, S.; Vandenberghe, L. *Introduction to Applied Linear Algebra*. Cambridge University Press, 2018. Disponível em: <https://doi.org/10.1017/9781108583664>.
- Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
- Cai, Z.; CHIN, T.-J.; LE, H.; SUTER, D. Deterministic consensus maximization with biconvex programming. In: *Computer Vision – ECCV 2018*. Springer International Publishing, 2018. p. 699–714. Disponível em: https://doi.org/10.1007/978-3-030-01258-8_42.
- Castano, M.; Tan, X. Model predictive control-based path-following for tail-actuated robotic fish. *Journal of Dynamic Systems, Measurement, and Control*, v. 141, 03 2019.
- Cherubini, A.; Chaumette, F.; Oriolo, G. An image-based visual servoing scheme for following paths with nonholonomic mobile robots. In: *2008 10th International Conference on Control, Automation, Robotics and Vision*. [S.l.: s.n.], 2008. p. 108–113.
- Clearpath Robotics. *Husky - Unmanned Ground Vehicle*. 2020. Disponível em: <https://www.clearpathrobotics.com/husky-unmanned-ground-vehicle-robot>.
- Corke, P. I. *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. 1^a. ed. [S.l.]: Springer Science and Business Media, 2011.
- de Lima, D. A.; Victorino, A. C. A visual servoing approach for road lane following with obstacle avoidance. In: *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2014. p. 412–417. ISSN 2153-0017.
- Derpanis, K. G. Overview of the ransac algorithm. In: . [S.l.]: ImageRochester NY, Vol. 4, 2010. p. 2–3.
- Du, X.; Tan, K. K.; Htet, K. K. K. Vision-based lane line detection for autonomous vehicle navigation and guidance. In: *2015 10th Asian Control Conference (ASCC)*. [S.l.: s.n.], 2015. p. 1–5.

Faulwasser, T.; FINDEISEN ROLF ADN MAGNI, L.; RAIMONDO, D. M.; ALLGÖWER, F. Nonlinear model predictive path-following control. In: _____. *Non-linear Model Predictive Control: Towards New Challenging Applications*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. p. 335–343. ISBN 978-3-642-01094-1.

Fischler, M. A.; Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, Association for Computing Machinery, New York, NY, USA, v. 24, n. 6, p. 381–395, jun. 1981. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/358669.358692>.

Forsyth, D. A.; Ponce, J. *Computer Vision - A Modern Approach, Second Edition*. [S.l.: s.n.], 2011.

Gorbunov, V.; BOBKOV, V.; HTET, N. W.; IONOV, E. Automated control system of fabrics parameters that uses computer vision. *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, p. 1728–1730, 2018.

Guennebaud, G.; Jacob, B. et al. *Eigen v3*. 2010. [Http://eigen.tuxfamily.org](http://eigen.tuxfamily.org).

Heydarzadeh, M.; Dusmez, S.; Nourani, M.; Akin, B. Remaining useful lifetime estimation for power mosfets under thermal stress with ransac outlier removal. *IEEE Transactions on Industrial Informatics*, PP, 02 2017.

Huang, A. S.; Moore, D.; Antone, M.; Olson, E.; Teller, S. Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, Springer Science and Business Media LLC, v. 26, n. 2-3, p. 103–122, mar. 2009. Disponível em: <https://doi.org/10.1007/s10514-009-9113-3>.

Jang, H.-J.; Baek, S.-H.; Park, S.-Y. Lane marking detection in various lighting conditions using robust feature extraction. In: . [S.l.: s.n.], 2014.

Jung, C. R.; Kelber, C. R. An improved linear-parabolic model for lane following and curve detection. In: *XVIII Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAP'05)*. [S.l.: s.n.], 2005. p. 131–138. ISSN 2377-5416.

Kim, Z. Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, v. 9, n. 1, p. 16–26, March 2008. ISSN 1558-0016.

Lee, D. Q. *NUMERICALLY EFFICIENT METHODS FOR SOLVING LEAST SQUARES PROBLEMS*. 2012. 1-15 p.

Lee, J.-W.; Cho, J.-S. Effective lane detection and tracking method using statistical modeling of color and lane edge-orientation. In: *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*. IEEE, 2009. Disponível em: <https://doi.org/10.1109/iccit.2009.81>.

- Lu, W.; Florez, S. A. R.; Seignez, E.; Reynaud, R. An improved approach for vision-based lane marking detection and tracking. In: *2013 International Conference on Electrical, Control and Automation Engineering, Dec 2013, Hong Kong, China. DEStech Publications*. [S.l.: s.n.], 2014. p. 382–386.
- López, A.; Serrat, J.; Cañero, C.; Lumbreras, F.; Graf, T. Robust lane markings detection and road geometry computation. *International Journal of Automotive Technology*, Springer Science and Business Media LLC, v. 11, n. 3, p. 395–407, maio 2010. Disponível em: <https://doi.org/10.1007/s12239-010-0049-6>.
- Mammeri, A.; Boukerche, A.; Lu, G. Lane detection and tracking system based on the msr algorithm, hough transform and kalman filter. In: *MSWiM '14*. [S.l.: s.n.], 2014.
- Mammeri, A.; Boukerche, A.; Tang, Z. A real-time lane marking localization, tracking and communication system. *Computer Communications*, v. 73, p. 132 – 143, 2016. ISSN 0140-3664. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0140366415003096>.
- McCall, J. C.; Trivedi, M. M. Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE Transactions on Intelligent Transportation Systems*, v. 7, n. 1, p. 20–37, March 2006. ISSN 1558-0016.
- Plataniotis, K. N.; Venetsanopoulos, A. N. *Color Image Processing and Applications*. Springer Berlin Heidelberg, 2000. Disponível em: <https://doi.org/10.1007/978-3-662-04186-4>.
- Raguram, R.; Frahm, J.-M.; Pollefeys, M. A comparative analysis of ransac techniques leading to adaptive real-time random sample consensus. In: FORSYTH, D.; TORR, P.; ZISSERMAN, A. (Ed.). *Computer Vision – ECCV 2008*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 500–513.
- Ribeiro, T. T.; Conceição, A. G. S. Nonlinear model predictive visual path following control to autonomous mobile robots. *Journal of Intelligent & Robotic Systems*, v. 95, n. 2, p. 731–743, ago. 2019. Disponível em: <https://doi.org/10.1007/s10846-018-0896-3>.
- Ribeiro, T. T.; Franco, I. J. P. B.; Fenandez, R. O.; Conceição, A. G. S. Seguimento de caminhos visuais: Uma abordagem para identificação de parâmetros através de visão computacional. In: *Proceedings XXII Congresso Brasileiro de Automática*. SBA Sociedade Brasileira de Automática, 2018. Disponível em: <http://dx.doi.org/10.20906/CPS/CBA2018-0848>.
- Sonka, M.; Hlavac, V.; Boyle, R. *Image Processing, Analysis and Machine Vision*. Thomson-Engineering, 2007. Disponível em: <https://doi.org/10.1007/978-1-4899-3216-7>.
- Stanford Artificial Intelligence Laboratory et al. *Robotic Operating System*. 2016. Disponível em: <https://www.ros.org>.

Su, Q. *Color Image Watermarking: Algorithms and Technologies (Advances in Computer Science)*. de Gruyter, 2016. ISBN 3110487578. Disponível em: <https://www.xarg.org/ref/a/3110487578/>.

Szeliski, R. *Computer Vision*. Springer London, 2011. Disponível em: <https://doi.org/10.1007/978-1-84882-935-0>.

Tianqi, L. A review of lane perception and automobile control based on computer vision. In: *2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE)*. [S.l.: s.n.], 2017. p. 6–11.

Tran, T.-T.; Bae, C.-S.; Kim, Y.-N.; Cho, H.-M.; Cho, S.-B. An adaptive method for lane marking detection based on HSI color model. In: *Communications in Computer and Information Science*. Springer Berlin Heidelberg, 2010. p. 304–311. Disponível em: https://doi.org/10.1007/978-3-642-14831-6_41.

Vadivel, A.; Sural, S.; Majumdar, A. Human color perception in the hsv space and its application in histogram generation for image retrieval. *Proceedings of SPIE - The International Society for Optical Engineering*, 01 2005.

Wang, J.; Hong, W.; Gong, L. Lane detection algorithm based on density clustering and ransac. In: *2018 Chinese Control And Decision Conference (CCDC)*. [S.l.: s.n.], 2018. p. 919–924. ISSN 1948-9447.

Wang, Y.; Dahnoun, N.; Achim, A. A novel system for robust lane detection and tracking. *Signal Processing*, Elsevier BV, v. 92, n. 2, p. 319–334, fev. 2012. Disponível em: <https://doi.org/10.1016/j.sigpro.2011.07.019>.

Wang, Y.; Teoh, E. K.; Shen, D. Lane detection and tracking using b-snake. *Image and Vision Computing*, Elsevier BV, v. 22, n. 4, p. 269–280, abr. 2004. Disponível em: <https://doi.org/10.1016/j.imavis.2003.10.003>.

Yan Jiang; Feng Gao; Guoyan Xu. Computer vision-based multiple-lane detection on straight road and in a curve. In: *2010 International Conference on Image Analysis and Signal Processing*. [S.l.: s.n.], 2010. p. 114–117. ISSN 2156-0129.

Zuliani, M. Ransac for dummies with examples using the ransac toolbox for matlab™ octave and more... In: . [S.l.: s.n.], 2014.