



**Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Mecânica**

**Análise abrangente do controle preditivo
baseado em modelo aplicado em sistemas de
assistência a manutenção de faixa**

James Duvan Garcia Montoya

DISSERTAÇÃO DE MESTRADO
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS MECATRÔNICOS

Brasília
2024

**Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Mecânica**

**Análise abrangente do controle preditivo
baseado em modelo aplicado em sistemas de
assistência a manutenção de faixa**

James Duvan Garcia Montoya

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre em Sistemas Mecatrônicos

Orientador: Prof. Dr. Evandro Leonardo Silva Teixeira
Coorientador: Prof. Dr. André Murilo de Almeida Pinto

Brasília
2024

G769a Garcia Montoya, James Duvan .
Análise compreensiva do controle preditivo baseado em modelo aplicado em sistemas de assistência a manutenção de faixa / James Duvan Garcia Montoya; orientador Evandro Leonardo Silva Teixeira; coorientador André Murilo de Almeida Pinto. -- Brasília, 2024.
111 p.

Dissertação de Mestrado (Programa de Pós-Graduação em Sistemas Mecatrônicos) -- Universidade de Brasília, 2024.

1. Lane Keeping. 2. Controle MPC. 3. ADAS. 4. Solucionador de programas quadráticos. I. Silva Teixeira, Evandro Leonardo , orient. II. Murilo de Almeida Pinto, André , coorient. III. Título

**Universidade de Brasília
Faculdade de Tecnologia
Departamento de Engenharia Mecânica**

**Análise abrangente do controle preditivo baseado em
modelo aplicado em sistemas de assistência a
manutenção de faixa**

James Duvan Garcia Montoya

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre em Sistemas Mecatrônicos

Trabalho aprovado. Brasília, 01 de março de 2024:

Prof. Dr. Evandro Leonardo Silva Teixeira,
FGA/UnB
Orientador

Prof. Dr. André Murilo de Almeida Pinto,
DAT/UFLA
Co-orientador

Prof. Dr. Bruno Augusto Angélico,
EP/USP
Examinador externo

Prof. Dr. Renato Vilela Lopes, FGA/UnB
Examinador interno

Brasília
2024

Agradecimentos

Quero expressar meu mais sincero agradecimento à minha família, cujo apoio inabalável e compreensão foram fundamentais durante minha jornada acadêmica. Também desejo agradecer profundamente ao meu orientador, o professor Evandro Leonardo Silva Teixeira, por sua orientação especializada, dedicação e paciência ao longo da minha pesquisa. Agradeço igualmente ao meu coorientador, André Murilo de Almeida Pinto, por proporcionar valiosas oportunidades de aprendizado e por sua contribuição significativa para o meu desenvolvimento acadêmico. Seu comprometimento foi crucial para o sucesso da minha dissertação de mestrado. Também expresse minha profunda gratidão à Fundação de Desenvolvimento da Pesquisa (FUNDEP) pelo apoio financeiro e logístico para desenvolvimento desta pesquisa.

Resumo

A integração de novas tecnologias nos veículos atuais demonstra a preocupação dos profissionais da indústria automotiva e do governo com a segurança, o desempenho e o conforto da condução. Por esse motivo, o uso de Sistemas Avançados de Assistência à Condução (ADAS) tem ganhado popularidade nos dias atuais. Esses sistemas permitem aumentar a segurança nos veículos, pois auxiliam a execução de diferentes tarefas que podem ser repetitivas ou de grande dificuldade para o ser humano. Uma dessas atividades é a manutenção da faixa de rodagem (LKAs). O LKAS foi desenvolvida para evitar a saída não intencional do centro da faixa. Essa função usa sensores como câmera, GPS e LIDAR para capturar a posição atual do veículo em relação à estrada e tomar uma decisão que será executada por meio do sistema de direção, alterando o ângulo das rodas. O LKAS usa diferentes algoritmos para o processamento das informações provenientes de sensores e também para a tomada de decisões em tempo real. As estratégias de controle devem garantir a segurança do veículo em diferentes situações de risco, como por exemplo mudanças do clima e condições de mudança da estrada, por esse motivo, o controlador preditivo baseado em modelo (MPC) foi escolhido para este trabalho, já que tem a capacidade de lidar com restrições para diferentes variáveis do sistema, como deslocamento lateral ou o ângulo máximo permitido do veículo; No entanto, esse controlador enfrenta um grande desafio na implementação, pois exige uma alta carga computacional. Por esse motivo, nesta pesquisa, foram implementadas estratégias de redução de carga computacional para o controlador MPC, a fim de serem verificados e validados em *Model-In-the-Loop* (MIL) e em *Hardware-In-the-Loop* (HIL) em uma plataforma de simulação em tempo real. Após a realização de simulações foi possível demonstrar o bom comportamento do controle na função LKAS e sua viabilidade para implementação em tempo real, uma vez que foi feita uma comparação entre o controlador de regulação quadrática linear (LQR). Foram propostas duas formulações para os modelos de veículos implementados na função LKAS. Uma delas visa o rastreamento de uma referência já estabelecida, como é comum em sistemas baseados em GPS, enquanto a outra se concentra no uso do estado atual do veículo em relação à posição desejada na pista, mais especificamente, o erro do centro de gravidade do veículo em relação ao centro da faixa. Esses modelos demonstraram estar fortemente relacionados à sensibilidade e às oscilações na resposta do controlador. Além disso, as formulações do controlador MPC em combinação com os diferentes solucionadores QP mostram que a estratégia de parametrização exponencial permite uma redução significativa do tempo de computação do controlador MPC.

Palavras-chave: Lane Keeping. Controle MPC. ADAS. Solucionador de programas quadráticos.

Abstract

The integration of new technologies into current vehicles demonstrates the concern of professionals in the automotive industry and the government with safety, performance, and driving comfort. For this reason, the use of Advanced Driver Assistance Systems (ADAS) has gained popularity in recent days. These systems allow for increased vehicle safety by assisting in the execution of various tasks that may be repetitive or challenging for humans. One of these activities is lane-keeping assistant system (LKAS). As the name suggests, this function was developed to prevent unintentional departure from the lane center. This function utilizes sensors such as cameras, GPS, LIDAR, etc, to capture the vehicle's current position relative to the road and make a decision that will be executed through the steering system by altering the wheel angle. LKAS employs different algorithms for processing information from sensors and also for making real-time decisions. These control strategies must ensure the safety and stability of the vehicle in different risk situations, such as changes in weather and road conditions. Therefore, Model Predictive Control (MPC) has been chosen for this work, which has the ability to handle constraints for different system variables, such as lateral displacement or the maximum allowed vehicle angle. However, this controller faces a significant challenge in implementation as it requires high computational load. For this reason, in this research, computational load reduction strategies were implemented for the MPC controller to be verified and validated in *Model-In-the-Loop* (MIL) and *Hardware-In-the-Loop* (HIL) simulations on a real-time simulation platform. After conducting the respective simulations, which demonstrated the good behavior of the control in the LKAS function and its feasibility for real-time implementation, as a comparison was made with the Linear Quadratic Regulator (LQR) controller. Two formulations were proposed for the vehicle models implemented in the LKAS function, one for tracking an established reference and the other for using the vehicle's current state relative to the lane. These models were found to be strongly related to the sensitivity and oscillations in the controller response. Furthermore, the formulations of the MPC controller in combination with different QP solvers show that the exponential parameterization strategy allows for a significant reduction in the MPC controller's computation time.

Keywords: Lane Keeping. MPC Control. ADAS. QP solvers.

Lista de ilustrações

Figura 1 – Processamento de imagem na função LKAS	22
Figura 2 – Visão real (a), visão de pássaro (b).	22
Figura 3 – Modelo cinemático do veículo (Modelo bicicleta)	26
Figura 4 – Esquema de funcionamento do controle preditivo baseado em modelo. .	31
Figura 5 – Modelos de rastreamento de trajetória	35
Figura 6 – Parametrização trivial (linha azul) e exponencial (linha verde) em todo o horizonte de predição para intervalos de três vezes ($n_r = 3$). Adaptado de (ALAMIR, 2013).	40
Figura 7 – Modelo V.	44
Figura 8 – MIL, SIL, PIL e HIL	46
Figura 9 – Desempenho do controlador MPC com QPSWIFT direita modelo 1 , esquerda modelo 2	66
Figura 10 – Desempenho do controlador LQR com modelo 1 de rastreamento da trajetória.	68
Figura 11 – Resultados do experimento com o controlador MPC clássico a) Erro do ângulo de direção e do deslocamento lateral e b) Tempo computacional dos solucionadores QP.	69
Figura 12 – Resultados de simulação para MPC com parametrização trivial parametrizado, incluindo teste de desempenho de rastreamento de referência .	70
Figura 13 – Parte superior Comando de entrada com erro de deslocamento lateral e parte inferior tempo computacional requerido pelo MPC com parametrização trivial parametrizado.	71
Figura 14 – Resultados de simulação obtidos com o teste de desempenho de rastreamento MPC com parametrização exponencial.	73
Figura 15 – Comando de entrada e erro de deslocamento lateral, parte superior, tempo computacional exigido pelo MPC com parametrização exponencial, parte inferior.	74
Figura 16 – Comparação do tempo de execução de todos os solucionador QPs quadráticos	75
Figura 17 – Maiores alternativas de candidatos a QPs.	76
Figura 18 – Desempenho do controlador MPC e Desempenho do controlador LQR para LKAS.	78
Figura 19 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC clássico	79
Figura 20 – Desempenho do controlador MPC com parametrização trivial para LKAS.	80

Figura 21 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC com parametrização trivial	81
Figura 22 – Desempenho do controlador MPC com parametrização exponencial para LKAS.	83
Figura 23 – Comando de entrada e tempo computacional dos solucionadores quadráticos do controlador MPC com parametrização exponencial	84
Figura 24 – Comparação de todos os solucionadores QP.	85
Figura 25 – Melhor tempo encontrado pela combinação do solucionador e da formulação.	86
Figura 26 – Desempenho do controlador MPC lado esquerdo e Desempenho do controlador LQR lado direito para LKAS.	88
Figura 27 – Comando de entrada e tempo de cálculo para solucionadores quadráticos de controlador MPC	89
Figura 28 – Desempenho do controlador MPC com parametrização trivial para LKAS.	90
Figura 29 – Comando de entrada e tempo de cálculo do solucionador para o controlador trivial mpc	91
Figura 30 – Desempenho do controlador MPC com parametrização exponencial para LKAS.	92
Figura 31 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC com parametrização exponencial	93
Figura 32 – Comparação de todos os solucionadores QP.	95
Figura 33 – Melhor tempo encontrado combinando o solucionador e a formulação	96
Figura 34 – Desempenho do controlador MPC e Desempenho do controlador LQR para LKAS com parâmetros diferentes do modelo (Aumento de massa em um 30% e uma redução do coeficiente de atrito em um 30%).	98

Lista de tabelas

Tabela 1 – Processo de classificação de artigos para leitura integral	51
Tabela 2 – Estado do Arte	64
Tabela 3 – Parâmetros do controlador MPC em MIL	65
Tabela 4 – Dados de entrada para os QPs para o MPC clássico.	68
Tabela 5 – Parâmetros de calibração de parametrização trivial.	71
Tabela 6 – Dados de entrada de QPs o controlador MPC com parametrização exponencial.	72
Tabela 7 – Comparação das diferentes formulações na redução do tempo médio de computação.	75
Tabela 8 – Parâmetros do controlador MPC nos ciclos de testes MIL e HIL para o modelo do erro	77
Tabela 9 – Parâmetros para controlador MPC-MIL	79
Tabela 10 – Parâmetros controle MPC com parametrização trivial MIL	82
Tabela 11 – Parâmetros controle MPC com parametrização exponencial MIL	83
Tabela 12 – Diminuição percentual no tempo médio de cálculo em MIL	85
Tabela 13 – Parâmetros para o modelo do erro do veículo com perturbação em relação à pista	87
Tabela 14 – Parâmetros para controlador MPC HIL	88
Tabela 15 – Parâmetros para controlador Trivial HIL	91
Tabela 16 – Parâmetros para controlador MPC com parametrização exponencial em HIL	94
Tabela 17 – Comparação das diferentes formulações na redução do tempo médio de computação	95
Tabela 18 – Resumo de tabela de resultados de redução porcentual do tempo	96
Tabela 19 – Novos parâmetros para o modelo do erro do veículo em relação à pista	97

Lista de abreviaturas e siglas

ADAS	Sistemas Avançados de Assistência à Condução	7
ECU	Electronic Control Unit.....	15
GPS	Global Positioning System	24
HIL	Hardware In the Loop	7
IMU	Inertial Measurement Unit.....	24
LIDAR	Light Detection and Ranging.....	54
LKAS	Lane Keeping Assist System	7
LQR	Linear Quadratic Regulator.....	7
MIL	Model In the Loop	7
MPC	Model Predictive Control	30
NHTSA	National Highway Traffic Safety Administration	15
PID	Proportional Integral Derivative	16
RANSAC	RANdom SAMple Consensus	55

Sumário

1	INTRODUÇÃO	15
1.1	Contribuições desta dissertação	17
1.2	Objetivo	17
2	REVISÃO DA LITERATURA	19
2.1	Sistemas Avançados de Assistência ao Condutor	19
2.2	Sistema de Assistência à Permanência na Faixa	19
2.3	Monitoramento da cena de trânsito	21
2.3.1	Pré-processamento de imagem	21
2.3.1.1	Extração das linhas	22
2.3.1.2	Levantamento de modelos	23
2.3.2	Algoritmos de <i>End-to-End</i>	24
2.4	Sensores para a função LKAS	24
2.5	Dinâmica veicular	25
2.5.1	Modelo cinemático	26
2.5.2	Dinâmica lateral	27
2.5.2.1	Modelo 1 de rastreamento de trajetória	28
2.5.2.2	Modelo 2 de rastreamento de trajetória	28
2.5.2.3	Modelo 3 em função do erro	29
2.6	Estratégias de controle	30
2.7	Formulação do controle MPC	31
2.7.1	Formulação do MPC para modelos de rastreamento de trajetória	32
2.7.1.1	Função de custo	33
2.7.1.2	Restrições impostas ao controlador	34
2.7.2	Formulação do MPC para modelo com perturbação	36
2.7.2.1	Função de custo	37
2.7.3	Definição das Restrições	38
2.7.4	Parametrização do controlador MPC	39
2.7.5	O controlador linear quadrático	42
2.8	Ciclo de desenvolvimento do produto automotivo	44
2.8.1	Solucionadores quadráticos	46
3	ESTADO DO ARTE	49
3.1	Planejamento	49
3.2	Coleta de dados	49
3.3	Análise temática	52

3.3.1	Tema 1: Definição e funcionamento do LKAS	52
3.3.2	Tema 2: Sensoriamento para a função LKAS	54
3.3.3	Tema 3: Modelagem, simulação e estratégias de controle do LKAS	56
3.3.4	Tema 4: Benefícios e limitações das LKAS	61
4	ESTUDOS DE CASOS	65
4.1	Modelos de rastreamento de trajetória em <i>Model-In-the-Loop</i> . . .	65
4.1.1	Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC clássico.	65
4.1.2	Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC com parametrização trivial.	70
4.1.3	Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC com parametrização exponencial.	72
4.1.4	Análise comparativa das soluções	74
4.2	Análise comparativa do erro de posição lateral e do erro de posição de guinada em <i>Model-In-the-Loop</i>	76
4.2.1	Controlador MPC com formulação clássico em <i>Model-In-the-Loop</i>	76
4.2.2	Controlador MPC com parametrização trivial em <i>Model-In-the-Loop</i>	78
4.2.3	Controlador MPC com parametrização exponencial em <i>Model-In-the-Loop</i>	81
4.2.4	Análise comparativa dos controladores MPCs em <i>Model-In-the-Loop</i>	84
4.3	Análise comparativa do erro de posição lateral e do erro de posição de guinada em <i>Hardware-In-the-Loop</i>	86
4.3.1	Análise do controlador MPC clássico em <i>Hardware-In-the-Loop</i>	87
4.3.2	Análise do controlador MPC com parametrização Trivial em <i>Hardware-In-the-Loop</i>	89
4.3.3	Análise do controlador MPC com parametrização exponencial em <i>Hardware-In-the-Loop</i>	92
4.3.4	Análise comparativa dos controladores MPCs em <i>Hardware-In-the-Loop</i>	94
4.3.5	Análise da redução percentual	96
4.3.6	Análise de robustez em <i>Hardware-In-the-Loop</i>	97
5	CONCLUSÕES	99
5.0.1	Limitações da pesquisa	100
5.0.2	Trabalhos futuros	100
	REFERÊNCIAS	102

1 Introdução

O automóvel é um dos produtos que mudou o modo de vida e o desenvolvimento do mundo como se conhece hoje. Há várias décadas, diferentes atores, especialmente governos e a indústria automobilística, uniram forças para desenvolver funções que auxiliem o condutor, devido ao crescimento do uso diário desse meio de transporte, e o crescente número de acidentes de trânsito. As estatísticas mostram que o maior índice de acidentes de trânsito é causado por erro humano, seja por fadiga, distração, intoxicação ou carga de trabalho, segundo o *National Highway Traffic Safety Administration* (NHTSA). Essas falhas humanas têm motivado a realização de pesquisa tanto no meio acadêmico quanto no setor industrial, impulsionados pela necessidade de desenvolver sistemas avançados de assistência capazes de auxiliarem os condutores em situações adversas (NHTSA, 2024).

Tanto os pesquisadores quanto os profissionais do setor automotivo têm trabalhado intensamente no desenvolvimento dos chamados sistemas avançados de assistência à condução (ADAS) (SON; LEE, S.-H.; CHUNG, 2013). Esses sistemas englobam funções que operam de forma reativa (passiva) ou proativa (ativa) e são desenvolvidos, para evitar um acidente de trânsito, ou até mesmo reduzir seus efeitos (RATHAI; AMIRTHALINGAM; JAYARAMAN, 2017). Essas funções permitem mitigar as principais falhas humanas ao dirigir, como regulação do ângulo do volante, aviso de ponto cego, ajuste da velocidade da roda, deslocamentos laterais, manutenção da faixa de rodagem, identificação de possíveis acidentes, etc. O uso de sistemas ADAS têm se intensificado nos últimos anos por meio de regulamentações emitidas pelos governos que buscam ter um impacto positivo na diminuição de acidentes fatais ou de seus efeitos colaterais.

Em geral, os sistemas ADAS utilizam uma combinação de diferentes tipos de sensores que lhes permitem perceber o ambiente ao redor do veículo. Por meio de algoritmos de controle, avaliam e tomam a melhor decisão para evitar situações de risco ou acidentes (LIANG, J. et al., 2021). A função de assistência à manutenção de faixa (LKAS) desempenha um papel importante na segurança ao manter o veículo nos limites estabelecidos pela faixa de rolagem. A função LKAS geralmente utiliza uma câmera para obter informações sobre a faixa e os arredores do veículo. Após processar essas informações e realizar os cálculos para manter o veículo no centro da faixa ela verifica o *status* da sinalização do condutor; caso não esteja ativada e o veículo estiver se afastando da faixa de rodagem, um comando com o valor do ângulo ou do torque de volante é enviado à Unidade de Controle Eletrônica (ECU) responsável pelo gerenciamento do sistema de direção (KIM, D. J. et al., 2018). O sistema de direção é então utilizado para ajustar a referência solicitada pelo sistema LKAS para que o veículo retorne ao centro da faixa.

Essa função de controle lateral usa diferentes algoritmos para obter o melhor desempenho na permanência no centro da faixa. Os algoritmos de controle usados devem garantir a permanência do veículo dentro das restrições de segurança impostas (LEE, J. et al., 2014). Restrições como o deslocamento máximo permitido em relação ao centro da faixa, o ângulo máximo da roda ou a velocidade na qual as mudanças no ângulo da roda, em geral, são consideradas pelo sistema LKAS. Diferentes estratégias de controle foram propostas na literatura, como o controle PID, controle fuzzy, controle LQR e MPC, entre outros (SALT DUCAJÚ et al., 2021). O controlador MPC permite o uso de restrições de sistemas controlados e usa um modelo matemático que utiliza para a otimização de uma sequência de ações por meio de um processo de otimização e penaliza a variável de estado do sistema e a variável de comando de controle por meio de uma equação de custo (CHEN, B.-C.; LUAN; LEE, K., 2014). Essa estratégia de controle otimizado necessita executar cálculos em um tempo menor do que o tempo de amostragem do sistema. Aliado a isso, o processamento em tempo real dos sensores (câmera, lidar, radar e sensores ultrassônicos) exige altos recursos computacionais (AN et al., 2020), o que representa um desafio para a implementação em sistemas de grande escala.

Geralmente, as funções ADAS utilizam uma elevada quantidade de dados provenientes dos sensores para avaliar o ambiente. Somando-se a isso uma maior capacidade de processamento de dados são necessárias nos processos de computação, neste caso, para a função LKAS, o processamento de imagens em tempo real e o uso de uma estratégia de controle que utiliza um processo de otimização enfrentam o desafio da carga computacional em um hardware limitado, o que ressalta a importância da redução computacional na execução de estratégias de controle avançadas como o MPC na função LKAS. Geralmente, as funções ADAS lidam com grandes volumes de dados provenientes dos sensores para a avaliação do ambiente. Além disso, é exigida uma maior capacidade de processamento de dados nos processos computacionais. Especificamente para a função LKAS, o processamento em tempo real de imagens e a aplicação de uma estratégia de controle baseada em otimização enfrentam o desafio da carga computacional em hardware limitado. Isso realça a importância da redução computacional na implementação de estratégias de controle avançadas, como o MPC na função LKAS.

Outro desafio imposto pelo uso de estratégias de controle avançadas é a validação e a análise do comportamento antes de sua implementação em um sistema comercial. Na atualidade existem metodologias para o desenvolvimento de produtos tecnológicos, como o modelo V (KIM, W.; SON; CHUNG, 2015), por meio da implementação de testes em cada etapa de projeto que permite a execução de diferentes testes, como a conhecida *Model-In-the-Loop* (MIL) que é realizada nos estágios iniciais do desenvolvimento do sistema ou simulação de *hardware-in-the-loop* (HIL) executada nos últimos estágios do processo de construção para validar o sistema desenvolvido antes de levá-lo ao sistema real (PFEFFER, P. et al., 2018). Essas metodologias permitem o desenvolvimento de sistemas complexos.

1.1 Contribuições desta dissertação

Para contribuir com o desenvolvimento da função LKAS, neste trabalho foram desenvolvidas e propostas duas estratégias principais para reduzir a carga computacional do controlador MPC. A primeira estratégia envolve a formulação matemática do controlador MPC em duas formas distintas: a forma trivial e a forma exponencial. Essas formulações são baseadas na proposta apresentada por Alamir ([ALAMIR, 2013](#)), visando simplificar o processo computacional e torná-lo mais eficiente.

Por outro lado, a segunda estratégia se baseia no uso de diferentes algoritmos de otimização disponíveis na literatura. Esses algoritmos incluem o método do conjunto ativo, o método do ponto interior, o método de Newton, o método de divisão multiplicativa, entre outros. A aplicação desses algoritmos tem o propósito de encontrar soluções ótimas de forma mais eficiente, levando em consideração a complexidade do problema e as características específicas do sistema LKAS. Esses métodos podem ser utilizados de forma isolada ou combinada, dependendo das necessidades e requisitos do sistema em questão.

Dois modelos de rastreamento de referência que utilizam uma câmera para impor limites de deslocamento lateral são propostos e usados para avaliar as estratégias de redução de carga computadas em um primeiro estágio. Adicionalmente, é proposto o uso de um terceiro modelo de erro em relação à faixa de rolagem que utiliza uma câmera para a coleta de informações como o deslocamento lateral, o ângulo do veículo em relação ao centro da faixa e a curvatura da faixa, que representarão uma perturbação do sistema. Além disso, também são propostas duas fórmulas para a função usada no processo de otimização, a primeira para os modelos de rastreamento de referência com base na fórmula exposta em ([ALAMIR, 2013](#)) e segunda para o modelo de erro com base em ([WANG, L., 2009](#)). O uso de modelos diferentes fornecerá uma perspectiva mais ampla de como a formulação do modelo afeta significativamente o desempenho do controlador.

Neste trabalho, ciclos de verificação e validação MIL e HIL são usados para avaliar o desempenho da estratégia de controle com suas respectivas estratégias de redução de carga computacional. O desenvolvimento das simulações é proposto para cada modelo de forma diferente e para a futura implementação em hardware com o uso de diferentes algoritmos de otimização.

1.2 Objetivo

Este trabalho tem como principal objetivo realizar uma análise compreensiva do controle preditivo baseado em modelo aplicado em sistemas de assistência a manutenção de faixa.

Em termos de objetivos específicos pode-se destacar:

1. Implementar estratégias do MPC para LKAS utilizando diferentes representações de modelos de dinâmica veicular, garantindo que estas estratégias sejam eficazes na correção da trajetória do veículo.
2. Desenvolver técnicas destinadas a reduzir a carga computacional do MPC através da parametrização adequada das sequências de controle. Este objetivo inclui a apresentação e avaliação de duas abordagens teóricas de parametrização: uma trivial e outra exponencial.
3. Realizar experimentos e uma análise comparativa de vários solucionadores de Programação Quadrática (QP) para avaliar tanto a eficiência do controlador como o tempo de computação resultante.
4. Aplicar técnicas de parametrização em vários solucionadores QP estabelecidos, efetuando comparações exaustivas em termos de desempenho do controlador e carga computacional entre diferentes formulações do MPC.
5. Analisar comparações em tempo real de soluções baseadas em MPC para LKAS e destacar áreas de trabalho futuro que possam melhorar ainda mais a eficiência e a eficácia destes sistemas.

Esta dissertação está organizada da seguinte forma: no capítulo dois, é apresentada uma revisão da literatura das temáticas que estão correlacionadas ao tema de pesquisa. No capítulo três, é apresentada uma análise do estado da arte atual da função LKAS, com suas principais características, desafios e estratégias de controle. No capítulo quatro são apresentados os cenários de simulação propostos para a validação do controlador e suas estratégias de redução de carga computacional. No capítulo quinto são mostrados os resultados da formulação do controlador, os solucionadores quadráticos usados em ciclos MIL e HIL para os modelos e a comparação com o controlador LQR. No capítulo final são apresentadas as conclusões, as limitações e os trabalhos futuros.

2 Revisão da Literatura

Este capítulo é uma revisão da literatura sobre sistemas ADAS, com foco principal na função LKAS. Ele explora pesquisas recentes, desenvolvimentos tecnológicos e implementações práticas, destacando desafios, soluções propostas e avanços na integração de sensores, técnicas de controle e sistemas de comunicação para melhorar a segurança e eficiência nas estradas.

2.1 Sistemas Avançados de Assistência ao Condutor

Os ADAS estão presentes nos veículos atuais, proporcionando assistência, segurança e conforto aos motoristas. Essas funções surgiram principalmente devido à necessidade de mitigar e reduzir acidentes de trânsito, pois estudos revelam que mais de 90% dos acidentes são causados por erros humanos (ROSEN et al., 2022). Esses erros podem ser causados por diferentes fatores, como sonolência, cansaço ou intoxicação (XU, L. H.; HU, S. G.; LUO et al., 2015). Por essa razão, a indústria automobilística e os governos têm empreendido ações para viabilizar o desenvolvimento de funções que protejam a integridade dos condutores, passageiros e pedestres, motivados principalmente pelo fato de os acidentes terem se tornado um problema de saúde pública, seja pelo número de vítimas, seja pelas consequências na saúde e na qualidade de vida das pessoas envolvidas (ORGANIZATION, 2018).

Na atualidade existem diferentes sistemas de segurança nos veículos com diferentes níveis de ação. Esses sistemas são classificados conforme o momento em que são ativados, em geral, os sistemas são agrupados em passivos e ativos, dentro dos sistemas de segurança passiva podemos encontrar os airbags e cintos, e dentro dos sistemas ativos estão o sistema de freios anti bloqueio (ABS), o controle eletrônico de estabilidade (ESC), a frenagem automática de emergência (AEB), o controle de cruzeiro adaptativo (ACC), o aviso de saída de faixa (LDW) e o assistente de manutenção de faixa (LKA). À medida que as funções de assistência ao condutor evoluem, permitem uma maior flexibilidade na tomada de decisões, e são suportadas por algoritmos e tecnologias sensoriais sofisticadas, isso estabelece uma base progressiva para a autonomia do veículo.

2.2 Sistema de Assistência à Permanência na Faixa

O Sistema de Assistência à Permanência na Faixa (LKAS) atualmente encontrado em veículos modernos, auxilia os motoristas a manter o veículo dentro da faixa, utilizando sensores e câmeras para detectar as linhas na estrada e ajustar a direção do veículo. Esse

sistema pode ser particularmente útil para os condutores que têm a tendência de desviar de sua faixa ou têm dificuldade em manter a posição adequada na pista (SARAÇOĞLU; ÜLEŞ; SCHMIDT, 2018). Há principalmente três níveis diferentes de monitoramento de faixa nos Sistemas avançadas de assistência ao condutor ADAS. No nível mais básico, há o LDW que, monitora a faixa e envia um aviso por meio de sons e luzes quando o veículo está se afastando da faixa sem a indicação de seta. Já no segundo nível, o Sistema LKAS funciona de forma reativa enviando um aviso ao condutor, e, em último caso, atuando no sistema de direção para retornar o veículo para o meio da faixa. Já no nível mais alto de monitoramento é o Assistente de Centralização de Faixa (LCA), que funciona de forma proativa e geralmente trabalha em conjunto com outras funções, como o ACC, e não requer monitoramento constante por parte do usuário para se manter no centro da faixa. O LKAS opera como um sistema de assistência à direção, usando uma combinação de sensores, câmeras e algoritmos de processamento de imagens para monitorar constantemente a posição do veículo em relação às linhas da pista. O principal objetivo do LKAS é detectar e corrigir automaticamente desvios não intencionais do veículo em relação ao centro da pista (XU, X.; WATERS et al., 2017). O LKAS geralmente executa três tarefas principais: (1) processamento de imagens, (2) cálculo de ângulo das rodas dianteiras e (3) acionamento da direção. Inicialmente, os sensores e as câmeras do veículo coletam dados sobre a posição do veículo em relação às linhas da pista. Esses dados são processados para determinar se o veículo está mantendo sua trajetória. Posteriormente, o sistema avalia se o veículo está se desviando da faixa sem que o condutor tenha sinalizado e calcula o ângulo usando uma estratégia de controle. Por fim, se for detectado um desvio indesejado, o LKAS pode intervir aplicando suavemente um torque na direção para corrigir a trajetória e guiar o veículo de volta ao centro da faixa.

A função LKAS utiliza diferentes sensores que trabalham em conjunto, sendo a câmera o principal sensor responsável pelo reconhecimento da faixa de rodagem (KANG; LEE, S.-H.; CHUNG, 2014). Em sistemas avançados, também é utilizado um LIDAR que permite um melhor reconhecimento do ambiente de condução. Além disso, o uso de sensores como radar ou sensores ultrassônicos são recomendados para aplicações com detecção de objetos que permitem melhorar os níveis de segurança do veículo dentro dessa função.

A informação do ambiente é captada pelos sensores e posteriormente enviada para processamento e cálculo das variáveis de controle do veículo como o ângulo da direção ou ações sobre a dinâmica longitudinal. Existe uma grande variedade de estratégias de controle que podem ser implementadas nessa função, como PID, Fuzzy, LQR (GUO, J. et al., 2020; CHU et al., 2018), *End-to-End Deep learning*, MPC, etc. Todas essas estratégias são utilizadas na atualidade, no entanto, o uso de controladores avançados, como MPC ou *End-to-End*, ou a combinação de diferentes estratégias mencionadas anteriormente, tem crescido como forma de garantir a robustez e a estabilidade do veículo (CHEN, Z.; LI, L.; HUANG, X., 2018).

A estratégia de controle é um dos recursos mais importantes, pois permite encontrar o torque ou o ângulo necessário para que o veículo permaneça em uma zona segura. Ao longo dos anos, a função LKAS tem enfrentado diferentes desafios devido a seus altos custos computacionais, desafios também podem estar presentes no desenvolvimento do software ou no nível da estrutura (qualidade) da estrada, pois é difícil garantir que todas as estradas estejam em condições adequadas para o uso da função. Possivelmente os maiores desafios sejam as condições climáticas, pois a luz do sol ou sua posição, neve, chuva ou neblina podem afetar diretamente o funcionamento correto do LKAS (YAN, M. et al., 2021).

Atualmente, há várias regulamentações na Europa, Ásia e América que regem o funcionamento das funções ADAS. Algumas dessas funções, como o sistema ABS, são regulamentadas como obrigatórias em determinados países devido à sua eficácia comprovada na redução de acidentes rodoviários. Além disso, existem programas que avaliam o desempenho dessas funções, sendo um dos mais proeminentes o novo Programa Europeu de Avaliação de Veículos (Euro NCAP)(IQBAL et al., 2021). Este programa avalia as funções de segurança dos veículos disponíveis no mercado ou que serão lançados em breve, fornecendo informações sobre seu nível de segurança. Essas regulamentações e avaliações, realizadas principalmente pelos governos, incentivam e motivam o aprimoramento contínuo dos recursos de segurança implementados nos veículos. Isso, por sua vez, faz com que as empresas desenvolvam e avancem na tecnologia, ao mesmo tempo, em que buscam a redução do custo de desenvolvimento.

2.3 Monitoramento da cena de trânsito

Há diferentes sensores que podem ser usados individualmente ou em conjunto para otimizar a operação de manutenção da faixa de rodagem; no entanto, o principal sensor é a câmera, que adquire informações do ambiente e as envia para que algoritmos possam extrair as informações necessárias em tempo de execução (KUO; LU, Y.; YANG, S., 2019). Em geral, o processo clássico consiste em três estágios: pré-processamento, extração das linhas, levantamento do modelo ou rastreamento como se mostra na figura 1.

2.3.1 Pré-processamento de imagem

A primeira etapa consiste no pré-processamento baseia-se no estabelecimento da região de interesse dentro da imagem capturada pela câmera. Após delimitar essa área, diferentes filtros são aplicados para a remoção de ruído, sendo aplicadas diferentes técnicas que transformam a região de interesse em uma visão aérea para ter uma visão de cima e ter um tratamento mais fácil dos dados capturados. Na figura 2 é representada a transformação em que a parte azul é a visão de interesse (ROI).



Figura 1 – Processamento de imagem na função LKAS

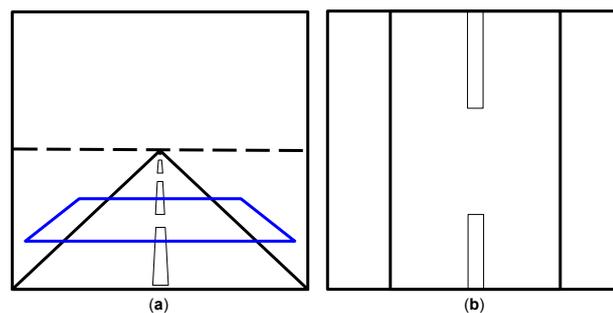


Figura 2 – Visão real (a), visão de pássaro (b).

Adaptado Fonte: (DING et al., 2017)

2.3.1.1 Extração das linhas

No estágio de extração de linhas ou detecção das características da faixa, há diferentes métodos, entre os mais reconhecidos estão os mencionados abaixo:

- O método gradiente se baseia no princípio de que as estradas são delineadas com linhas bem marcadas e reconhecíveis, buscando, assim, encontrar e detectar alterações de intensidade na imagem frequentemente associadas com as bordas da pista. Operadores de detecção de bordas Sobel ou Canny são usados para aprimorar esses recursos e destacar as linhas da pista (WU; WANG, L.-H.; WANG, K.-C., 2018).
- O método baseado em cores utiliza limites de cores para filtrar as linhas de cor amarela ou branca, que atualmente são estritamente regulamentadas em muitos países do mundo. No entanto, essa abordagem pode ser suscetível a alterações na intensidade

da luz ou à presença de sombras, o que pode afetar a detecção de linhas (WANG, H. et al., 2019).

- O método baseado em segmentação busca encontrar regiões que tenham um recurso previamente determinado. Este método pode ser usado para encontrar as linhas objetivas e separá-las de outros elementos diferentes da estrada, como o pavimento (HTET; KIONG; XINXIN, 2015).

É essencial enfatizar que estes métodos não são mutuamente exclusivos e que eles podem ser usados em conjunto para melhorar o desempenho geral de um algoritmo de detecção de linha. A combinação dessas abordagens permite que cada um de seus pontos fortes seja aproveitado e que as possíveis limitações individuais sejam tratadas, contribuindo para uma detecção de faixa mais robusta e precisa.

2.3.1.2 Levantamento de modelos

Para esse estágio, o principal objetivo é determinar, da maneira mais confiável, a posição e a forma da faixa. Para tanto, existem diferentes modelos e técnicas de ajuste disponíveis conforme os requisitos exigidos, por exemplo, os modelos paramétricos são usados para ter uma forte restrição na suavidade da curva da faixa, de modo que geralmente se utiliza um polinômio de terceira ordem. Esta suavidade tem certas vantagens, como a eliminação de ruído; no entanto, isso também pode representar certas desvantagens em estradas com curvas íngremes ou cenários complexos.

Modelos semi paramétricos são aqueles em que há uma combinação de componentes paramétricos e não paramétricos, podendo incluir o modelo linear paramétrico, o modelo linear por partes entre outros. A principal vantagem desse modelo é que ele não está restrito a uma forma geométrica específica, podendo abranger um número maior de cenários. Nesse modelo, a inclusão de restrições deve ser fortemente considerada, pois pode resultar em uma superestimação do modelo ou em uma curvatura inexistente e, como ele contém mais parâmetros em seu modelo, pode gerar uma carga computacional maior.

Por fim, há os modelos não paramétricos que estimam os parâmetros do modelo com base em informações reais existentes, ou seja, não possuem um tipo de modelo previamente estabelecido. Isto permite maior flexibilidade no reconhecimento de novos padrões da via, capturando a variabilidade do ambiente de forma mais ampla. Considerando o exposto acima, esse modelo, por ser adaptável, requer uma quantidade maior de dados, o que, por sua vez, gera uma carga computacional maior, podendo também apresentar uma maior sensibilidade.

Nessa etapa, também ocorre o rastreamento, que é de vital importância, pois permite a integração das informações extraídas nos quadros anteriores e atuais para usá-las a fim de melhorar a precisão e a robustez de todo o sistema. Conforme o mencionado acima, o sistema

obtem robustez contra ruídos de mudanças repentinas ou quadros ausentes, enquanto a correção de erros mantém a confiabilidade devido a fatores que afetam a câmera, como fatores ambientais. Os métodos mais comuns, como o filtro Kalman (BORKAR; HAYES; SMITH, 2011), filtro de Kalman estendido, o filtro de partículas e os mínimos quadrados (BAR-SHALOM; LI, X. R.; KIRUBARAJAN, 2004) estimam o comportamento dinâmico e a percepção, adaptando-se à variabilidade do mundo real.

2.3.2 Algoritmos de *End-to-End*

Os métodos anteriores baseiam-se principalmente na suposição de uma geometria predefinida para a estrada. Na execução desses algoritmos, há fatores externos, como condições climáticas, exposição a luzes artificiais ou sombras, que dificultam a detecção das características da pista. Por esse motivo, muitos pesquisadores aplicaram métodos de aprendizado de máquina, especialmente o aprendizado profundo utilizando Redes Neurais Convolucionais (CNN). Nesses algoritmos, em vez de programar regras explícitas ou presumir padrões de pista, esses modelos são treinados com grandes conjuntos de dados de amostras de imagens anotadas ou extraídas para aprender recursos e padrões que representam marcações de pista.

Para que esses algoritmos funcionem adequadamente, é preciso treinar a CNN com um conjunto de dados que contenha imagens com marcações de pista rotuladas. Durante o treinamento, a rede ajusta seus parâmetros internos (pesos) para otimizar sua capacidade de detectar faixas em vários cenários, como resultado desse treinamento, o algoritmo é capaz de classificar cada píxel da imagem à frente do veículo como parte de uma linha de faixa ou não (CHEN, Z.; LI, L.; HUANG, X., 2018). No entanto, esse último aspecto também representa uma grande desvantagem desses algoritmos, pois eles precisam usar uma grande quantidade de dados de treinamento rotulados, bem como possíveis vieses nos dados e a complexidade do projeto e do treinamento da rede.

2.4 Sensores para a função LKAS

Os veículos atuais incorporam uma variedade de sensores que permitem a aquisição de informações precisas sobre o estado do veículo e o ambiente ao seu redor. Esses sensores incluem o sistema de posicionamento global (GPS), a unidade de medição inercial (IMU), sensores ultrassônicos, LIDAR, radares e câmeras. A partir dos dados capturados por esses dispositivos, os sistemas de bordo podem gerar uma representação detalhada e interpretativa do ambiente, o que lhes permite tomar decisões antecipadas e agir de forma otimizada em vários sistemas de bordo, como o ângulo do volante ou a velocidade do veículo. O foco desse desenvolvimento será principalmente nos sensores projetados para obter uma compreensão abrangente do ambiente ao redor do veículo. Especificamente, os sensores que permitem a

implementação da função LKAS.

No campo das funções ADAS, as câmeras são sensores muito comuns devido ao baixo custo (HTET; KIONG; XINXIN, 2015). Existem diferentes funções desenvolvidas sob esse sensor, que podem ser de visão única (mono oculares), cuja taxa de atualização de quadros por segundo (fps) pode ser de 30 a 60 fps. No entanto, os algoritmos usados para identificação ou reconhecimento são de alta complexidade para este tipo de câmera.

As câmeras de visão estéreo, permitem reconhecer objetos e sua distância (XING et al., 2018). Por outro lado, a complexidade dos algoritmos poderia ser menor em comparação com aqueles usados para sensores de câmera única, devido ao uso de dois sensores ópticos ao mesmo tempo, a quantidade de fps pode ser reduzida para apenas 30 fps. Dependendo do tipo de aplicação, é possível usar quatro ou seis câmeras orientadas em diferentes ângulos para reconstruir imagens tridimensionais do ambiente.

No sistema LKAS é comum usar uma câmera localizada atrás do para-brisa dianteiro. Essa câmera é capaz de detectar e seguir as marcações da faixa, além de monitorar objetos próximos e placas de trânsito. Em situações perigosas (como saída de pista, possível colisão com um pedestre ou com um veículo próximo) a câmera desempenha um papel importante para impedir essa situação de risco.

No setor automotivo, esses sensores ganharam uma importância relevante. Os avanços nos algoritmos de processamento de imagens e os processadores mais potentes permitiram a detecção, a caracterização e o rastreamento de características da estrada e de objetos no ambiente (KANG; LEE, S.-H.; CHUNG, 2014).

Normalmente, também são utilizados dois tipos de radares, os de longo alcance e os de curto alcance, sendo que os radares de longo alcance podem cobrir distâncias relativamente grandes (cerca de 200 metros) em altas velocidades, enquanto os radares de curto alcance operam em velocidades mais baixas e são usados para perceber o entorno imediato do veículo (aproximadamente 30 metros). Sua alta precisão e capacidade de operar em diversas condições climáticas os tornam uma escolha importante para a maioria dos sistemas ADAS e um importante suporte no funcionamento da função LKAS (LIANG, J. et al., 2021).

2.5 Dinâmica veicular

O modelo matemático é de vital importância no desenvolvimento da função LKAS, pois permite emular o do comportamento aproximado do veículo em situações reais de condução. O modelo pode ser inserido em cenários de simulação com condições extremas, além de incluir a geometria da estrada, os sinais de trânsito, a presença de outros veículos e obstáculos. O modelo de dinâmica veicular permite também o desenvolvimento de funções de controle como ABS ou ESC, fatores importantes como a estabilidade do veículo, sistema

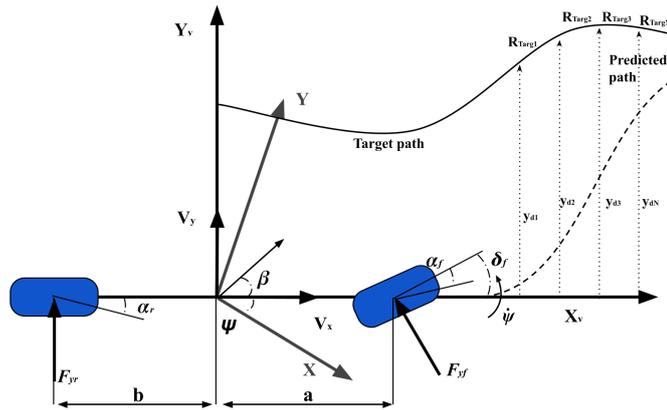


Figura 3 – Modelo cinemático do veículo (Modelo bicicleta)

de freio ou o controle de direção são normalmente validados em modelos que permitem avaliar o conforto e a segurança do veículo real.

O modelo de bicicleta tem sido utilizado em uma extensa quantidade de trabalhos de pesquisa para representar matematicamente a dinâmica veicular. Dependendo do trabalho de pesquisa, podem ser utilizados diferentes níveis de complexidade do modelo, com dois graus de liberdade para modelos mais simples ou uma grande quantidade de graus de liberdade, sempre buscando um equilíbrio entre o rendimento e a carga de computacional (RAJAMANI; RAJESH, 2012).

2.5.1 Modelo cinemático

No modelo cinemático, a descrição do movimento do veículo é considerada como um movimento plano, portanto, pode ser descrita por duas coordenadas X , Y , (em que x e y representam as coordenadas inerciais do centro de gravidade do veículo, indicando sua localização e ψ a orientação do veículo). Na figura 3 essas coordenadas são apresentadas, a velocidade do veículo v , o ângulo de deslizamento do veículo β , (em que a e b representam a distância da roda dianteira e traseira até o centro do veículo, respectivamente). Também são mostrados os ângulos de deslizamento dianteiro (α_f) e traseiro (α_r) das rodas, sendo a e b a distância dos eixos dianteiro e traseiro em relação ao COG, respectivamente. Em geral, as equações de movimento do veículo são representadas pelas equações 2.1:

$$\begin{aligned} \dot{X} &= v \cos(\psi + \beta) \\ \dot{Y} &= v \sin(\psi + \beta) \\ \dot{\psi} &= v \frac{\cos(\beta)}{a+b} (\tan(\alpha_f) - \tan(\alpha_r)) \end{aligned} \quad (2.1)$$

2.5.2 Dinâmica lateral

A dinâmica lateral do veículo refere-se ao comportamento do veículo enquanto se desloca lateralmente, ou seja, quando se move de lado a lado em relação à sua direção de movimento. Esta dinâmica é fundamental para entender como um veículo responde às forças laterais, como aquelas geradas durante uma curva ou mudança de faixa. A dinâmica lateral é representada por um modelo com dois graus de liberdade é representado pela posição lateral do veículo y e o ângulo de direção do veículo ψ , conforme apresentado na figura 3 para esse caso, a velocidade do veículo V_x é considerada constante, o movimento ao longo do eixo y pode ser representado pela segunda lei de Newton e a representação do momento de equilíbrio em relação ao eixo z , conforme mostrado nas equações 2.2:

$$\begin{aligned} m a_y &= -m v_x \dot{\psi} + F_{F_y} + F_{R_y} \\ I_z \ddot{\psi} &= l_F F_{F_y} - l_R F_{R_y} \end{aligned} \quad (2.2)$$

Em que m é a massa do veículo; $\dot{\psi}$ e $\ddot{\psi}$ são a taxa de guinada e a aceleração em torno do eixo z no centro de gravidade (COG) do veículo; a_y é a aceleração lateral e v_x a velocidade longitudinal do veículo; I_z é o momento de inércia em torno do eixo z . As forças laterais são uma simplificação e linearização das forças não lineares modeladas pela fórmula mágica de Pacejka. Essa linearização é crucial para obter um modelo mais simples a ser utilizado pelo sistema de controle. As forças laterais no pneu F_{F_y} e F_{R_y} podem ser expressas pelas seguintes equações 2.3:

$$\begin{aligned} F_{F_y} &= 2 C_{\alpha_f} \alpha_f \\ F_{R_y} &= 2 C_{\alpha_r} \alpha_r \end{aligned} \quad (2.3)$$

Em que C_{α_f} e C_{α_r} representam a rigidez dianteira e traseira em curvas, respectivamente. Como o ângulo de deslizamento (α) é uma métrica importante para calcular a trajetória real da roda e a direção do veículo, os ângulos de deslizamento dianteiro (α_f) e traseiro (α_r) podem ser definidos pela equação 2.4:

$$\alpha_f = \delta - \frac{v_y + a \dot{\psi}}{v_x} \quad ; \quad \alpha_r = -\frac{v_y - b \dot{\psi}}{v_x} \quad (2.4)$$

Para obter uma representação em espaço de estado do modelo, as equações 2.2, 2.3, 2.4 são substituídas para se chegar à equação 2.5:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A} \mathbf{x}(t) + \mathbf{B} u(t) \\ \mathbf{y}(t) &= \mathbf{C} \mathbf{x}(t) \end{aligned} \quad (2.5)$$

Em que **A**, **B** e **C** são as matrizes de estado, entrada e saída, respectivamente. O vetor de estado é representado por $\mathbf{x}(t)$, enquanto $u(t)$ (ângulo do volante) e $y(t)$ a saída do sistema.

2.5.2.1 Modelo 1 de rastreamento de trajetória

A representação linear invariante no tempo (LTI) do modelo 1 de bicicleta com rastreamento de trajetória é apresentada em 2.6 como e mostrado a seguir:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mv_x} & 0 & -v_x - \frac{2aC_{\alpha f} - 2bC_{\alpha r}}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2aC_{\alpha f} - 2bC_{\alpha r}}{I_z v_x} & 0 & -\frac{2a^2 C_{\alpha f} + 2b^2 C_{\alpha r}}{I_z v_x} \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2aC_{\alpha f}}{I_z} \end{bmatrix} ; \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \quad (2.6)$$

Na representação matemática no vetor de estados é $\mathbf{x}(t) = [y \ \dot{y} \ \psi \ \dot{\psi}]^T$.

2.5.2.2 Modelo 2 de rastreamento de trajetória

Neste trabalho, foi considerada outra formulação matemática em que a velocidade lateral é substituída no vetor de estados pelo ângulo de deslizamento lateral, denotado como β , que é aproximadamente igual a $\beta \approx \frac{v_y}{v_x}$ devido à suposição de velocidade longitudinal constante e pequenas alterações no ângulo de direção, conforme discutido em (ULSOY; PENG, 2012). Como resultado, a representação vetorial atualizada do espaço de estado passa a ser $\mathbf{x}(t) = [y \ \psi \ \beta \ \dot{\psi}]^T$, em que y representa a posição lateral, ψ denota o ângulo de guinada, β representa o ângulo de deriva e $\dot{\psi}$ é a taxa de guinada. O sinal de controle, denotado como $u(t)$, corresponde ao ângulo de direção. As matrizes necessárias para a representação do modelo modificado são apresentadas na equação 2.7:

$$\begin{aligned}
\mathbf{A} &= \begin{bmatrix} 0 & v_x & v_x & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{C_{\alpha f} + C_{\alpha r}}{I_z} & \frac{aC_{\alpha f} - bC_{\alpha r}}{I_z v_x} - 1 \\ 0 & 0 & \frac{bC_{\alpha r} - aC_{\alpha f}}{I_z} & -\frac{mv_x^2}{(b^2 C_{\alpha r} + a^2 C_{\alpha f}) I_z v_x} \end{bmatrix} \\
\mathbf{B} &= \begin{bmatrix} 0 \\ 0 \\ \frac{C_{\alpha f}}{I_z} \\ \frac{aC_{\alpha f}^m}{I_z} \end{bmatrix} ; \quad \mathbf{C} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}^T
\end{aligned} \tag{2.7}$$

2.5.2.3 Modelo 3 em função do erro

O terceiro modelo apresentado neste trabalho se concentra no uso de uma câmera para a aquisição dos estados atuais do veículo na pista. Por esse motivo, é importante que as variáveis de estado sejam uma função do erro em relação à posição e_1 e à orientação e_2 atual na pista. Com isso, o modelo apresentado por (RAJAMANI; RAJESH, 2012) pode ser utilizado, em que as variáveis do modelo 1 de rastreamento de trajetória 2.6 podem ser redefinidas como uma função do erro. Considerando V_x constante e um raio constante (R) da estrada, é possível definir a taxa de mudança da orientação desejada do veículo como em 2.8:

$$\dot{\psi}_{des} = \frac{V_x}{R} \tag{2.8}$$

e a aceleração desejada de guinada do veículo pode ser escrita como:

$$\frac{V_x^2}{R} = V_x \dot{\psi}_{des} \tag{2.9}$$

definindo \dot{e}_1 e e_2 conforme descrito abaixo:

$$\dot{e}_1 = (\ddot{y} + V_x \dot{\psi}) - \frac{V_x^2}{R} = \ddot{y} + V_x (\dot{\psi} - \dot{\psi}_{des}) \tag{2.10}$$

$$e_2 = \psi - \psi_{des} \tag{2.11}$$

Com isso, é possível definir o erro de taxa de mudança de deslocamento lateral como:

$$\dot{e}_1 = \ddot{y} + V_x (\dot{\psi} - \dot{\psi}_{des}) \tag{2.12}$$

A nova representação do espaço de estado do modelo é dada por um sistema $\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}_1 u(t) + \mathbf{B}_2 \dot{\psi}_{des}(t)$, em que o vetor de espaço de estado é $\mathbf{x}(t) = [e_1 \dot{e}_1 e_2 \dot{e}_2]^T$ e as matrizes do sistema são mostradas na equação 2.13:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{\alpha f} + 2C_{\alpha r}}{mv_x} & \frac{2C_{\alpha f} + 2C_{\alpha r}}{m} & -\frac{2aC_{\alpha f} + 2bC_{\alpha r}}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & -\frac{2aC_{\alpha f} - 2bC_{\alpha r}}{I_z v_x} & \frac{2aC_{\alpha f} - 2bC_{\alpha r}}{I_z} & -\frac{2a^2C_{\alpha f} + 2b^2C_{\alpha r}}{I_z v_x} \end{bmatrix}$$

$$\mathbf{B}_1 = \begin{bmatrix} 0 \\ \frac{2C_{\alpha f}}{m} \\ 0 \\ \frac{2aC_{\alpha f}}{I_z} \end{bmatrix}; \mathbf{B}_2 = \begin{bmatrix} 0 \\ -\frac{2aC_{\alpha f} + 2bC_{\alpha r}}{mv_x} - v_x \\ 0 \\ -\frac{2a^2C_{\alpha f} + 2b^2C_{\alpha r}}{I_z v_x} \end{bmatrix} \quad (2.13)$$

e_1 representa o erro de posição lateral do veículo e e_2 é o erro do ângulo de guinada do veículo em relação à pista. Nesse novo modelo, tem-se uma segunda entrada para o sistema, que é a taxa de guinada desejada $\dot{\psi}_{des}$. Esses modelos são usados na representação da dinâmica lateral do veículo. Nessas representações, busca-se um equilíbrio entre a exatidão e o comportamento em tempo real do modelo.

2.6 Estratégias de controle

No desenvolvimento da função LKAS, existem parâmetros como segurança, suavidade e restrições físicas que devem ser considerados como objetivos de controle. O controlador MPC teve uma recepção positiva nas últimas décadas, utilizado em muitos trabalhos de pesquisa nas funções ADAS, para melhorar o rendimento e seguridade em funções como LKAS. Esse controlador permite a aplicação de restrições físicas e a segurança do processo em sua estratégia de controle.

O controle preditivo baseado em modelo (MPC) tem sido fundamental na melhoria de processos industriais desde sua concepção na década de 1970. Surgido no contexto da indústria química e de processos, o MPC evoluiu significativamente ao longo do tempo, graças aos avanços na teoria de controle, computação e otimização. Sua capacidade de lidar com sistemas dinâmicos multivariáveis, bem como de incorporar restrições e mudanças nas condições operacionais, o tornou uma ferramenta indispensável em uma variedade de indústrias, incluindo automotiva, petroquímica e farmacêutica (FARAG, W. A., 2020). Através de sua aplicação, o MPC tem consistentemente demonstrado sua capacidade de melhorar o desempenho, estabilidade e eficiência dos processos industriais, contribuindo assim para o avanço contínuo da tecnologia e produção (KAMAT, 2020).

O MPC consiste em uma estratégia de controle que usa um modelo matemático do sistema para prever seu comportamento futuro e otimizar as entradas de controle com base

nas informações do estado atual e passado (KIM, D. J. et al., 2018). Ele tem sido amplamente utilizado em diversos campos, incluindo a eletrônica de potência, controle de motores industriais e, nos últimos anos, tem sido amplamente estudado na aplicação de funções de segurança e de suporte ao condutor. Em cada instante, o controlador recebe o estado atual da planta e calcula uma sequência de ações de controle que minimiza uma função de custo ao longo do horizonte de previsão para encontrar o ângulo para se manter no centro da faixa, resolvendo um problema de otimização sujeito a restrições (LI, R. et al., 2020). Essas restrições são necessárias porque, em um veículo real, o melhor ângulo das rodas tem um limite físico, também restrições de segurança ou de conforto podem ser impostas. O cálculo realizado a cada tempo de amostragem é baseado em um modelo interno da planta e depende do estado atual do sistema. Na figura 4 é apresentado o funcionamento geral do controle MPC.

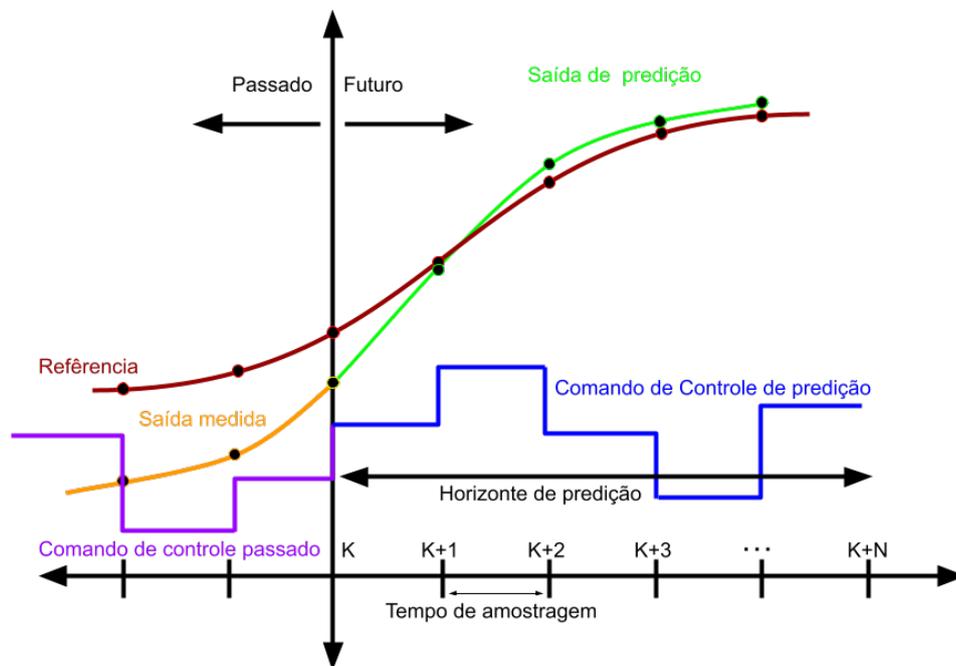


Figura 4 – Esquema de funcionamento do controle preditivo baseado em modelo.

O controlador, então, aplica apenas a primeira ação de controle calculada, sem considerar as outras ações futuras. Outra vantagem presente no MPC é que ele pode ser aplicado em sistemas de múltiplas entradas e saídas. Ele também pode lidar com restrições tanto nas entradas quanto nas saídas. O MPC tem a capacidade de prever eventos futuros (FARAG, W. A., 2020).

2.7 Formulação do controle MPC

O MPC é uma abordagem que utiliza um modelo para otimizar o controle do sistema ao longo de um horizonte de previsão. A formulação matemática incorpora restrições di-

nâmicas e estáticas, garantindo sua estabilidade e seu desempenho. Ao aplicar o MPC no LKAS, é possível representar a dinâmica do veículo, considerar limites de segurança, físicos (ângulo das rodas) ou de conforto (taxa de guinada), e otimizar a trajetória para manter o veículo dentro das faixas.

Na atualidade, a utilização do controlador MPC pode ser encontrado em diversas áreas da ciência e indústria. Para cada caso específico, é possível encontrar uma formulação matemática que abrange as restrições e objetivos do problema aplicado. A seguir, serão apresentadas duas formulações aplicadas à função LKAS.

2.7.1 Formulação do MPC para modelos de rastreamento de trajetória

Como já mencionado na abordagem dos dois primeiros modelos, foram usados os critérios de formulação encontrados em (ALAMIR, 2013). Nesse cenário, a discretização do modelo do veículo pode ser estabelecida pela equação 2.14:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}_d \mathbf{x}(k) + \mathbf{B}_d u(k) \\ \mathbf{y}(k) &= \mathbf{C}_d \mathbf{x}(k) \end{aligned} \quad (2.14)$$

Em que k é o instante atual, $\mathbf{A}_d \in \mathbb{R}^{n,n}$ e $\mathbf{B}_d \in \mathbb{R}^{n,n_u}$ são matrizes de estado espacial discretizadas da dinâmica lateral (Equação 2.14). Um mapa de predição, usado para estimar os estados futuros do veículo, é obtido calculando-se as N sequências de ações futuras. Assim, as entradas de controle antecipadas em N passos $\tilde{u}(k)$ ser obtidas com base na equação 2.15:

$$\tilde{u}(k) = (u(k) \ u(k+1) \ \dots \ u(k+N-1))^T \in \mathbb{R}^{N \cdot n_u} \quad (2.15)$$

Um mapa de predição para N passos adiante permite estimar a trajetória de estados futuros $\tilde{x}(k|\tilde{u}(k))$ com base em uma sequência de ações futuras $\tilde{u}(k)$. Assim, a trajetória do estado futuro a partir de $x(k)$ pode ser escrita da seguinte forma:

$$\tilde{x}(k|\tilde{u}(k)) = (x(k+1) \ \dots \ x(k+N))^T \in \mathbb{R}^{N \cdot n} \quad (2.16)$$

Dadas as matrizes \mathbf{A}_d e \mathbf{B}_d , uma expressão geral para $\tilde{x}(k|\tilde{u}(k))$ pode ser escrita como:

$$x(k+i) = \Phi_i x(k) + \Psi_i \tilde{u}(k) \quad (2.17)$$

Em que as matrizes Φ_i e Ψ_i são dadas por:

$$\Phi_i = \mathbf{A}_d^i \quad (2.18)$$

$$\Psi_i = \left(\mathbf{A}_d^{i-1} \mathbf{B} \quad \dots \quad \mathbf{A}_d \mathbf{B} \quad \mathbf{B} \right)^T \begin{pmatrix} \mathbf{\Pi}_1^{(n_u, N)} \\ \mathbf{\Pi}_2^{(n_u, N)} \\ \vdots \\ \mathbf{\Pi}_i^{(n_u, N)} \end{pmatrix} \quad (2.19)$$

Em que $\mathbf{\Pi}_i^{(n_u, N)}$ é a matriz utilizada para selecionar a i^{th} parte da dimensão n_u de um vetor que contém a concatenação de N vetores e pode ser definida como sendo:

$$\mathbf{\Pi}_i^{(n, N)} = \begin{pmatrix} \mathbf{0}_{n \times n(i-1)} & \mathbf{I}_{n \times n} & \mathbf{0}_{n \times n(N-i)} \end{pmatrix} \quad (2.20)$$

O controlador MPC para a função LKAS pretende regular o deslocamento lateral do veículo (y) seguindo uma trajetória desejada. Da mesma forma, as primeiras variáveis de estado são a saída a ser rastreada, independentemente da dinâmica veicular. Assim, a equação de saída do sistema pode ser formalizada da seguinte forma:

$$\mathbf{y}(k) = \mathbf{C}_r \mathbf{x}(k) \quad (2.21)$$

$$\mathbf{C}_r = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \quad (2.22)$$

Em que \mathbf{C}_r é a matriz de saída regulada.

2.7.1.1 Função de custo

Na teoria de controle do MPC, uma função de custo é uma expressão matemática utilizada para balancear a exatidão do rastreamento e o esforço de controle. Na maioria das vezes, uma função de custo é usada para calcular o desempenho do MPC em um horizonte de predição de n passos adiante (JÚNIOR; MURILO; LOPES, 2022). A função de custo utilizada para projetar o controlador MPC para o LKAS pode ser dada pela equação 2.23:

$$J(k) = \sum_{i=1}^N \|y_r(k+i) - y_d(k+i)\|_{Q_y}^2 + \sum_{i=1}^N \|\mathbf{\Pi}_i^{(n_u, N)} \tilde{u}(k) - u_d\|_{Q_u}^2 \quad (2.23)$$

Em que Q_u e Q_y são as matrizes utilizadas para penalizar o comando de controle (u) e o erro de trajetória ($y_r - y_d$), respectivamente; u_d é o comando de controle desejado, enquanto y_d é a saída desejada da variável regulada $y(k)$. Além disso, uma função de custo executada a cada instante de tempo (k), em um horizonte de predição ($[k, k+N]$), pode ser descrita como na equação 2.24:

$$J(k) = \frac{1}{2} \tilde{\mathbf{u}}^T(k) \mathbf{H} \tilde{\mathbf{u}}(k) + \mathbf{F}^T(k) \tilde{\mathbf{u}}(k) + Cte \quad (2.24)$$

Em que:

$$\begin{aligned} \mathbf{H} &:= 2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y C_r \Psi_i + (\Pi_i^{(n_u, N)})^T Q_u (\Pi_i^{(n_u, N)})] \\ \mathbf{F}(\mathbf{k}) &:= F_1 x(k) + F_2 y_d + F_3 u_d \\ \mathbf{F}_1 &:= 2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y C_r \Phi_i] \\ \mathbf{F}_2 &:= -2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y \Pi_i^{(n_r, N)}] \\ \mathbf{F}_3 &:= 2 \sum_{i=1}^N [(\Pi_i^{(n_u, N)})^T Q_u] \end{aligned} \quad (2.25)$$

\mathbf{F}_1 , \mathbf{F}_2 e \mathbf{F}_3 são matrizes off-line da modelagem do sistema e dos horizontes de predição selecionados. \mathbf{F} pode mudar conforme o vetor de estado $\mathbf{x}(\mathbf{k})$ e a referência de saída desejada (y_d) e deve ser atualizado em cada instante de tempo (k). Além disso, $\mathbf{H} \in \mathbb{R}^{N n_u \times N n_u}$ é a Hessiana definido positivamente da função quadrática que garante um problema de otimização bem proposto (ALAMIR, 2013). O valor mínimo encontrado na sequência $\tilde{\mathbf{u}}(k)$ não é afetado pelo termo *Cte*.

2.7.1.2 Restrições impostas ao controlador

Um dos avanços mais significativos das soluções baseadas em MPC está no estabelecimento de limites de controle. Os limites de controle são restrições impostas por sistemas físicos e funções que devem ser consideradas pela lei de controle. Para aplicações em manutenção de faixa, a restrição principal é a necessidade de manter o veículo dentro da faixa de rodagem. Qualquer solução baseada em MPC para o LKAS deve incluir o deslocamento lateral máximo permitido enquanto segue a trajetória desejada do veículo. O deslocamento lateral pode ser limitado conforme mostrado na equação 2.26 :

$$\begin{aligned} \mathbf{y}_c^{\min} &< \mathbf{y}_c < \mathbf{y}_c^{\max} \\ \mathbf{y}_c &= \mathbf{C}_c \mathbf{x} \quad ; \quad \mathbf{C}_c = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix} \end{aligned} \quad (2.26)$$

Em que \mathbf{y}_c^{\min} e \mathbf{y}_c^{\max} representam os limites de deslocamento lateral, \mathbf{y}_c e \mathbf{C}_c são o vetor de saída e as restrições de matriz aplicadas ao deslocamento lateral do veículo, respectivamente. No LKAS, a trajetória de destino (isto é, a trajetória a ser seguida) é determinada com o uso de sistema de visão computacional. Por meio de imagens de estrada pré-processadas

e filtradas, uma trajetória-alvo (ou ponto de ajuste) é gerado em tempo real e devendo ser seguida pelo controlador MPC. Portanto, a equação 2.27 mostra os limites de deslocamento lateral, incluindo a trajetória de destino, podem ser expressos como:

$$\begin{aligned} y_c^{min} &= -(D_r/2) + (D_v/2) \\ y_c^{max} &= (D_r/2) - (D_v/2) \end{aligned} \quad (2.27)$$

D_r e D_v são a largura da faixa de rodagem e do veículo, respectivamente. Na figura 5 são apresentados estes limites. Outra restrição importante está relacionada ao comando de direção do condutor $u(k)$. O comando da direção, que é o comando de controle para o controlador, deve estar dentro de uma faixa de ângulos de giro máximos pré-definidos. Assim, o comando de direção do condutor deve ser restringido pelos seguintes limites:

$$\mathbf{u}^{min} < \mathbf{u} < \mathbf{u}^{max} \quad (2.28)$$

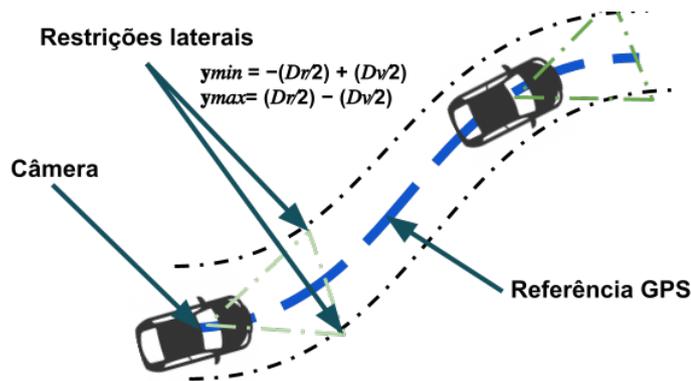


Figura 5 – Modelos de rastreamento de trajetória

Em que \mathbf{u}^{min} e \mathbf{u}^{max} são as entradas mínimas e máximas permitidas para o sistema de direção. A taxa de variação das variáveis de controle desempenha um papel importante na dinâmica veicular. É importante mencionar que modificação abrupta das variáveis de controle (como o ângulo de direção) deve ser evitada por motivos de conforto e segurança. A restrição aplicada à taxa de variação das variáveis de controle (Δu) pode ser formalizada pela seguinte equação 2.29:

$$\Delta \mathbf{u}^{min} < \mathbf{u}(\mathbf{k} + \mathbf{i}) - \mathbf{u}(\mathbf{k} + \mathbf{i} - 1) < \Delta \mathbf{u}^{max} \quad (2.29)$$

Em que $\Delta \mathbf{u}^{min}$ e $\Delta \mathbf{u}^{max}$ são as restrições da taxa de variação de entrada. Reunindo as equações 2.26-2.29, a restrição de trajetória do veículo com o sistema LKAS pode ser escrita em uma forma de matriz compacta (equação 2.30):

$$\begin{pmatrix} y_c^{min} \\ \mathbf{u}^{min} \\ \Delta \mathbf{u}^{min} \end{pmatrix} < \begin{pmatrix} y_c(\mathbf{k} + \mathbf{i}) \\ \mathbf{u}(\mathbf{k} + \mathbf{i} - 1) \\ \mathbf{u}(\mathbf{k} + \mathbf{i}) - \mathbf{u}(\mathbf{k} + \mathbf{i} - 1) \end{pmatrix} < \begin{pmatrix} y_c^{max} \\ \mathbf{u}^{max} \\ \Delta \mathbf{u}^{max} \end{pmatrix} \quad (2.30)$$

Em que $i \in \{1, \dots, N\}$ são os valores intermediários dentro de todo o horizonte de predição (N). Uma vez que a matriz de restrição da trajetória do veículo tenha sido determinada, uma função de custo pode ser escrita combinando as equações 2.23 e 2.30 como:

$$\begin{aligned} \tilde{\mathbf{u}}^{\text{opt}}(k) : &= \arg \min_{\tilde{\mathbf{u}}} \left[\frac{1}{2} \tilde{\mathbf{u}}^T \mathbf{H} \tilde{\mathbf{u}} + \mathbf{F}^T(k) \tilde{\mathbf{u}} \right] \\ &\text{subject to:} \\ &\mathbf{A}_{\text{ineq}} \tilde{\mathbf{u}} \leq \mathbf{B}_{\text{ineq}}(k) \\ &\tilde{\mathbf{u}}^{\min} \leq \tilde{\mathbf{u}} \leq \tilde{\mathbf{u}}^{\max} \end{aligned} \quad (2.31)$$

Em que \mathbf{A}_{ineq} e $\mathbf{B}_{\text{ineq}}(k)$ são os estados do veículo e as matrizes de restrição de taxa de alteração incluídas na função de custo. Uma descrição completa da formulação de \mathbf{A}_{ineq} e $\mathbf{B}_{\text{ineq}}(k)$ pode ser encontrada em (ALAMIR, 2013).

2.7.2 Formulação do MPC para modelo com perturbação

No desenvolvimento da função LKAS é importante destacar que podem ser consideradas diversas opções de modelos matemáticos para representar a dinâmica lateral do veículo com maior exatidão. A escolha do modelo adequado dependerá de vários fatores, como a complexidade do sistema, a disponibilidade de dados de entrada e a capacidade computacional do sistema de controle. A formulação apresentada (WANG, L., 2009) na equação (2.32) é utilizada com o modelo do erro 2.13, a taxa de guinada desejada é uma perturbação do sistema determinada pela velocidade e pela curvatura da estrada, conforme visto na equação 2.8. Para a nova formulação após a discretização do modelo é apresentada na equação 2.32:

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_{1d} \delta(k) + \mathbf{B}_{2d} \dot{\psi}_{des}(k) \quad (2.32)$$

Para esse modelo, o controlador MPC deve penalizar a taxa de alteração do ângulo das rodas, a fim de promover uma condução mais confortável. Para isso, é proposta a seguinte variável que representa a alteração do ângulo de direção:

$$\Delta \delta(k) = \delta(k) - \delta(k-1) \quad (2.33)$$

Isso é necessário devido à formulação da função de custo que penaliza o custo do controle e, portanto, o sinal de controle é transformado de forma incremental, conforme mostrado na equação 2.33. Após realizar essa transformação, a nova forma do sinal de controle é apresentada como mostrado na equação 2.34:

$$x(k+1) = \mathbf{A}_d x(k) + \mathbf{B}_{1d} \delta(k-1) + \mathbf{B}_{1d} \Delta \delta(k) + \mathbf{B}_{2d} \dot{\psi}_{des}(k) \quad (2.34)$$

Com essa nova formulação, a representação do espaço de estado do sistema ampliado pode ser estabelecida da seguinte forma (equação 2.36):

$$\begin{bmatrix} x(k+1) \\ \delta(k) \end{bmatrix} = \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_{1d} \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ \delta(k-1) \end{bmatrix} + \begin{bmatrix} \mathbf{B}_{1d} \\ I \end{bmatrix} \Delta\delta(k) + \begin{bmatrix} \mathbf{B}_{2d} \\ 0 \end{bmatrix} \dot{\psi}_{des}(k) \quad (2.35)$$

$$y(k) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(k) \\ \delta(k-1) \end{bmatrix} \quad (2.36)$$

e as matrizes aumentadas são dadas por:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_d & \mathbf{B}_{1d} \\ 0 & I \end{bmatrix}, \mathbf{B}_1 = \begin{bmatrix} \mathbf{B}_{1d} \\ I \end{bmatrix}, \mathbf{B}_2 = \begin{bmatrix} \mathbf{B}_{2d} \\ 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} C & 0 \end{bmatrix} \quad (2.37)$$

2.7.2.1 Função de custo

Para o modelo de erro, a função de custo otimiza a resposta para $\Delta\delta$ tendo a seguinte forma:

$$J(k) = (\mathbf{x}(k))^T \mathbf{Q}_y \mathbf{x}(k) + \Delta\delta^T \mathbf{Q}_{\Delta\delta} \Delta\delta \quad (2.38)$$

Em que \mathbf{Q}_y e \mathbf{Q}_u são as matrizes de penalização de estado e de entrada, respectivamente. As previsões das variáveis do sistema no controlador MPC são feitas dentro de um intervalo de predição (horizonte de predição) n_p considerando uma sequência de entradas de controle (horizonte de controle) n_c , a evolução dessas variáveis pode ser representada conforme mostrado na equação 2.39:

$$\begin{aligned} \Delta\bar{\delta} &= \begin{bmatrix} \Delta\delta(k) & \Delta\delta(k+1) & \dots & \Delta\delta(k+n_c-1) \end{bmatrix}^T \\ \bar{\mathbf{x}}(k) &= \begin{bmatrix} \mathbf{x}(k+1|k)^T & \mathbf{x}(k+2|k)^T & \dots & \mathbf{x}(k+n_p|k)^T \end{bmatrix}^T \\ \bar{\kappa} &= \begin{bmatrix} \kappa(k) & \kappa(k+1) & \dots & \kappa(k+n_p-1) \end{bmatrix}^T \end{aligned} \quad (2.39)$$

Conforme as variáveis anteriormente definidas, a equação do espaço de estado é estendida conforme mostrado em 2.40:

$$\begin{aligned}
\mathbf{x}(k+1 | k) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}_1\Delta\delta(k) + \mathbf{B}_2\dot{\psi}_{des}(k) \\
\mathbf{x}(k+2 | k) &= \mathbf{A}\mathbf{x}(k+1 | k) + \mathbf{B}_1\Delta\delta(k+1) + \mathbf{B}_2\dot{\psi}_{des}(k+1) \\
&= \mathbf{A}^2\mathbf{x}(k) + \mathbf{A}\mathbf{B}_1\Delta\delta(k) + \mathbf{A}\mathbf{B}_2\dot{\psi}_{des}(k) \\
&\quad + \mathbf{B}_1\Delta\delta(k+1) + \mathbf{B}_2\dot{\psi}_{des}(k+1) \\
&\quad \vdots \\
\mathbf{x}(k+n_p | k) &= \mathbf{A}^{n_p}\mathbf{x}(k) + \mathbf{A}^{n_p-1}\mathbf{B}_1\Delta\delta(k) + \\
&\quad + \mathbf{A}^{n_p-1}\mathbf{B}_2\dot{\psi}_{des}(k) + \dots \\
&\quad + \mathbf{A}^{n_p}\mathbf{B}_2\dot{\psi}_{des}(k+n_p-1)
\end{aligned} \tag{2.40}$$

Já a equação 2.17, a equação dos espaços de estado pode ser reescrita como sendo:

$$\bar{\mathbf{x}}(k) = F\mathbf{x}(k) + \Phi_1\Delta\bar{\delta} + \Phi_2\dot{\psi}_{des} \tag{2.41}$$

Em que F , Φ_1 y Φ_2 são definidos como:

$$F = \begin{bmatrix} \mathbf{A} & \mathbf{A}^2 & \dots & \mathbf{A}^{n_p} \end{bmatrix}^T \tag{2.42}$$

$$\Phi_1 = \begin{bmatrix} \mathbf{B}_1 & 0 & 0 & \dots & 0 \\ \mathbf{A}\mathbf{B}_1 & \mathbf{B}_1 & 0 & \dots & 0 \\ \mathbf{A}^2\mathbf{B}_1 & \mathbf{A}\mathbf{B}_1 & \mathbf{B}_1 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \vdots \\ \mathbf{A}^{n_p-1}\mathbf{B}_1 & \mathbf{A}^{n_p-2}\mathbf{B}_1 & \mathbf{A}^{n_p-3}\mathbf{B}_1 & \dots & \mathbf{A}^{n_p-n_c}\mathbf{B}_1 \end{bmatrix} \tag{2.43}$$

$$\Phi_2 = \begin{bmatrix} \mathbf{B}_2 & 0 & 0 & \dots & 0 \\ \mathbf{A}\mathbf{B}_2 & \mathbf{B}_2 & 0 & \dots & 0 \\ \mathbf{A}^2\mathbf{B}_2 & \mathbf{A}\mathbf{B}_2 & \mathbf{B}_2 & \dots & 0 \\ \dots & \dots & \dots & \ddots & \vdots \\ \mathbf{A}^{n_p-1}\mathbf{B}_2 & \mathbf{A}^{n_p-2}\mathbf{B}_2 & \mathbf{A}^{n_p-3}\mathbf{B}_2 & \dots & \mathbf{A}^{n_p-n_c}\mathbf{B}_2 \end{bmatrix} \tag{2.44}$$

Na formulação anterior $N = n_p = n_c$, o horizonte de controle n_c deve ser menor ou igual ao horizonte de predição.

2.7.3 Definição das Restrições

As restrições impostas para o modelo rastreamento da trajetória, assim como as restrições para o modelo de erro, envolvem o deslocamento lateral, o ângulo de direção máximo permitido e a taxa de variação do ângulo de direção do veículo 2.30; para essa formulação, as restrições são apresentadas a seguir:

A restrição para a taxa de mudança do volante:

$$\begin{bmatrix} -I \\ I \end{bmatrix} \Delta\delta \leq \begin{bmatrix} -\Delta\delta_{min} \\ \Delta\delta_{max} \end{bmatrix}$$

Restrição do ângulo de direção:

$$\begin{bmatrix} -\mathbf{H}_2 \\ \mathbf{H}_2 \end{bmatrix} \Delta\delta \leq \begin{bmatrix} -\delta_{min} + \mathbf{H}_1\delta(k-1) \\ \delta_{max} - \mathbf{H}_1\delta(k-1) \end{bmatrix}$$

E o deslocamento máximo permitido:

$$\begin{bmatrix} -M_1 \\ M_1 \end{bmatrix} \Delta\delta \leq \begin{bmatrix} -Y_{e_{min}} + \mathbf{F}\mathbf{x}(k) + M_2\dot{\psi}_{des} \\ Y_{e_{max}} - \mathbf{F}\mathbf{x}(k) - M_2\dot{\psi}_{des} \end{bmatrix}$$

Em que $\mathbf{H}_1 \in \mathbb{R}^{n_c}$ é um vetor de coluna e $\mathbf{H}_2 \in \mathbb{R}^{n_c \times n_c}$ é uma matriz triangular menor; \mathbf{F} , \mathbf{M}_1 e \mathbf{M}_2 , representam a função que restringe o estado permitido de deslocamento lateral, além disso, δ_{min} , δ_{max} , $\Delta\delta_{min}$, $\Delta\delta_{max}$, Y_{min} e Y_{max} são os valores fixos que representam os limites definidos para cada uma das variáveis restritas. As restrições são apresentadas como na equação 2.45 para a representação da programação quadrática.

$$\begin{aligned} \mathbf{A}_{ineq}\Delta\delta &\leq \mathbf{B}_{ineq}(k) \\ \tilde{\mathbf{u}}^{min} &\leq \Delta\delta \leq \tilde{\mathbf{u}}^{max} \end{aligned} \quad (2.45)$$

2.7.4 Parametrização do controlador MPC

Um dos desafios mais significativos nas aplicações de controle MPC diz respeito ao tempo de computacional necessário para obter a sequência de comandos de saída. Como o MPC calcula uma saída de controle otimizada, em cada instante de tempo, para um sistema dinâmico restrito, ele impõe uma sobrecarga computacional significativa (FARONI et al., 2017)(CHEN, Y. et al., 2020). Neste contexto, os métodos de parametrização surgem como uma possível alternativa para tornar o MPC viável em aplicações em tempo real (ALAMIR, 2013). Esse método reduz o número de graus de liberdade ($N \cdot n_u$) do processo de otimização ao manter um perfil de controle constante em períodos intermediários do horizonte de predição. A Figura 6 destaca os dois métodos (trivial e exponencial) comumente utilizados para a parametrização do MPC.

Na parametrização trivial (PT), as variáveis de controle são interpoladas e mantidas constantes em algumas amostras específicas visando diminuir o tamanho das variáveis de decisão n_p . Além disso, os valores intermediários são, por sua vez, obtidos por meio de interpolação linear. O espaço de busca do processo de otimização é reduzido de $N \times n_u$ para $n_r \times n_u$ (em que n_r é o número de intervalos referentes à parametrização trivial). Com a parametrização trivial, a dimensão da variável de decisão pode ser escrita da seguinte forma:

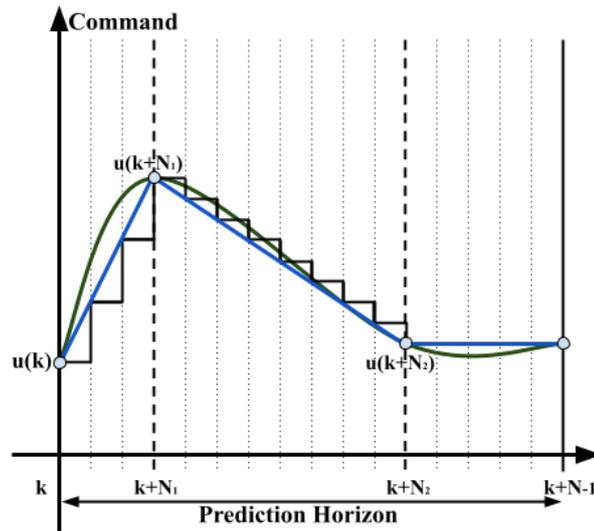


Figura 6 – Parametrização trivial (linha azul) e exponencial (linha verde) em todo o horizonte de predição para intervalos de três vezes ($n_r = 3$). Adaptado de (ALAMIR, 2013).

$$p = (u(k) \quad u(k + N_1) \cdots u(k + N_{n_r-1}))^T \quad (2.46)$$

A parametrização exponencial (EP) é outra estratégia usada para melhorar a eficiência computacional (JÚNIOR; MURILO; LOPES, 2022) reduzindo a complexidade do problema de otimização (ALAMIR, 2013). Diferentemente da PT, essa técnica representa uma sequência de controle (u) usando um termo exponencial (RODRIGUES et al., 2019). A principal vantagem dessa abordagem consiste em sua capacidade de produzir uma função matemática por meio da combinação de exponenciais, permitindo o ajuste direto dos coeficientes para configurar a suavidade e tempo de resposta.

Considerando n_u atuadores e um vetor de tempo de estabilização ($\tau_r \in \mathbb{R}^{n_u}$) define o tempo necessário para que cada atuador atinja a referência dentro de uma determinada faixa de tolerância. Essa restrição evita uma modificação repentina nos comandos de controle ou até mesmo excede os limites de velocidade do atuador. Assim, o perfil de controle exponencial pode ser definido da seguinte forma (equação 2.47):

$$u_j(k + i) = \sum_{l=1}^{n_e^{(j)}} \underbrace{\left[e^{\frac{-\lambda_j(i\tau)}{(l-1)\alpha+1}} \right]}_{m_{j,i}(l)} \cdot p_l^{(j)}; \quad \alpha > 1 \quad (2.47)$$

Na equação mencionada acima, $\alpha > 1$ e λ são os parâmetros de ajuste que podem ser definidos com base no tempo de estabilização dos atuadores. Para evitar oscilações indesejadas, é possível definir λ como $3/\tau_r$, em que τ_r representa o tempo de estabilização desejável. Esta escolha garante que os sinais de controle apresentem um tempo de resposta

de 95% em relação aos limites dos atuadores. Assim, o perfil de controle resultante é definido com o vetor p dado por:

$$p = \begin{pmatrix} p^{(1)} \in \mathbb{R}^{n_e^{(1)}} \\ \vdots \\ p^{(n_u)} \in \mathbb{R}^{n_e^{(n_u)}} \end{pmatrix} \in \mathbb{R}^{n_p} \quad (2.48)$$

Em que n_e é o número de exponenciais da sequência de controle com a dimensão do vetor p é definido como:

$$n_p = \sum_{j=1}^{n_u} n_e^{(j)} \quad (2.49)$$

O novo conjunto de n_p é determinado apenas pelo número de atuadores n_u e exponenciais $n_e^{(j)}$ selecionados, independentemente do horizonte de predição utilizado. Consequentemente, a sequência de controle pode ser expressa de forma compacta da seguinte forma:

$$u_j(k+i) = \sum_{l=1}^{n_e^{(j)}} [m_{j,l}(i)] \cdot p_l^{(j)} \quad (2.50)$$

que pode ser formalizado como uma matriz de produto da forma:

$$u_j(k+i) = [M_j(i)] \cdot p^{(j)}; \quad p^{(j)} \in \mathbb{R}^{n_e^{(j)}} \quad (2.51)$$

Em que $M_j(i) \in \mathbb{R}^{n_e^{(j)}}$ é definido por:

$$M_j(i) = (m_{j,1}(i) \cdots m_{j,n_e^{(j)}}(i)) \quad (2.52)$$

Uma vez que $M_j(i)$ possa ser escrito como um vetor off-line pré-computado, a variável de decisão p pode ser obtida por meio de uma transformação direta da equação 2.52, para $j = 1, \dots, n_u$. A expressão do vetor de entrada $u(k+i)$ em um instante de tempo futuro $k+i$ pode ser representada da seguinte forma:

$$u(k+i) = \underbrace{\text{BlockDiag} \left(M_j(i)_{j=1}^{n_u} \right)}_{M(i)} \begin{pmatrix} p^{(1)} \\ \vdots \\ p^{(n_u)} \end{pmatrix} \quad (2.53)$$

$$u(k+i) = [M(i)]$$

Ao concatenar a equação acima para $i = 0, \dots, N - 1$, a expressão para a parametrização exponencial Π_e pode ser formalizada da seguinte forma:

$$\tilde{u} = \Pi_e \cdot p; \quad (2.54)$$

Em que Π_e é a matriz de seleção exponencial definida como:

$$\Pi_e := \begin{pmatrix} M(0) \\ \vdots \\ M(N - 1) \end{pmatrix} \quad (2.55)$$

Para reduzir o esforço computacional, uma nova função de custo (J) é redefinida com a incorporação da nova variável de controle (p) e da matriz (Π_e), levando à seguinte formulação:

$$J(p) = \frac{1}{2} p^T (\Pi_e^T \mathbf{H} \Pi_e) p + (\Pi_e^T \mathbf{F})^T p \quad (2.56)$$

sujeito a um novo conjunto de restrições definido da seguinte forma:

$$\mathbf{A}_r p \leq \mathbf{B}_r(k); \quad \tilde{\mathbf{u}}^{\min} \leq \Pi_e \cdot p \leq \tilde{\mathbf{u}}^{\max} \quad (2.57)$$

Em que \mathbf{A}_r e $\mathbf{B}_r(k)$ são as matrizes de desigualdades reduzidas da parametrização exponencial que representa a definição das restrições do problema de otimização (2.31), sendo formalizadas como:

$$\mathbf{A}_r = \begin{pmatrix} \mathbf{A}_{\text{ineq}} \cdot \Pi_e \\ -\Pi_e \\ +\Pi_e \end{pmatrix} ; \quad \mathbf{B}_r(k) = \begin{pmatrix} \mathbf{B}_{\text{ineq}}(k) \\ -\tilde{\mathbf{u}}^{\min} \\ +\tilde{\mathbf{u}}^{\max} \end{pmatrix} \quad (2.58)$$

2.7.5 O controlador linear quadrático

O Controlador Quadrático Linear (LQR) é um tipo de controlador baseado na teoria de controle ideal para projetar uma lei de controle de malha fechada. Sua aplicação se concentra na estabilização de sistemas lineares invariantes no tempo. Os sistemas LTI com custos quadráticos são especificados como sendo $\dot{x} = Ax + Bu$ e $J(x, u, t) = x^T \mathbf{Q}x + u^T \mathbf{R}u$, em que \mathbf{Q} e \mathbf{R} são matrizes simétricas de tamanho $n \times n$ e $m \times m$, respectivamente. A matriz \mathbf{Q} penaliza o erro em relação ao ponto de referência, e a magnitude das entradas de \mathbf{R} penaliza o esforço da variável de controle. Nesse controlador, a lei de controle é dada como sendo:

$$u(t) = -\mathbf{K}x(t) \quad (2.59)$$

Em que \mathbf{K} é o ganho de *feedback* de estado e $\mathbf{x}(t)$ é o vetor de estados, a matriz \mathbf{K} é definida por um processo de otimização que calcula uma função de custo que penaliza cada um dos estados e saídas, mas sem restrições:

$$\dot{\mathbf{x}}(t) = (\mathbf{A} - \mathbf{BK})\mathbf{x}(t) \quad (2.60)$$

O objetivo do controlador LQR é encontrar cada um dos autovalores da matriz $\mathbf{A} - \mathbf{BK}$ no semiplano esquerdo S para uma resposta que garanta a estabilidade e minimize a função de custo quadrático para as entradas e saídas do sistema real; a função de custo J (equação 2.61) é definida como:

$$J(u) = \int_0^{\infty} \mathbf{x}^T(t)Q\mathbf{x}(t) + u^T(t)Ru(t)dt \quad (2.61)$$

Nessa equação, a matriz $Q \in \mathbb{R}^{n_x \times n_x}$ e $R \in \mathbb{R}^{n_u \times n_u}$ penalizam os estados do sistema e a entrada de energia no sistema, respectivamente, Q e R . Com essas matrizes, é necessário encontrar um equilíbrio entre o rastreamento da referência e o esforço de comando necessário para atingir a referência, com a matriz de ponderação simétrica e semi definida positiva e a matriz R como a matriz de ponderação positiva definida $\mathbf{K} = R^{-1}B^T\mathbf{P}$ é definido como uma função de uma matriz desconhecida \mathbf{P} . \mathbf{P} é uma matriz simétrica de tamanho $n \times n$ que resolve a seguinte equação de Riccati [$A^T\mathbf{P} + \mathbf{P}A - \mathbf{P}B R^{-1}B^T\mathbf{P} + Q = 0$]. Há métodos numéricos disponíveis para resolver a equação de Riccati e encontrar \mathbf{P} .

O controlador Linear Quadrático (LQR) pode ser conceptualizado como uma formulação específica do MPC, diferenciando-se por suas condições particulares, tais como a ausência de restrições e um horizonte de predição infinito (ALAMIR, 2013). Enquanto o MPC busca resolver problemas de otimização sujeitos a restrições ao longo de um horizonte finito de tempo, o LQR simplifica esse problema, eliminando as restrições e permitindo que o horizonte de predição se estenda até o infinito. Matematicamente, o LQR busca minimizar uma função de custo quadrática, envolvendo a minimização da soma do custo presente e futuro, com uma penalização nos erros do estado e do controle. Essa abordagem simplificada do LQR torna-o particularmente adequado para sistemas onde as restrições podem ser negligenciadas. Em geral, a solução analítica da equação de Riccati para determinar os ganhos da matriz K é essencialmente equivalente à abordagem numérica utilizada no MPC quando o horizonte de predição tende ao infinito e não são aplicadas restrições (RAWLINGS; MAYNE; DIEHL, 2017).

2.8 Ciclo de desenvolvimento do produto automotivo

O ciclo de desenvolvimento do produto automotivo (também conhecido como modelo V) é usado atualmente para resolver problemas associados ao projeto de sistemas complexos em setores como robótica, aeroespacial ou automotivo. No setor automotivo, ele tem sido cada vez mais implementado devido à demanda por padronização de processos (por exemplo, ISO 26262) para garantir o desenvolvimento seguro ou rápido de sistemas complexos, como funções ADAS, sistemas de controle e componentes de E/S (HUANG, J.; TAN, 2015). Em geral, o modelo V possui as seguintes fases:

- Requisitos.
- Arquitetura.
- Projeto.
- Simulação e implementação de modelos.
- Verificação e validação.

Na figura 7, é possível verificar que o modelo V é dividido em dois estágios, sendo o primeiro a fase *downstream*, que representa o estágio de desenvolvimento do produto, que inclui desde o levantamento de requisitos, a arquitetura, o projeto, a implementação e o teste de unidade, e a fase *upstream* que representa o estágio de testes, verificações e validações. É neste estágio são realizados testes de integração, testes de sistema e testes de aceitação do sistema para a produção subsequente.

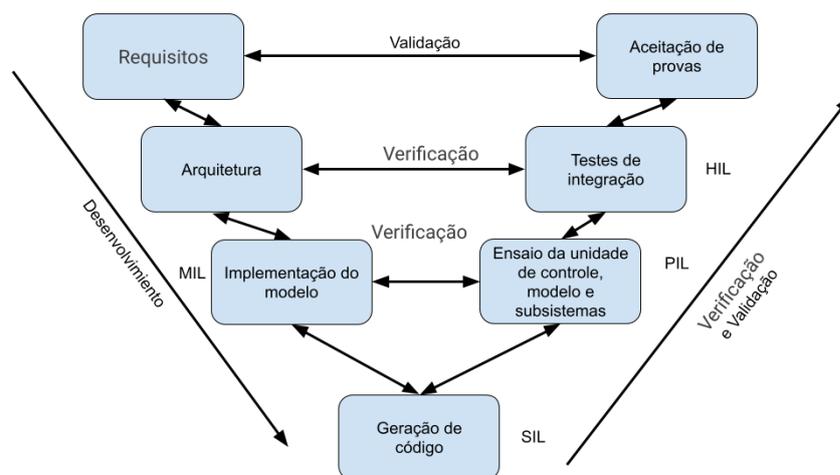


Figura 7 – Modelo V.

Existem métodos de verificação e validação no modelo V, que estão incorporados nas etapas diferentes estágios do modelo V. Os métodos estão descritos na figura 8 e em mais detalhes abaixo:

- **Model-In-the-Loop (MIL):** É o primeiro estágio de verificação e validação no ciclo V e consistem no teste de modelos individuais ou integrados em um ambiente de desenvolvimento que captura a maioria dos aspectos cruciais do sistema para sua funcionalidade. Após desenvolver um modelo matemático da planta, o MIL é usado para validar se o módulo do controlador pode controlar a planta conforme desejado e verifica se a lógica do controlador produz o comportamento desejado. Nessa etapa, a estratégia de controle e o modelo da planta a ser controlada são estabelecidos no ambiente de desenvolvimento mencionado acima.
- **Software-In-the-Loop (SIL):** Após a verificação do modelo no estágio anterior, o código da estratégia de controle é gerado a partir do modelo do controlador e usado para substituir o bloco do controlador do estágio anterior; o principal objetivo desse método é verificar a funcionalidade do software embarcado, os algoritmos lógicos propostos e os loops de controle sem que eles sejam incorporados ao hardware real. O que envolve o estabelecimento do código do controlador em uma linguagem como C ou C++. Esse processo é usado para verificar se o estágio anterior do modelo está correto e se a conversão da estratégia de controle em instruções de programa do controlador está livre de erros de sintaxe.
- **Processor-In-the-Loop (PIL):** Nesse estágio, o código e a lógica do controlador serão carregados em um processador real, em vez do computador usado nas duas etapas anteriores, e simulações de malha fechada serão executadas com o modelo da planta. Isso garante que o hardware possa executar o software do controlador verificando a capacidade de memória, do processo de conversão analógica digital ou a carga computacional. Os resultados da simulação indicarão se é necessário retornar às fases anteriores ou se o código e a lógica do controlador precisam ser modificados.
- **Hardware-In-the-Loop (HIL):** Essa é a última etapa antes de conectar o processador com o código de controle incorporado à planta real. Nessa etapa, o modelo da planta é embarcado em um computador em tempo real (RTPC). Essa plataforma trabalha com modelos criados usando software de modelagem virtual e inclui recursos reais, como módulos de comunicação (digital, analógico, CAN, Ethernet). Com a ajuda dessa etapa, é possível capturar a interação em tempo real entre a estratégia de controle e o ambiente de hardware real e pode identificar falhas e outros problemas que não foram encontrados nas etapas anteriores. O HIL é necessário em várias aplicações, como aeroespacial e automotiva, especialmente em funções críticas de segurança.

Uma das principais vantagens da utilização destes ciclos de verificação e validação está na execução de testes iterativos e incrementais em cada estágio do processo, o que, juntamente com a forte integração entre as fases de desenvolvimento e validação, permite a detecção precoce de problemas. Essa abordagem resulta em uma redução significativa dos

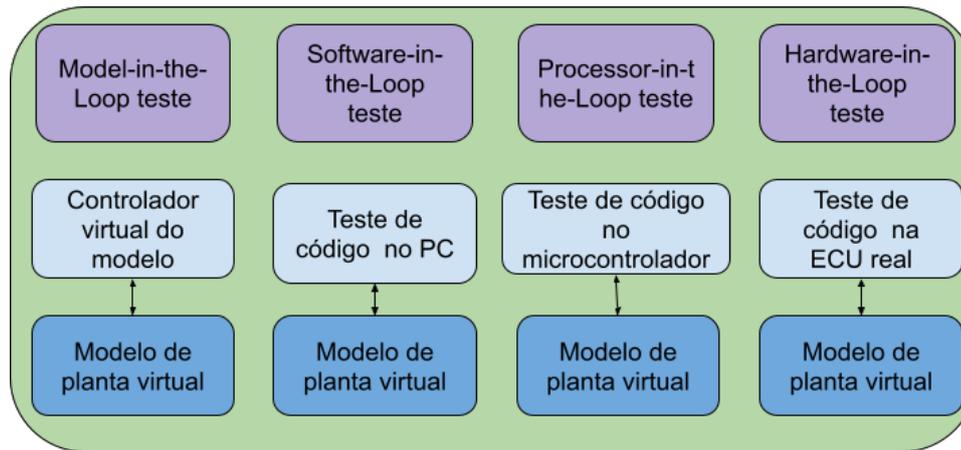


Figura 8 – MIL, SIL, PIL e HIL

custos e dos esforços necessários para solucionar bugs no final do ciclo de desenvolvimento do produto automotivo. Além disso, atenção constante dada à validação e à verificação no ciclo V leva a um nível mais alto de capacidade de manutenção e qualidade tanto no software quanto nos sistemas desenvolvidos e contribui substancialmente para a melhoria geral da eficiência e da qualidade no desenvolvimento de sistemas altamente complexos.

2.8.1 Solucionadores quadráticos

Os algoritmos de otimização combinam geralmente o problema de otimização e a função de custo para encontrar uma solução viável em cada instante de tempo. Esses algoritmos permitem a execução do MPC em tempo real, o que é crucial para aplicações de controle automotivo. Quatro algoritmos de otimização foram investigados para projetar e implementar os controladores MPC:

- Métodos de pontos interiores
- Métodos de conjunto ativo
- Métodos de direção alternada de multiplicadores
- Método de Newton

Os métodos de ponto interior são amplamente utilizados para resolver problemas de otimização convexa com restrições de desigualdade. Eles modelam normalmente as restrições do problema usando funções de penalidade parametrizadas (também conhecidas como funções de barreira (PANDALA; DING; PARK, 2019)). Esses métodos promovem a convergência do algoritmo de otimização, independentemente do tipo de condições, dados e tamanho. Os problemas de otimização sem restrições com diferentes funções de barreira podem ser resolvidos em cada interação até que o ponto estável seja alcançado. Os métodos podem ser subdivididos em barreiras primárias e barreiras primárias-duais. A primeira

substitui as restrições de desigualdade do problema de programação quadrática (QP), incluindo uma função de barreira ponderada por objetivo. A última combina um loop interno e externo de barreira primal reduzindo o peso da barreira a cada iteração do método de Newton (FERREAU et al., 2014).

Os métodos de conjunto ativo (ASMs) estabelecem um conjunto de trabalho para resolver o problema de QP resultante com restrições de igualdade. O conjunto de trabalho selecionado (um subconjunto linearmente independente de restrições ativas) é atualizado continuamente até que os valores ideais sejam encontrados. Os ASMs são classificados como métodos primários, duais e paramétricos. Os métodos primários produzem uma sequência de interações primárias viáveis até que a viabilidade dupla com uma solução ideal seja alcançada. Da mesma forma, os métodos duais geram uma sequência de interações viáveis duais até que a viabilidade primal seja alcançada com uma solução ideal. No caso estritamente convexo, isso é equivalente a resolver o dual do problema quadrático com um método de conjunto ativo primário. Por outro lado, os métodos paramétricos consideram o conceito de rastreamento de uma solução para a homotopia linear. A homotopia linear, parametrizada por $t_a \in [0, 1]$, é frequentemente utilizada para fazer a transição entre um problema de programação quadrática com uma solução conhecida ($t_a = 0$) e um problema que precisa ser resolvido ($t_a = 1$) (FERREAU et al., 2014). Embora os ASMs sejam reconhecidos pela convergência rápida e pela inicialização a quente em pequenos problemas de QP, eles são bastante complexos de implementar ou até mesmo não são robustos para o término antecipado, além de serem difíceis de converter da teoria para a prática. As aplicações atuais são comumente encontradas em problemas de pequeno e médio porte (LIAO-MCPHERSON; KOLMANOVSKY, 2020).

Os Métodos de Multiplicadores de Direção Alternada (ADMMS), também conhecidos como Métodos de Primeira Ordem (FOMs), calculam uma solução ideal usando as informações de primeira ordem sobre a função de custo. Como os FOMs usam apenas informações de sub gradiente, ele pode ter uma configuração *start warm* que o leva a taxas de execução rápidas. Se as restrições forem simples, os métodos de gradiente projetado se tornam atraentes devido à simplicidade e à disponibilidade de limites de complexidade rígidos. Vários métodos FO resolvem o problema dual usando a subida do gradiente ou atualizações multiplicativas, por exemplo, (LIAO-MCPHERSON; KOLMANOVSKY, 2020). Apesar dos avanços dos FOMs, eles não conseguem detectar a inviabilidade primal ou dual. Além disso, a convergência nos FOMs é altamente dependente dos dados e da entrada do parâmetro de tamanho de etapa do algoritmo (STELLATO et al., 2020). Além disso, os FOMs fornecem uma convergência assintótica mais lenta do que os e ASMs.

O método newtoniano (NM) realiza uma aproximação de duas funções diferenciáveis usando uma expansão de Taylor de segunda ordem. O NM busca um valor mínimo da função aproximada de forma interativa (HAN; LEE, H., 2021). Esse método é baseado

em aproximações sucessivas da função objetiva (função quadrática local) e na solução do problema quadrático resultante. Alguns benefícios do NM incluem a convergência quadrática e a redução de erros em cada interação do algoritmo. No entanto, o NM exige geralmente o cálculo da matriz Hessiana, o que se torna difícil de obter em alguns casos. No entanto, esse método não pode atingir a convergência global se o ponto inicial não estiver próximo o suficiente do valor ideal.(ADLER; HU, Z. T.; LIN, 2020).

Os algoritmos de otimização mencionados acima têm sido utilizados há décadas em muitas aplicações. Além disso, algumas variantes também foram propostas para a implementação de sistemas em tempo real.

3 Estado do arte

A revisão sistemática da literatura realizada para o desenvolvimento desta dissertação permitiu identificar os trabalhos mais recentes e com relevância para a realização deste trabalho. Ela permitiu orientar a pesquisa sobre as funções LKAS e seu potencial de melhoria nas áreas de autonomia, segurança e conforto. O desenvolvimento desse método foram realizadas as etapas mencionadas abaixo:

- Planejamento
- Coleta de dados
- Análise temática de dados

3.1 Planejamento

Nessa primeira etapa, as perguntas de pesquisa são estabelecidas para obter uma revisão mais completa possível da literatura ou, em outras palavras, para determinar o problema de pesquisa. Em particular, nesta pesquisa, as perguntas foram feitas para abranger as áreas mais importantes das funções ADAS e, mais especificamente, das funções do LKAS:

- O que é o *Lane Keeping Assist System* (LKAS)?
- Quais são os componentes presentes no Lane Keeping Assist System (LKAS)?
- O que é utilizado no processo de simulação do LKAS?
- Quais são os benefícios e as limitações do LKAS?

3.2 Coleta de dados

Continuando com a etapa 2, as palavras-chave de cada pergunta de pesquisa foram combinadas para a formulação de uma *string* de pesquisa (*search string*). As bases de dados foram selecionadas para a pesquisa: SCOPUS, IEEE xporer e Web of Science, como resultado das combinações de palavras-chave em cada banco de dados durante o período de 2012 até 2022, foram obtidas as seguintes combinações:

- SCOPUS (440 RESULTADOS): (("Lane Keeping Assist System"OR "Pilot Assist"OR "Direct Adaptive Steering"OR "Lane Centering Assist"OR "Automated Lane Keeping"OR

lkas* OR lka* OR ldw* OR "lane departure warning"OR "active steer"OR autosteer* OR lane-keeping OR "Lane keeping"OR "Lane Keeping Assist"OR "Lane Detection"OR "Self-Driving CarS") AND (behavior* OR interaction* OR performance) AND ("control strategies"OR control* OR "control implementation"OR "control techniques") AND (simulation* OR HIL OR "Hardware in the loop"OR real-time))

- IEEE xplore (202 RESULTADOS): ("All Metadata":Lane Keeping Assist System) OR ("All Metadata":Lane Centering Assist) OR ("All Metadata":lane departure warning) OR ("All Metadata":Lane keeping) OR ("All Metadata":Lane Detection) AND ("All Metadata":control strategies) AND ("All Metadata":control) AND ("All Metadata":HIL) AND ("All Metadata":Hardware in the loop)
- Webofscience(281 RESULTADOS): (("Lane Keeping Assist System"OR "Pilot Assist"OR "Direct Adaptive Steering"OR "Lane Centering Assist"OR "Automated Lane Keeping"OR lkas* OR lka* OR ldw* OR "lane departure warning"OR "active steer"OR autosteer* OR lane-keeping OR "Lane keeping"OR "Lane Keeping Assist"OR "Lane Detection"OR "Self-Driving CarS") AND (behavior* OR interaction* OR performance) AND ("control strategies"OR control* OR "control implementation"OR "control techniques") AND (simulation* OR HIL OR "Hardware in the loop"OR real-time))

Após a soma de todos os artigos, foram obtidos 923 meta dados de artigos nas bases mencionadas. Após o *download* dos metadados e combinação em uma planilha do excel, procedeu-se com a seleção dos artigos mais relevantes. Quando a análise de títulos duplicados (EXCEL) foi realizada, foram excluídos 279 artigos, resultando em 644 artigos.

A partir de um processo de filtragem interativo e incremental, buscou-se selecionar os artigos mais relevantes para a pesquisa. Os filtros funcionam como ferramentas de seleção de recursos, que permitem a extração e levantamento dos artigos relevantes. Na segunda etapa de filtragem foi utilizada a seleção automática de títulos mediante a agrupação dos nomes dos artigos e um arquivo Excel. Para este filtro foram selecionadas duas palavras-chave que representavam o foco principal da pesquisa (*Lane keeping e Control*), após obtidos 48 artigos que incluíam essas palavras em seu título e foram representados como o agrupamento Y1=48. Os artigos restantes representaram a nova população para o próximo filtro, no qual foi a seleção manual de títulos a partir da nova população representada por (Y0-Y1). Nesta etapa foi realizada a leitura de cada um dos títulos dos artigos por dois pesquisadores principais, fornecendo uma classificação de Y (*Yes*), N (*No*) ou M (*Maybe*). A nova população Y5=315 é composta pela soma dos artigos que obtiveram a classificação representada como Y2=8, Y3=58 e Y4=201, somando-se a ele o agrupamento Y1=48. Na tabela 1 é mostrado o processo de seleção dos artigos.

O seguinte filtro foi a seleção automática de resumos, com o uso do tópico principal da pesquisa; nesse caso, foi "Lane Keeping"; os artigos que contêm essa palavra são representados

Tabela 1 – Processo de classificação de artigos para leitura integral

PASSO	Ação	Y	Total de artigos
1	Combinacao de 3 bases de dados (EBSCO, Web of Science, Scopus)		923
2	Duplicatas removidas e informações irrelevantes descartadas		279
3	Nova população	Y0	644
SELEÇÃO AUTOMÁTICA DE TÍTULOS			
-	População	Y0	644
4	Seleção automática de títulos (lane keeping E controle)	Y1	48
3	Nova população	Y0-Y1	596
SELEÇÃO MANUAL DE TÍTULOS			
-	População	Y0-Y1	596
4	Pesquisador 1 / Pesquisador 2 (Y&Y)	Y2	8
5	Pesquisador 1 / Pesquisador 2 (Y&M OU M&Y)	Y3	58
6	Pesquisador 1 / Pesquisador 2 (Y&N OU M&M OU N&Y)	Y4	201
7	Pesquisador 1 / Pesquisador 2 (N&N OU N&M OU M&N)		329
-	Nova população (Y1+Y2+Y3+Y4)	Y5	315
SELEÇÃO AUTOMÁTICA DE RESUMOS			
-	População	Y5	315
8	Busca por "Lane keeping"no resumo (VERDADEIRO)	Y6	131
9	Busca por "Lane keeping"no resumo (FALSO)		184
-	Nova população	Y5-Y6	184
SELEÇÃO MANUAL DE RESUMOS			
-	População	Y5-Y6	184
10	Pesquisador 1 / Pesquisador 2 (Y&Y)	Y7	6
10	Pesquisador 1 / Pesquisador 2 (Y&M OU M&Y OU N&Y)	Y8	73
10	Pesquisador 1 / Pesquisador 2 (N&N OU N&M OU M&N OU Y&N)		105
-	Nova população (Y6+Y7+Y8)	Y9	210
SELEÇÃO DE TEXTO COMPLETO			
-	População	Y9	210
11	Não encontrado ou Sem acesso		38
12	Crterios de inclusão (Contribuição = >2; Teoria >=2; Metodologia >=1; Análise de dados = TODOS)		69
13	Crterios de exclusão (Contribuição <=1; Teoria <=1)		103
RESULTADOS			
TOTAL DE ARTIGOS INCLUÍDOS			69
TOTAL DE ARTIGOS EXCLUÍDOS			854
PORCENTAGEM DE ARTIGOS INCLUÍDOS			8%

por Y6=131. De maneira semelhante que na anterior fase depois desse filtro automático foi feito a seleção manual de resumos com a nova população representada por Y5-Y6=184, como no título, a classificação manual foi feita pelo uso de Y (*Yes*), N (*No*) ou M (*Maybe*) de cada artigo após a leitura do resumo, essa classificação foi feita pelos dois pesquisadores e, depois disso, foram selecionados aqueles representados por Y7=6 e Y8=73.

Na última fase dos filtros que foram aplicados é a *full-text screening*, nessa etapa, todos os artigos compreendendo Y6=131, Y7=6, Y8=73 resultando em Y9=210, após atingir esse número de artigos, procedemos à busca e ao download. Por fim foram aplicados os critérios de inclusão e exclusão, onde para a seleção dos artigos a serem lidos na íntegra, foi feita uma avaliação de cada um deles, essa avaliação foi realizada por meio da revisão rápida de cada artigo e da atribuição de um valor de 0 a 3 para contribuição, teoria, metodologia e análise de dados, ao final dessa avaliação foram selecionados 69 artigos para leitura total, o que representa 8% do (*dataset*) de artigos encontrados.

3.3 Análise temática

Nesta seção, o pesquisador busca responder às perguntas de pesquisa com base na literatura. às perguntas feitas na seção de planejamento. Após a leitura dos artigos selecionados, eles serão descritos em termos dos principais temas que agrupam as perguntas, para a elaboração desse etapa foi utilizado o software ©Nvivo o qual foi utilizado para fazer o agrupamento (clusterização) das informações obtidas nos artigos, como um esforço do pesquisador em responder às perguntas de pesquisa com base na análise da literatura.

- Tema 1: Definição e funcionamento do LKAS.
 - O que é o Lane Keeping Assist System (LKAS)?
- Tema 2: Sensoriamento para o LKAS.
 - Quais são os componentes presentes no Lane Keeping Assist System (LKAS)?
- Tema 3: Modelagem, simulação e estratégias de controle do LKAS
 - O que é utilizado no processo de simulação do LKAS?
- Tema 4: Benefícios e limitações do LKAS.
 - Quais são os benefícios e as limitações do LKAS?

3.3.1 Tema 1: Definição e funcionamento do LKAS

No campo dos Sistemas Avançados de Assistência ao condutor (ADAS), as funções podem ser classificadas conforme a dinâmica que envolve seu funcionamento no veículo e são geralmente divididas em duas categorias principais: controle longitudinal e controle lateral (RAJAMANI, R., 2011). Essas funções são implementadas em veículos para realizar o controle da velocidade e da direção do veículo simultaneamente (LIANG, J. et al., 2021). O sistema de manutenção de faixa (LKAS) é um componente essencial do conjunto de tecnologias (ADAS) que controlam a dinâmica lateral do veículo, cujo objetivo é permitir que os veículos sigam uma trajetória dentro da faixa sem a intervenção direta do condutor (LEE, K.; LI, S. E.; KUM, 2018). Além disso, a função LKAS integra uma funcionalidade conhecida como Aviso de Saída de Faixa (LDW), que alerta o condutor sobre situações em que o veículo está saindo da faixa (KAMAT, 2020).

Nos primeiros sistemas LKAS, uma das soluções apresentadas foi a correção da direção do veículo por meio da frenagem diferencial, (XU, X.; GRIZZLE et al., 2017). No entanto, os sistemas atuais evoluíram para fornecer torque à coluna de direção, mantendo assim o veículo em sua faixa de rodagem, emitindo, muitas vezes, alerta aos condutores por meio de mensagens visuais ou sonoras (CHEN, Z.; LI, L.; HUANG, X., 2018). Esses

sistemas foram os primeiros a serem utilizados no setor, pois simplesmente realizam o cálculo do tempo de cruzamento da linha. Dependendo do modo de operação selecionado ou das funções integradas em um veículo, o sistema de prevenção de saída de faixa assume o papel LKAS (atuação) ou LDWS (advertência) (MERAH; HARTANI; DRAOU, 2016), sendo o primeiro responsável por atuar no ângulo de direção do veículo e o segundo focado em alertar contra possíveis saídas de faixa. Essas funções apoiam o condutor durante situações em que a sonolência, o consumo de álcool ou outras distrações podem induzir uma saída de faixa, antecipando e induzindo reações imediatas do condutor (XU, L. H.; HU, S. G.; LUO et al., 2015).

Em geral, o principal objetivo do LKAS é manter o veículo na linha central da faixa (LIANG, J. et al., 2021). Isso é alcançado por meio do uso de computação e operação baseada em sensores, como câmeras e radares, integrados a uma plataforma de computação de alto desempenho, que permite o processamento de dados que vem desde dentro e fora do veículo, essas operações contribuindo para uma condução mais segura e precisa. Por meio da solução de algoritmos que permitem manter uma zona de segurança para o veículo, essa função também é projetada para a zona de segurança do veículo (CHEN, J. et al., 2021). Esta função tem demonstrado que permite para minimizar a carga de trabalho a que os condutores estão sujeitos em viagens longas (HUANG, J.; TAN, 2015).

O LKAS é executada em tempo real (NGUYEN, A.-T.; SENTOUH; POPIEUL, 2018) usando vários sensores e, principalmente, uma câmera para detectar as linhas da faixa e, com essas informações, determinar o erro em relação ao centro da faixa, à curvatura da estrada (YAN, N.; LIU, C.; SUN, 2021) e o erro do ângulo de direção (MERAH; HARTANI; DRAOU, 2016). Para tanto, vários métodos tem sido empregados no processo de reconhecimento das características da faixa, como a transformada de Hough (HTET; KIONG; XINXIN, 2015), ou usando redes neurais (CHEN, Z.; LI, L.; HUANG, X., 2018).

Os dados detectados são enviados para uma Unidade de Controle Eletrônico (ECU) (KIM, D. J. et al., 2018) que, por meio da execução de várias estratégias de controle como PID, LQR, redes neurais ou MPC. E considerando a velocidade do veículo e o torque medido exercido pelo condutor, calcula o ângulo e o torque de referência para o sistema de direção alcançar, subsequentemente, a ECU controla o motor para gerar o torque necessário para fornecer assistência à direção (KIM, W.; SON; CHUNG, 2015).

Em geral, o LKAS calcula o ângulo de direção a partir dos sinais do veículo e das medições da geometria da estrada para permanecer dentro dos parâmetros operacionais seguros. Isso envolve manter os deslocamentos laterais e os erros de ângulo de direção mínimos para evitar a saída da faixa (SARAÇOĞLU; ÜLEŞ; SCHMIDT, 2018). Desse modo, para garantir seu desempenho e confiabilidade é necessário um projeto algorítmico preciso para o controle de manutenção de faixa (GE et al., 2021) Em cenários de condições reais de estrada, o desafio se torna substancial, pois o veículo deve permanecer dentro da faixa,

mantendo o conforto para os passageiros, apesar da presença de distúrbios (por exemplo, curvas na estrada, rajadas de vento e risco de aquaplanagem) e incertezas nos parâmetros do veículo (por exemplo, massa do veículo, centro de gravidade e rigidez dos pneus em curvas) (LEE, K.; LI, S. E.; KUM, 2018).

Nesse contexto, o LKAS deve implementar um algoritmo em tempo real dos sistemas encarregado de calcular a trajetória de viagem de um veículo e determinar o erro de deslocamento lateral (GUO, J. et al., 2020), definido como a distância entre a posição atual do centro de gravidade do veículo e o centro da faixa (HTET; KIONG; XINXIN, 2015). Esses elementos destacam a complexidade e a variedade de condições às quais o sistema LKAS deve se adaptar para uma operação eficiente e precisa. Com base nesta análise é possível inferir a seguinte descoberta:

Descoberta: *o Sistema de Assistência à Manutenção na Faixa de Rodagem (LKAS) desempenha um papel importante na segurança e no conforto do usuário, mantendo os veículos em sua faixa. Em geral, ele utiliza sensores para determinar sua posição dentro da faixa e realizando cálculos para ajustar o torque da direção. Os primeiros sistemas LKAS possuíam um mecanismo de atuação baseado em frenagem diferencial, ao passo que os atuais aplicam torque na coluna de direção e emitem alerta ao condutor para enfrentar os desafios das condições reais da estrada devido a distúrbios e à variabilidade dos parâmetros do veículo.*

3.3.2 Tema 2: Sensoriamento para a função LKAS

O LKAS pertence ao controle lateral do veículo e baseia-se principalmente no monitoramento e na compreensão do ambiente à frente do veículo com o objetivo principal de manter o veículo no centro da faixa (HU, C. et al., 2019). Isso é possível com a execução de comandos enviados para modificar o ângulo das rodas dianteiras. As informações do sistema e do ambiente podem ser obtidas diferentes tipos de sensores, como câmeras, GPS, IMU, sensores magnéticos integrados à estrada, radar, LIDAR e sensores ultrassônicos (KUO; LU, Y.; YANG, S., 2019). Os sensores mencionados fornecem dados usados como entradas necessárias para a tomada de decisão LKAS, uma vez que o ângulo é calculado, é enviado para a ECU da direção (KIM, D. J. et al., 2018).

Embora atualmente seja possível calcular a rota e a posição do veículo usando esses sensores, a precisão das informações fornecidas é limitada pela resolução, disponibilidade e atualização em tempo real (KUO; LU, Y.; YANG, S., 2019). Como resultado, o principal sensor utilizado nesse sistema é geralmente uma câmera (SON; LEE, S.-H.; CHUNG, 2013), que permite o processamento da cena de trânsito por meio de algoritmos de reconhecimento de linha. Esses algoritmos tradicionalmente envolvem técnicas como aprimoramento de cores, detecção de bordas, uso de informações de visão aérea e mapeamento de perspectiva reversa (WANG, H. et al., 2019), aprendizagem profunda, entre outras (HTET; KIONG; XINXIN, 2015).

Outro sensor que deve ser mencionado por sua exatidão e capacidade de identificar o ambiente é o LIDAR, que pode ser usado no reconhecimento do ambiente e fundido com outros dados para manter o veículo no centro da faixa (REDDY et al., 2020). No entanto, seu alto custo limita a aplicação em veículos, de modo que os fabricantes optam pelo uso de sensores mais baratos, como a câmera, tendo levado ao desenvolvimento de diferentes métodos para o reconhecimento de linhas nas estradas (WU; WANG, L.-H.; WANG, K.-C., 2018).

A maioria dos métodos de detecção de faixa baseados em visão compartilha três estágios comuns: no primeiro estágio, o pré-processamento da imagem, transformando-a em uma vista aérea usando o mapeamento de perspectiva inversa, (ou vista de pássaro). Já o segundo estágio é a extração de recursos da faixa por meio de métodos baseados em bordas e o terceiro estágio é o ajuste da faixa por meio de consenso de amostras aleatórias (RANSAC) combinado com métodos para obter um melhor nível de reconhecimento e rastreamento (HTET; KIONG; XINXIN, 2015), como o ajuste de par hiperbólico, transformada de Hough, mínimos quadrados ou B-spline e ajuste de clotoide (WU; WANG, L.-H.; WANG, K.-C., 2018).

A transformação da visão de pássaro permite uma representação mais simplificada do ambiente da estrada, o que facilita a detecção de características da faixa na frente do veículo (JIANG, R. et al., 2009). A extração de características da faixa, seja por meio de bordas ou cores, permite a identificação de regiões de interesse para análise posterior. Por fim, o ajuste da faixa usando técnicas e algoritmos de ajuste de curva fornece uma estimativa precisa da geometria da faixa, o que é essencial para a navegação e o controle do veículo (RUI, 2022).

Em contraste com os algoritmos clássicos, foram apresentadas abordagens de aprendizado de ponta a ponta utilizando redes neurais convolucionais para a detecção das principais características da faixa (CHEN, Z.; LI, L.; HUANG, X., 2018) o que demonstrou melhorias significativas no reconhecimento de faixas diante das perturbações apresentadas por condições externas, como eventos naturais ou artificiais, como luzes na câmera (BOJARSKI et al., 2016). No entanto, esse algoritmo possui como desvantagem a necessidade de uma grande quantidade de informações para treinamento (LIU, X.; LIANG, J.; XU, B., 2019). Somando-se a isso, ele pode apresentar um desvio na resposta encontrada por uma superestimação ou subestimação nos dados obtidos.

Atualmente, estão disponíveis no mercado sensores ópticos especializados na detecção de linhas, como o módulo Mobileye 660 (YOFFIE, 2014), que fornece detalhes sobre as principais características da faixa, incluindo deslocamento lateral, erro no ângulo de direção, curvatura e sua derivada, qualidade da detecção de linhas, alcance máximo detectável e tipo de marcação da faixa. Esse módulo foi utilizado por vários pesquisadores (XU, S. et al., 2020), (LEE, J. et al., 2014), (LIU, C. et al., 2015). Em seus estudos, eles usaram essas infor-

mações para aproximar uma função que descreve o modelo atual da estrada, além de prever a geometria da estrada.

Outros sensores, como GPS e IMU, são usados para capturar informações essenciais relacionadas à posição, velocidade e orientação do veículo (FARAG, W., 2021). Esses dados são de vital importância para a execução do controle lateral do veículo na função LKAS, no entanto, é importante mencionar que cada um desses sensores pode apresentar diferenças em sua taxa de amostragem, como a câmera e o GPS, essas diferenças na taxa de amostragem apresentam desafios em termos de sincronização de dados (SON; KIM, W. et al., 2014). Para resolver esse problema, algoritmos de filtragem têm sido usados na literatura, como o conhecido filtro de Kalman ou o filtro de Kalman estendido (YANG, J.; KIM, S.; HUH, 2020), esses filtros demonstraram não apenas sua capacidade de unir todos os dados do sensor, mas também de estimar e fornecer variáveis não mensuráveis e diminuir o ruído no processo de medição. Com base nesta análise é possível se chegar a seguinte descoberta:

Descoberta: *Diversos sensores podem ser aplicados na função LKAS, sendo a câmera o principal sensor usado devido à sua implementação econômica. Esse componente desempenha um papel fundamental na captura de informações da cena de trânsito, permitindo a determinação de características como posição, ângulo e geometria da faixa. A partir desses dados, algoritmos são empregados no processamento e na extração de informações, sendo necessário, em muitos casos, utilizar filtros para estimativa e remoção de ruídos para manter o veículo seguro e alinhado na faixa.*

3.3.3 Tema 3: Modelagem, simulação e estratégias de controle do LKAS

No contexto da assistência de manutenção de faixa (LKAS), é útil usar um modelo matemático que represente a dinâmica lateral do veículo, permitindo o desenvolvimento de controladores para manter o veículo em uma referência específica dentro da faixa, um modelo mais complexo consegue uma representação com mais precisão (YAKSHA et al., 2021). Existem diferentes modelos presentes na literatura para representar a da dinâmica de veículos, sendo o modelo de bicicleta um dos mais utilizados atualmente. Este modelo é amplamente utilizado no desenvolvimento de sistemas de controle de estabilidade, mas também é usado em outros sistemas, como o controle de assistência à manutenção de faixa LKAS (KANG; KIM, W.; CHUNG, 2018). Este modelo se baseia no princípio de que o veículo se comporta como uma bicicleta, o que significa que as rodas traseiras estão unidas em uma só, assim como as rodas dianteiras (RAJAMANI, R., 2011), simplificando sua representação. Também note-se que esse modelo é geralmente usado em sua forma linear, mas a representação do modelo linear pode se distanciar da representação de um veículo real quando as condições de direção são variáveis ou extremas, mas utilizar um modelo não linear pode deixar um grande custo computacional (LIU, C. et al., 2015).

Em geral, o modelo não linear reproduz com maior exatidão o comportamento do veículo quando comparado com um modelo linear em condições extremas devido às não linearidades dos pneus (BENINE-NETO; MAMMAR, 2012). Nesses casos, é necessária uma representação mais complexa para modelar adequadamente essas condições específicas (SATRIA et al., 2022). Diferentes autores desenvolveram seus controladores com base no modelo de bicicleta (RAJAMANI, R., 2011), (KANG; LEE, S.-H.; CHUNG, 2016), (SAMUEL et al., 2021), usando diferentes graus de liberdade para descrever, as características do sistema. Porém, diferentes autores utilizaram o modelo linear porque com utilização de simplificações consegue diminuir o tempo computacional e aproxima-se para uma representação mais perto da realidade (SHU et al., 2017).

Na atualidade pode se encontrar diferentes modelos utilizados na função LKAS, cada um dos modelos encontrados pode utilizar combinação de modelos de rastreamento de trajetória com utilização da câmera (AN et al., 2020), com modelos do erro com respeito na faixa (RAJAMANI, R., 2011) ou considerando o torque aplicado na coluna da direção (KIM, W.; SON; CHUNG, 2015). Todos esses modelos podem variar no número de graus de liberdade, como por exemplo de 2 D.O.F (HU, C. et al., 2019), (GUO, J. et al., 2020), (LEE, J. et al., 2014)], 3 D.O.F (SON; KIM, W. et al., 2014), (SALT DUCAJÚ et al., 2021), 6 D.O.F (RAJAMANI; RAJESH, 2012), 7 D.O.F (YAKSHA et al., 2021), (CHEN, C. et al., 2012) até 14 D.O.F (SETIAWAN; SAFARUDIN; SINGH, 2009). Os graus de liberdade representam liberdade dos movimentos longitudinais, laterais e de guinada, bem como podem representar os movimentos presentes nas rodas do veículo. Modelos com mais graus de liberdade podem representar de maneira mais fidedigna um veículo real, não tanto pelo aumento do número de graus de liberdade que a complexidade dos cálculos aumenta, também se forem consideradas as não linearidades dos sistemas poderia chegar que a implementação em controladores de tempo real não é viável (SALT DUCAJÚ et al., 2021).

É por isso que algumas simplificações são feitas para os modelos como as seguintes:

- No mundo real, as dinâmicas lateral e longitudinal do veículo são acopladas, porém, é normal descobrir que são modeladas de forma independente (JIANG, J.; ASTOLFI, 2018).
- As duas rodas dianteiras e as duas rodas traseiras são combinadas em uma única roda dianteira e uma única roda traseira (RAJAMANI; RAJESH, 2012).
- Os termos não lineares podem ser linearizados assumindo os pequenos ângulos de direção e deslizamento (CHU et al., 2018).
- A velocidade longitudinal é constante e positiva. Também a superfície da estrada seja plana e não tenha gradientes na estrada (PARKASH; SWARUP, 2020).

- A taxa de guinada pode ser obtida usando a geometria de Ackerman (YANG, J.; KIM, S.; HUH, 2020).

No entanto, alguns autores consideram a variação da distribuição de carga e as condições da estrada, além da característica altamente não linear encontrada na dinâmica das rodas (LI, W. et al., 2021). Com isso as forças laterais são representadas na sua maioria pela fórmula mágica de Pacejka (BENINE-NETO; MAMMAR, 2012). Além disso, diversos autores redefiniram um modelo dinâmico que permite usar as variáveis de estado do modelo em termos de erro de posição e orientação com relação à faixa de direção (RATHAI; AMIRTHALINGAM; JAYARAMAN, 2017), (SON; KIM, W. et al., 2014), (YAN, N.; LIU, C.; SUN, 2021).

Como mencionado anteriormente, o modelo matemático da planta permite o desenvolvimento de controladores. Já os controladores permitem manter ou alterar o estado de funcionamento da planta. Na literatura, diferentes controladores foram desenvolvidos para o LKAS como o controle PID (Proporcional-Integral-Derivativo) que, em geral, é um dos controladores mais implementados na indústria. Nas funções LKAS, diferentes autores o utilizaram isoladamente (LEE, K.; LI, S. E.; KUM, 2018), (SAMUEL et al., 2021) ou PID em cascata (SARAÇOĞLU; ÜLEŞ; SCHMIDT, 2018), os controladores fuzzy foram aplicados no LKAS usando regras linguísticas e conjuntos fuzzy para modelar e controlar sistemas em entornos complexos (NGUYEN, A.-T.; SENTOUH; POPIEUL, 2018), (BIAN et al., 2017), o controlador de H-infinito é uma ferramenta de controle robusta poderosa, equipada com restrição do ruído para função LKAS (YAN, N.; LIU, C.; SUN, 2021).

O controlador LQR é uma abordagem que busca minimizar uma função de custo que penaliza o erro das variáveis rastreadas do sistema e a ação de controle. Nos sistemas LKAS, esse controlador tem sido utilizado graças à ponderação realizada pelas matrizes de penalidade que permitem manter um equilíbrio entre o rastreamento da referência (deslocamento lateral) e a suavidade do comando de controle (ângulo de direção) (SHERIF et al., 2019). Já o controle MPC que é uma técnica de controle avançada que usa um modelo do sistema para prever como ele se comportará e, em seguida, otimiza uma sequência de ações de controle para encontrar, no caso da função LKAS, o ângulo das rodas dianteiras para manter o veículo no centro da faixa (KAMAT, 2020)

O controlador MPC apresenta diferentes vantagens sobre outras técnicas. Esta técnica pode prever o comportamento futuro do veículo para um horizonte de tempo definido. Esse recurso ajuda o controlador a fornecer ações de controle mais precisas que levam em conta a dinâmica do veículo (FARAG, W. A., 2020). Em particular, o MPC tem sido amplamente utilizado no rastreamento de referência de trajetória para a função LKAS, pois permite calcular e otimizar a sequência de entradas de controle futuras usando um modelo explícito (SALT DUCAJÚ et al., 2021). O MPC tem a capacidade de otimização on-line e manipulação

de restrições, o que representa uma oportunidade para alcançar a alocação de controle dinâmico enquanto satisfaz as restrições de segurança (GUO, C. et al., 2017).

Existem diferentes abordagens para o controlador MPC nas funções LKAS, que se focam na solução de problemáticas como compensar a perturbação de uma entrada externa como a curvatura da faixa, o que permite melhorar o desempenho do deslocamento lateral (KIM, D. J. et al., 2018). Também tem sido utilizado para proporcionar conforto do condutor já que permite que o sinal de entrada da direção do motorista foi transformado e otimizado pelo controlador MPC e o torque da direção do controlador não será percebido pelo motorista. Este método não só melhora a segurança e o conforto da condução do veículo, como também torna a tarefa de condução mais confortável, permitindo oferecer uma ampla solução para alcançar uma cooperação suave, segura e eficiente em futuros sistemas de transporte rodoviário (TIAN et al., 2020). O comando de direção do motorista é corrigido pelo sistema de automação somente quando o veículo está fora das restrições de segurança definida pelos limites de controle do veículo e limitações espaciais formadas por limites de faixa e obstáculos (LIANG, Y.; YIN; NIE, 2021).

No funcionamento da função de custo são utilizados modelos do veículo que permitem olhar N passos na frente, fazendo uma predição do que ira acontecer. Assim, o processo de otimização encontra o melhor ângulo ou torque que seja necessário na direção das rodas do veículo para permanecer no centro da faixa. Esse processo é repetido a cada tempo de mostragem e é aplicada só o valor do primeiro ângulo encontrado na sequência de ações feitas na predição. esses modelos podem ser categorizados como modelos lineares invariantes no tempo (LTI). Os modelos LTI só atualizariam o modelo de previsão em cada tempo de amostragem. Portanto, deve ser capaz de gerar uma trajetória mais precisa e melhorar o desempenho e a robustez do LKAS (CHEN, B.-C.; LUAN; LEE, K., 2014).

O MPC procura uma solução ótima acumulando as saídas previstas das equações de espaço de estado. Portanto, a complexidade computacional é muito maior do que os controladores não preditivos, além os tamanhos das matrizes utilizadas aumenta com o tamanho do horizonte de predição, o que faz com que sejam utilizados mais recursos computacionais. A fim superar estas limitações, diferentes métodos de simplificação são sugeridos para reduzir o tamanho das matrizes utilizadas no processo de otimização do MPC (ALAMIR, 2013), (FARONI et al., 2017), (LI, S. et al., 2014), também tem sido apresentado que se o tempo de amostragem de discretização for aumentado para reduzir a complexidade computacional, a precisão do modelo poderá ser reduzida (YANG, J.; KIM, S.; HUH, 2020).

Atualmente, os métodos usados para simulação e validação de controladores e modelos para sistemas ADAS abrangem vários softwares e plataformas. Esses métodos permitem avaliar e validar o desempenho do LKAS em vários cenários de forma controlada e reproduzível, sem os riscos associados aos testes em estradas.

Nesses cenários, podem ser criadas situações que levam os controladores projetados aos seus limites e permitem aprimorar e calibrar os parâmetros necessários antes de testar em cenários reais (CHEN, Z.; LI, L.; HUANG, X., 2018). Os cenários em que os controladores são simulados correspondentes a curvas ou na primeira etapa para o projeto de controladores, é essencial usar um software para a aplicação de simulações que mostrem o comportamento do veículo; na literatura, há diferentes softwares para essa finalidade; dos mais conhecidos é o MATLAB/Simulink que é um ambiente de programação que permite a modelagem e a simulação de sistemas dinâmicos (MATWORKS, 2023). Ele é amplamente utilizado em pesquisa e desenvolvimento de sistemas de controle por sua facilidade de análise de dados e sua capacidade de abordar problemas complexos a partir da programação. Por esse motivo, os sistemas de assistência ao condutor, como o LKAS, foram desenvolvidos e simulados com essas ferramentas.

Também é utilizado o CarSim e TruckSim que são ferramentas de simulação para a dinâmica de veículos. O CarSim é aplicado na simulação da dinâmica do veículo e permite a análise de manobras e comportamentos específicos do veículo. Já o TruckSim também permite a análise desses cenários, mas se concentra em veículos comerciais, como caminhões e veículos pesados. Nos últimos anos, as simulações foram ampliadas para incluir terrenos complicados e outros "atores", como veículos de trânsito, pedestres, sinais e semáforos e sensores incorporados necessários para cenários de simulação de ADAS e veículos autônomos (CARSIM, 2023), (DAI; KOUTSOUKOS, 2020). Outro software que permite a integração como MATLAB/Simulink é o CarMaker que é uma ferramenta de simulação de veículos desenvolvida pela IPG Automotive. Ele permite a simulação conjunta de modelos de veículos e sistemas de controle implementados no Simulink (CARMAKER, 2023), (PFEFFER, P. et al., 2018).

Outros pesquisadores também utilizaram CARLA que é um simulador de código aberto projetado especificamente para pesquisa de direção autônoma. É útil para sistemas de assistência ao condutor, como o LKAS. Ele oferece um ambiente de simulação 3D realista com a capacidade de criar cenários personalizados e avaliar algoritmos de controle em um ambiente virtual, configurando sensores e condições climáticas (DOSOVITSKIY et al., 2017), (YAKSHA et al., 2021).

Entre os simuladores mencionados, o MATLAB/Simulink é um dos mais utilizados na literatura devido à sua versatilidade e ao desenvolvimento de sistemas de controle. No entanto, o CARLA também tem ganhado uma grande popularidade na pesquisa de direção autônoma devido ao seu foco na simulação realista de veículos autônomos e sistemas de assistência ao condutor. Há também trabalhos que focaram suas pesquisas na aplicação do LKAS em plataforma de tempo real. Uma das plataformas mais utilizadas em pesquisas é a MicroAutobox da dSPACE, essa plataforma é utilizada para testar em cenários reais os controladores ou algoritmos desenvolvidos na parte de simulação (DSPACE, 2023), (KANG;

LEE, S.-H.; CHUNG, 2014). Há também plataformas de software embarcadas, como a ARM (KUO; LU, Y.; YANG, S., 2019), usadas na aplicação de testes reais; em geral, todos os testes realizados nesses controladores são feitos em ambientes controlados com características e situações de curva bem conhecidas. Com base nesta análise é possível se chegar a seguinte descoberta:

Descoberta: *na função LKAS, o modelo matemático torna-se uma ferramenta essencial para representar o comportamento do veículo e desenvolver controladores adequados. Entre os modelos mais utilizados para representar a dinâmica lateral está o modelo de bicicleta, que simplifica a representação do veículo com menos graus de liberdade em relação ao sistema real. Entre os controladores aplicados no LKAS estão o PID, o controle Fuzzy, o H infinito, o LQR e o MPC, sendo que o último apresenta características superiores aos demais controladores. A escolha de um controlador adequado depende do equilíbrio entre a complexidade do modelo, as limitações computacionais e a fidelidade da representação do comportamento real do veículo.*

3.3.4 Tema 4: Benefícios e limitações das LKAS

Os sistemas avançados de assistência ao condutor (ADAS) têm sido bem recebidos entre os sistemas atuais de assistência e segurança no veículo, devido ao auxílio que proporcionam estas funções na mitigação dos erros humanos, que representam anualmente mais de 90% dos acidentes na estrada (KITAJIMA et al., 2022), seja por fadiga, distrações ou alta carga laboral. Entre essas funções de assistência está o LKAS, que pode distinguir entre saídas de faixa intencionais ou não intencionais (SON; LEE, S.-H.; CHUNG, 2013) por meio de informações da posição do freio, sinais direcionais, volante ou sensores. Em um mercado em constante crescimento, onde há cada vez mais proprietários de automóveis, a possibilidade de enfrentar situações de risco aumenta, o que, por sua vez, aumenta a probabilidade de acidentes de trânsito fatais ou lesões graves, já que dirigir pode ser considerado uma atividade perigosa (SENTOUH et al., 2018).

Com o objetivo principal de auxiliar os condutores, o LKAS permite que os usuários melhorem sua capacidade de detectar situações de risco representadas pela saída da faixa (LIANG, J. et al., 2021), alertando-os sobre erros e diminuindo sua carga de trabalho. Um dos desafios da tarefa de dirigir para os seres humanos é a intermitência do rastreamento de objetivos, influenciada pelo sistema sensório-motor e pela necessidade de reduzir os custos computacionais do nosso cérebro (CHEN, J. et al., 2021). Nesse contexto, sistemas como o LKAS são úteis para fornecer rastreamento constante de referências externas (centro da faixa), o que ajuda a manter o veículo em uma zona de segurança (REDDY et al., 2020) o mais próximo no meio da faixa mediante a correção do ângulo das rodas dianteiras.

Ao aprimorar a experiência do condutor, LKAS traz consigo vários benefícios adicionais. Esses benefícios incluem a redução de acidentes de trânsito, a redução de congestionamentos (YAN, M. et al., 2021) e a possibilidade de prestar assistência a idosos e pessoas com

deficiência (JIANG, J.; ASTOLFI, 2018). Além disso, a função LKAS pode contribuir para a redução do consumo de energia e custos em viagens longas (FLORES; LU, X.-Y., 2020), o que é uma grande vantagem para os usuários finais. Também as funções LKAS são amplamente adotadas na indústria automotiva para melhorar a segurança, maior conforto e conveniência para motoristas. Os veículos rodoviários equipados com LKAS são capazes de realizar um desempenho de direção automática parcial, aliviar a fadiga do motorista e evitar acidentes rodoviários (GE et al., 2021).

Outro benefício que pode ser destacado na função LKAS não é apenas o suporte na saída involuntária, mas também na mudança de faixa, pois alertará o usuário sobre o uso inadequado dos sinais de mudança de direção, evitando colisões laterais e traseiras (YANG, J.; KIM, S.; HUH, 2020). Além disso, também está sendo estudada atualmente a adaptação ao perfil do usuário, já que nem todos os usuários quando dirigem o fazem no centro da faixa, o que permitirá que os usuários se adaptem a esse tipo de tecnologia (SENTOUH et al., 2018).

O uso dessa função está atraindo cada vez mais atenção devido aos seus possíveis benefícios, mas atualmente enfrenta vários obstáculos. Um dos principais desafios da função LKAS é que sua operação é altamente dependente das condições da estrada, como a qualidade do pavimento ou as marcações de linha, o que faz difícil a tarefa do reconhecimento dos parâmetros como curvatura e deslocamento lateral (WANG, H. et al., 2019). Esses são fatores importantes, e garantir que a grande maioria das estradas esteja em condições ideais representa uma barreira considerável. Além disso, a função LKAS também deve garantir a segurança e o conforto sob condições de altos ventos, hidroplanagem, mudança de massa do veículo, mudança de coeficiente de fricção da estrada ou mudança no centro de gravidade (LEE, K.; LI, S. E.; KUM, 2018).

Em várias situações, o sistema pode emitir alerta sobre possíveis saídas da faixa (XU, L. H.; HU, S. G.; LUO et al., 2015) quando, na realidade, não existem tais ameaças. Isso, somado com a falta de adaptação do usuário ao seu modo de operação, é comum encontrar usuários com perfis de preferência na condução que nem sempre conduzem no meio da faixa (CHEN, Z.; LI, L.; HUANG, X., 2018), o que gera dificuldades para que condutores mais experientes se acostumem (WEI; PFEFFER, P. E.; EDELMANN, 2023). A insatisfação do usuário também ocorre quando o sistema toma decisões exercendo torque no volante. Isso faz com que os usuários se sintam em conflito com o sistema de controle de direção, forçando-os a exercer grande força se não concordarem com a decisão do sistema (LIANG, Y.; YIN; NIE, 2021). Aliado a isso, existe a possibilidade final de que o usuário desligue essa função.

Por outro lado, as condições climáticas, a hora do dia, a aderência dos pneus ou a presença de obstáculos podem afetar o funcionamento do LKAS (DIAB; AMMAR; SHALABY, 2020). Essas condições podem causar falhas porque o principal sensor usado para obter

informações é uma câmera, que é usada devido ao seu custo relativamente baixo (KANG; LEE, S.-H.; CHUNG, 2014). A câmera está geralmente localizada atrás do para-brisa do veículo ou na frente do espelho retrovisor. No entanto, a câmera pode ser afetada pela luz solar direta, por obstruções causadas por sujeira, chuva, neve ou sombras, o que afeta diretamente o reconhecimento da linha da faixa em que o veículo está localizado (REDDY et al., 2020). Esses fatores podem interferir no processo de computação necessário para manter o veículo no centro da faixa.

Devido à grande quantidade de informação que é captada e o tempo de processamento necessário para processá-las, é necessário utilizar um algoritmo de processamento em tempo real que necessita de grande quantidade de recursos computacionais. Além disso, as estratégias de controle atuais exigem alto desempenho computacional (KANG; LEE, S.-H.; CHUNG, 2016). Também o LKAS exigem redundância e segurança (HUANG, J.; TAN, 2015), o que implica a incorporação de sensores, câmeras e outros elementos, resultando em um aumento significativo no valor final do produto (NGUYEN, A.; CHEVREL; CLAVEAU, 2017). Somando a isso, esses sistemas tendem a estar presentes em veículos de maior valor comercial, o que significa que os consumidores finais constituem uma pequena proporção do número total de veículos em circulação. Isso pode limitar seu impacto na redução de acidentes. Outra consideração importante é que esses sistemas exigem manutenção e reparos especializados, o que pode resultar em custos adicionais para os usuários finais (LITMAN, 2017). Com base nessa análise, é viável alcançar a seguinte descoberta

Descoberta: *O LKAS é capaz de diferenciar entre desvios intencionais e não intencionais e se destaca por seu potencial de evitar colisões. Ao melhorar a experiência do condutor, o LKAS não apenas reduz acidentes e congestionamentos, mas também ajuda pessoas com mobilidade limitada. No entanto, também enfrenta obstáculos e barreiras em termos de infraestrutura, condições climáticas, adaptação do usuário e preocupações com acidentes anteriores. Além disso, o aumento do valor do veículo e os custos de manutenção especializada representam desafios adicionais.*

Na tabela 2 são destacadas as principais descobertas para cada grupamento das informações obtidas nos artigos, juntamente com uma breve descrição do número de artigos, bases de dados e software utilizados na realização do estado da arte para a função LKAS.

Tabela 2 – Estado do Arte

Estado do Arte	
Planejamento	<p>Perguntas de pesquisa</p> <ul style="list-style-type: none"> • O que é o Lane Keeping Assist System (LKAS)? • Quais são os componentes presentes no Lane Keeping Assist System (LKAS)? • O que é utilizado no processo de simulação do LKAS? • Quais são os benefícios e as limitações do LKAS? <p>Base de dados utilizadas IEEE Explorer - Scopus - Web of Science Totais artigos 923 Total artigos escolhidos 69</p> <p>Software utilizado para análise de dados Nvivo</p>
Coleta de dados	
Analise de dados	<p>Definição e funcionamento do LKAS.</p> <p>O Sistema de Assistência à Manutenção na Faixa de Rodagem (LKAS) desempenha um papel importante na segurança e no conforto do usuário, mantendo os veículos em sua faixa. Em geral, ele utiliza para determinar sua posição dentro da faixa e realizando cálculos para ajustar o torque da direção. Os primeiros sistemas LKAS possuíam um mecanismo de atuação baseado em frenagem diferencial, ao passo que os atuais aplicam torque na coluna de direção e emitem alerta ao condutor para enfrentar os desafios das condições reais da estrada devido a distúrbios e à variabilidade dos parâmetros do veículo.</p> <p>Modelagem, simulação e estratégias de controle do LKAS</p> <p>Na função LKAS, o modelo matemático torna-se uma ferramenta essencial para representar o comportamento do veículo e desenvolver controladores adequados. Entre os modelos mais utilizados para representar a dinâmica lateral está o modelo de bicicleta, que simplifica a representação do veículo com menos graus de liberdade em relação ao sistema real. Entre os controladores aplicados no LKAS estão o PID, o controle Fuzzy, o H infinito, o LQR e o MPC, sendo que o último apresenta características superiores aos demais controladores.</p> <p>A escolha de um controlador adequado depende do equilíbrio entre a complexidade do modelo, as limitações computacionais e a fidelidade da representação do comportamento real do veículo.</p> <p>Sensoriamento para a função LKAS</p> <p>Diversos sensores podem ser aplicados na função LKAS, sendo a câmera o principal sensor usado devido à sua implementação econômica. Esse componente desempenha um papel fundamental na captura de informações da cena de trânsito, permitindo a determinação de características como posição, ângulo e geometria da faixa. A partir desses dados, algoritmos são empregados no processamento e na extração das informações e, em muitas aplicações, é necessário usar filtros para estimativa e remoção de ruídos para manter o veículo seguro e alinhado na faixa</p> <p>Benefícios e limitações do LKAS</p> <p>O LKAS melhorou significativamente a segurança nas estradas. O LKAS é capaz de diferenciar entre desvios intencionais e não intencionais e se destaca por seu potencial de evitar colisões. Ao melhorar a experiência do condutor, o LKAS não apenas reduz acidentes e congestionamentos, mas também ajuda pessoas com mobilidade limitada. No entanto, ele enfrenta obstáculos e barreiras em termos de infraestrutura, condições climáticas, adaptação do usuário e preocupações com acidentes anteriores. Além disso, o aumento do valor do veículo e os custos de manutenção especializada representam desafios adicionais.</p>

4 Estudos de casos

Este capítulo apresenta os resultados da formulação de controle MPC e suas variantes considerando as restrições impostas na aplicação da função LKAS. Embora o MPC tenha vantagens no gerenciamento de restrições, requer recursos computacionais elevados. A pesquisa focou em formulação de redução de custos, incluindo a viabilidade em hardware real. Além do MPC, implementou-se o controlador LQR, mais eficiente computacionalmente. Formulação computacionais foram desenvolvidas considerando a formulação matemática do controlador e métodos de otimização. Além disso, foram desenvolvidos cenário de testes para os modelos, com simulações finais em MIL e HIL.

4.1 Modelos de rastreamento de trajetória em *Model-In-the-Loop*

Este estudo de caso teve como principal objetivo de analisar e avaliar o comportamento do controlador MPC no controle da direção na função LKAS com os modelos de rastreamento da referência. Foram projetados e executados cenários utilizando o Matlab instalado num computador com processador Intel Core i7-9700 3,0 GHz e um 14 GB de memória RAM, parâmetros do controlador MPC e formulação de redução são apresentados na tabela 3 .

Tabela 3 – Parâmetros do controlador MPC em MIL

Descrição	Parâmetro	Valor	Unidade
Controlador MPC	N	10	-
	u^{max}	25	deg
	u^{min}	-25	deg
	y_c^{max}	$-(D_r/2) + (D_v/2)$	m
	y_c^{min}	$(D_r/2) - (D_v/2)$	m
	Δu^{max}	5	deg/s
	Δu^{min}	5	deg/s

4.1.1 Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC clássico.

Para a análise de um controlador MPC aplicado na função LKAS (seguimento de referência), é essencial considerar uma série de condições de contorno. Em geral, a manobra utilizada neste contexto geralmente implica em manter o veículo dentro da faixa, ajustando continuamente a direção para seguir uma trajetória de referência. Esse enfoque pode envolver

o controle da orientação do veículo (ângulo de direção das rodas) com base nos sinais de referência fornecidos por um sistema de navegação GPS ou sensores.

É importante levar em conta diversos limites operacionais durante o projeto do controlador MPC. Isso inclui limites na físicos do veículo, assim como limites na taxa de mudança do ângulo da direção, controlando a taxa de variação da velocidade para evitar comportamentos bruscos ou inseguros. Esses limites podem variar dependendo das condições específicas da estrada, para o estudo de caso atual, foram levadas em consideração as limitações físicas do veículo, as condições de conforto para o condutor e a segurança com relação ao máximo deslocamento lateral permitido. Os limites máximos de deslocamento lateral são calculados a partir da largura geral da faixa e do centro calculado a partir das informações obtidas pela câmera.

Rastreando a referência para o controlador MPC e comparando dois modelos apresentados na Figura 9, para o modelo mostrado na parte esquerda da Figura, o controlador MPC permitiu que o deslocamento lateral (linha azul) siga a referência (linha verde) satisfatoriamente, com um perfil de comando ondulante é observado ao longo do experimento. Além de uma grande variação na velocidade de mudança do ângulo do volante para este modelo, o que pode impactar negativamente na vida útil do atuador e no conforto dos usuários ao utilizar a função LKAS.

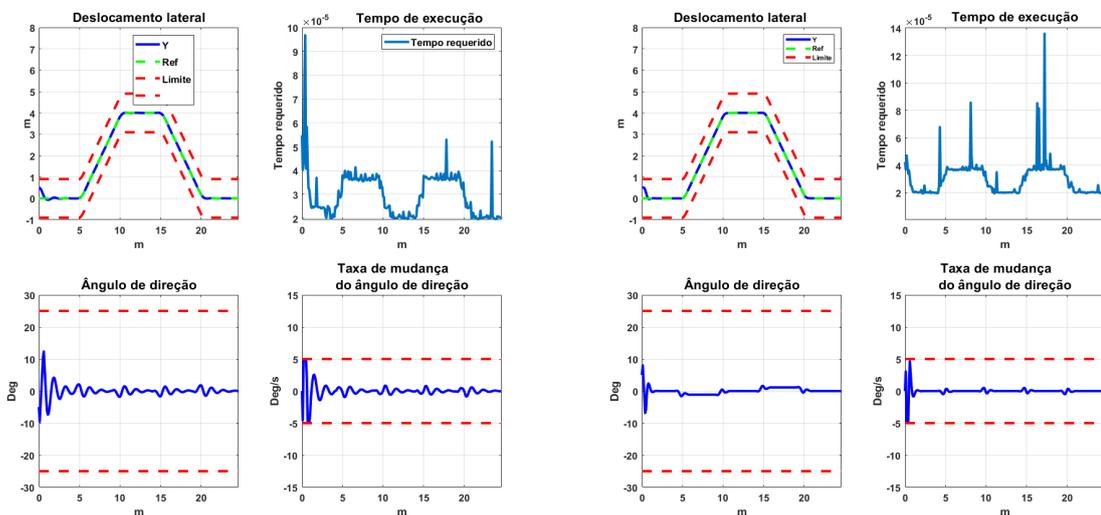


Figura 9 – Desempenho do controlador MPC com QPSWIFT direita modelo 1 , esquerda modelo 2

Para o modelo 1 da figura 9 (a direita), o deslocamento lateral (linha azul) sempre atinge a referência (linha verde pontilhada) e permanece mais constante ao longo da manobra em relação ao modelo 2. É importante destacar também que o controlador consegue se manter dentro das restrições (linhas vermelhas), como o deslocamento lateral máximo permitido estabelecido para ambos os modelos. Porém, nos instantes de tempo iniciais, a taxa de variação do ângulo das rodas dianteiras atinge o valor máximo permitido devido

à condição inicial do veículo (50 cm do centro da faixa). Em geral, o controlador permite manter os dois modelos dentro das restrições e de forma estável, porém o modelo 2 exibe um comportamento mais oscilante quando sujeito aos mesmos parâmetros, o que pode resultar em uma vida útil reduzida do atuador. Por essa razão, os resultados subsequentes neste capítulo para os modelos de rastreamento de trajetória se concentram no modelo 1. Este modelo, provavelmente, oferece uma representação mais estável e previsível do sistema, minimizando o risco de danos ao atuador e garantindo uma operação mais confiável a longo prazo.

A Figura 10 apresenta o comportamento do controlador LQR para comparar seu desempenho com o controlador MPC. Para realizar este teste foi utilizado o modelo 1 considerando as mesmas condições utilizadas no caso anterior. Pelos resultados obtidos após a simulação do controlador LQR, pode-se observar que o controlador foi capaz de rastrear a referência ao longo do tempo de simulação com um tempo médio de cálculo de $4,66\mu s$. Isto demonstra que este controlador é um bom candidato para execução em tempo real em uma ECU. Porém, como este controlador não possui restrições para o mínimo e máximo das variáveis de estado e saída do controlador, o controlador atingiu um valor de ângulo do volante de $[-180$ a 120 graus]. O que em um sistema real não é possível, pois existem limites físicos para este ângulo $[-25$ $25]$ por esta razão o comando encontrado por este controlador não é fisicamente realizável. Por esta razão o comando encontrado por este controlador não é fisicamente realizável. Além disso, a velocidade de mudança do ângulo do volante atingiu um valor máximo de 150 graus por segundo, ultrapassando mais uma vez os limites físicos do sistema, cabe ressaltar que quando o controlador o envia o comando com esses valores ao sistema de direção, o comando só consegue atingir o máximo e fica saturado no ângulo máximo, o que pode reduzir a vida útil do atuador e não permitir a estabilidade do sistema.

Os solucionadores quadráticos QPs são essenciais para garantir um desempenho eficiente e preciso do sistema de controle. Estes algoritmos são a espinha dorsal do MPC, já que se encarregam de re-solucionar os problemas de otimização associados com a tomada de decisões em tempo real. Os QPs permitem gerir eficazmente os problemas de otimização dos sistemas dinâmicos e variáveis ao longo do tempo. A escolha adequada destes algoritmos pode influenciar significativamente no tempo de execução, a estabilidade do sistema e capacidade de lidar com restrições e perturbações no processo controlado.

Estes algoritmos abordam a minimização de uma função quadrática sujeita a restrições lineares, otimizando a tomada de decisões em tempo real. Eles utilizam métodos iterativos para ajustar as variáveis de controle de modo a minimizar a função objetivo, respeitando as restrições estabelecidas. Esta abordagem é iterativa e com capacidade de lidar com restrições não lineares os tornam essenciais para a implementação eficiente de sistemas de controle avançado que utilizam o controlador MPC.

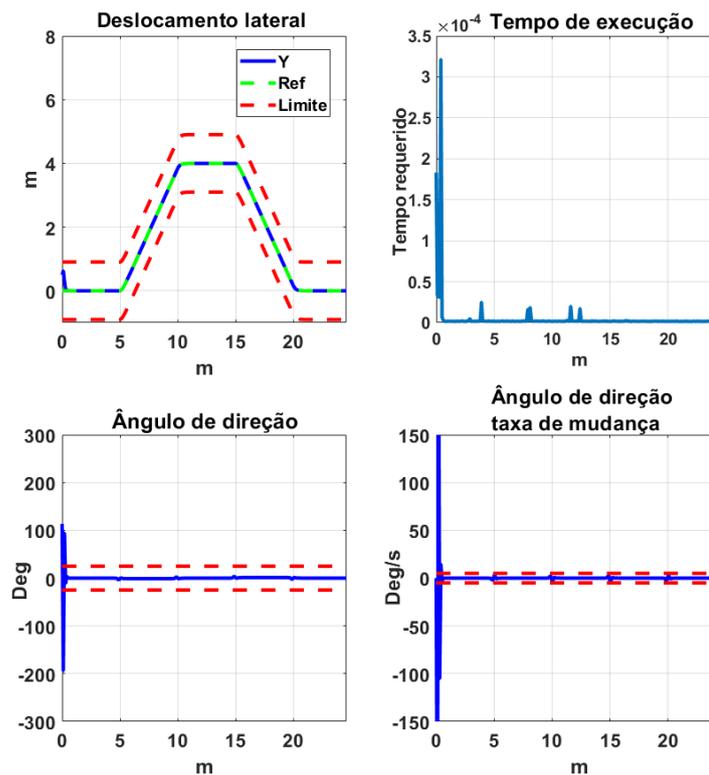


Figura 10 – Desempenho do controlador LQR com modelo 1 de rastreamento da trajetória.

Para avaliar os tempos médios de cada um dos solucionadores QP considerando o controle MPC clássico, uma simulação foi realizada para cada um dos solucionadores QP. A tabela 4 apresenta cada um dos parâmetros configurados. O horizonte de predição N foi configurado em 10 passos para frente, o que gera uma janela de predição de um 1s. Este valor foi configurado após comprovação de que proporciona o melhor equilíbrio entre tempo de cálculo e rastreamento de referência. Já os valores de penalidade, (Q_u) penaliza a resposta do comando e (Q_y) o rastreamento de referência. Outro parâmetro configurado é o número máximo de iterações para cada solucionador QP. Este valor varia dependendo do método e testes anteriores determinaram os valores mostrados.

Tabela 4 – Dados de entrada para os QPs para o MPC clássico.

Parâmetro	QP[1]	QP[2]	QP[3]	QP[4]	QP[5]	QP[6]
Q_y	1E+03	1E+02	1E+03	1E+03	1E+02	1E+03
Q_u	1E+03	1E+03	1E+03	1E+03	1E+02	1E+03
N	10	10	10	10	10	10
Max Iter	50	100	100	60	150	80
Tolerância	1.E-08	1.E-06	1.E-09	1.E-08	1.E-06	1.E-08
Tempo médio (s)	3.E-05	2.E-04	8.E-05	5.E-04	7.E-05	2.E-04

Solucionador: **QP[1]** - QPSWIFT; **QP[2]** - Solucionador conjunto Ativo ; **QP[3]** - QPOASES; **P[4]** - QUADPROG; **QP[5]** - OSQP and **QP[6]** - Fbstab

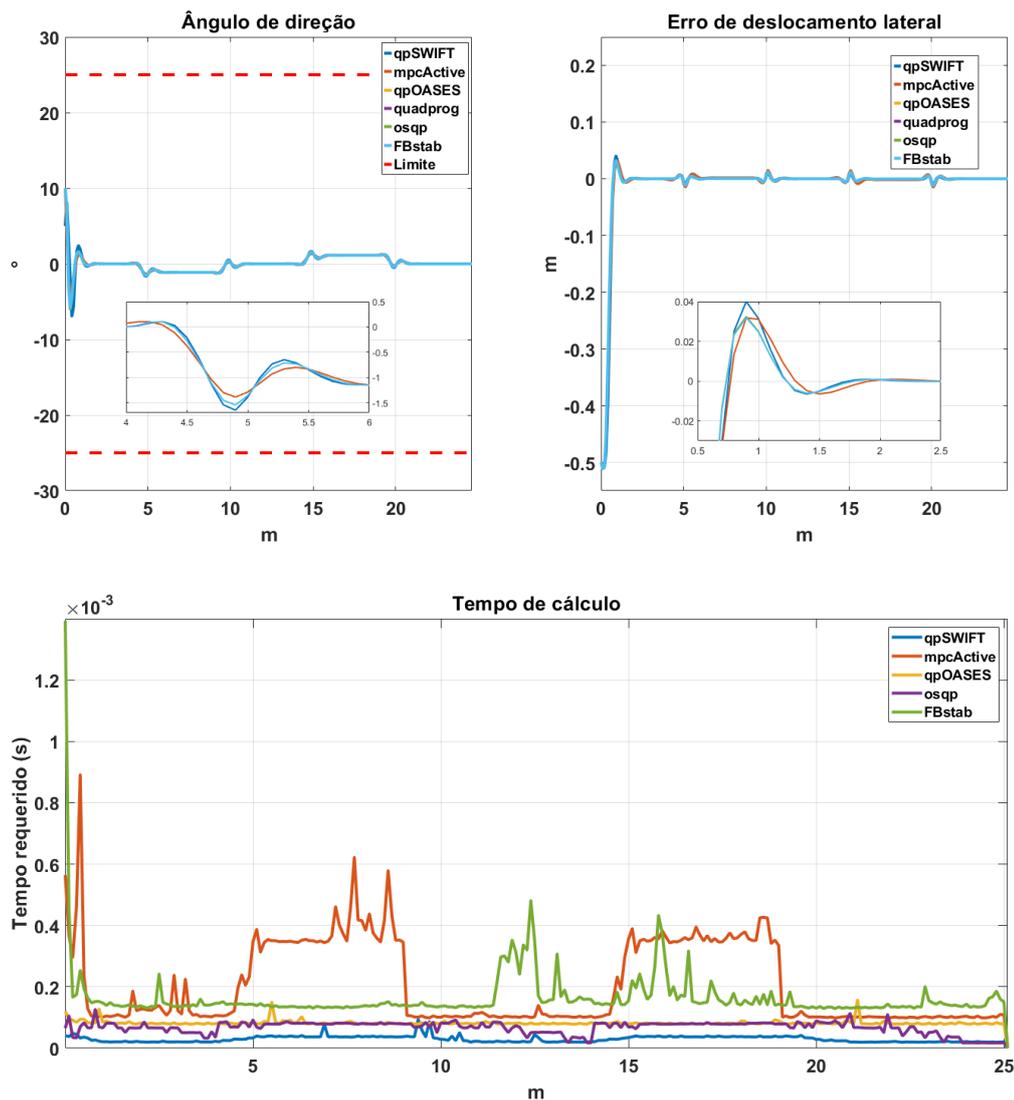


Figura 11 – Resultados do experimento com o controlador MPC clássico a) Erro do ângulo de direção e do deslocamento lateral e b) Tempo computacional dos solucionadores QP.

O perfil de comando, o deslocamento lateral e os tempos de cálculo são mostrados na figura 11. Nesta figura é possível de se observar que todos os solucionadores quadráticos atingiram um nível semelhante de referência. A figura 11, parte inferior que o tempo computacional aumenta quando é necessário um cálculo de ângulo devido a uma mudança constante da referência seguida, para o controlador MPC clássico as melhores médias de tempo computacional foram QPSWIFT com $29.55\mu s$ seguido por OSQP com $(67.14\mu s)$. Os piores resultados computacionais foram obtidos para os solucionadores FBstab e Quadprog com $2.29ms$ e $5.30ms$ respectivamente, devido ao tempo gasto pelo QP.

4.1.2 Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC com parametrização trivial.

Nesta seção, é utilizada a formulação de parametrização trivial para bloquear os comandos de entrada. Essa formulação consiste em interpolar uma sequência de controle e mantê-la constante durante intervalos específicos. Isso tem o efeito de reduzir os graus de liberdade durante o processo de otimização, diminuindo assim o tamanho das matrizes e, conseqüentemente, o tempo computacional necessário para o controlador.

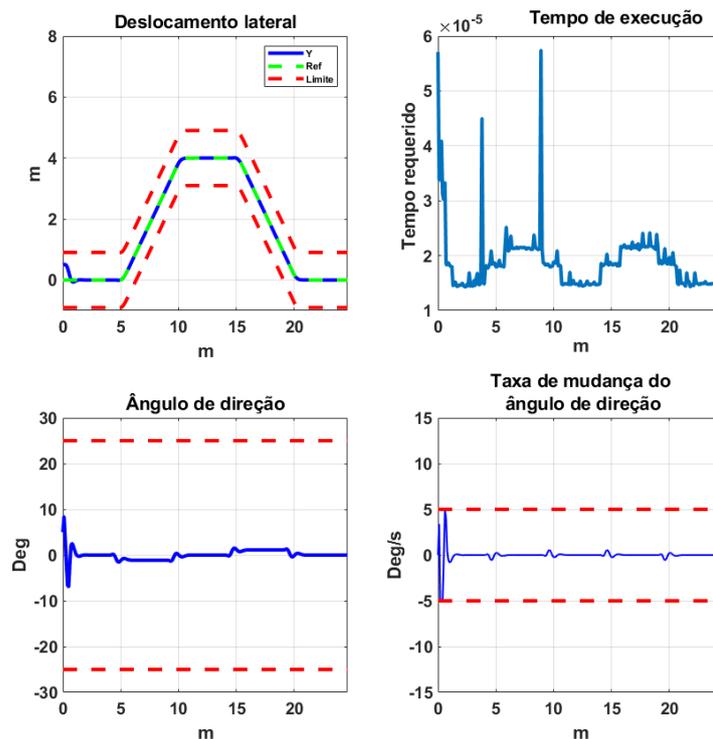


Figura 12 – Resultados de simulação para MPC com parametrização trivial parametrizado, incluindo teste de desempenho de rastreamento de referência

Com base no resultado do controlador MPC, os parâmetros da proposta de parametrização foram configurados para encontrar um perfil de comando semelhante ao do controlador MPC clássico. Isto permitiu utilizar como referência para comparar o comportamento e o desempenho, por isso foi ajustado de forma iterativa até encontrar um perfil de comando semelhante ao da tabela 5 os valores dos parâmetros encontrados para a parametrização trivial podem ser observados.

Os resultados obtidos utilizando o MPC com parametrização trivial é mostrada na figura 12 . Neste cenário é possível observar que o perfil do comando permanece semelhante à resposta do controle MPC clássico (ver figura 11). Enquanto mantém as restrições impostas para o deslocamento lateral, o ângulo e a taxa de variação do ângulo do volante, na parte superior da figura 13 mostra a similaridade de todos os perfis de comando para

Tabela 5 – Parâmetros de calibração de parametrização trivial.

Parâmetro	QP[1]	QP[2]	QP[3]	QP[4]	QP[5]
Qy	1E+03	1E+02	1E+02	1E+02	1E-02
Qu	1E+01	1E+02	1E+02	1E+01	5E-05
DOF	[5;10]	[2;5;9]	[3;5;7]	[5;8]	[2;9]
N	10	10	10	10	10
Max Iter	50	100	100	150	80
Tolerância	1E-08	1E-06	1.E-09	1E-06	1.8E-08
Tempo médio (s)	1.78E-05	1.E-04	6.2-05	5.5E-05	1.8E-04

Solucionador: **QP[1]** - QPSWIFT; **QP[2]** - Conjunto ativo;
QP[3] - QPOASES; **QP[4]** - OSQP and **QP[5]** - Fbstab

todos os solucionadores QPs e, além disso, observa-se que não há desvio lateral significativo, sugerindo que o desempenho é mantido para essa redução proposta no tempo computacional.

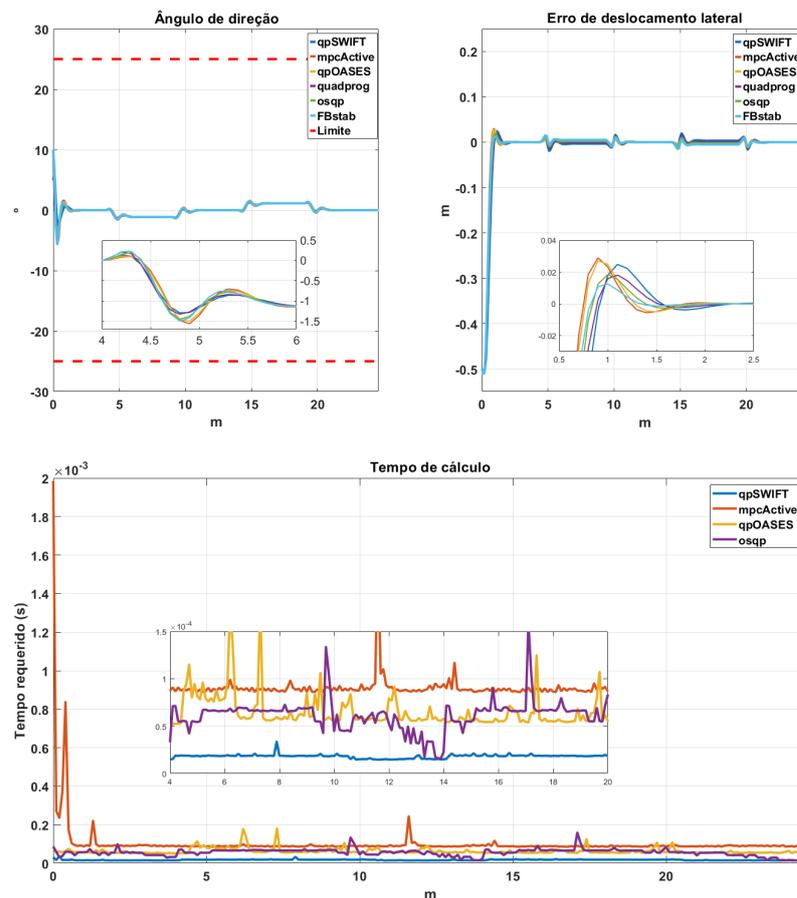


Figura 13 – Parte superior Comando de entrada com erro de deslocamento lateral e parte inferior tempo computacional requerido pelo MPC com parametrização trivial parametrizado.

O tempo computacional necessário para os melhores tempos de execução exigidos pelo MPC com parametrização trivial é apresentado na figura 13. Na figura, não está representado o solucionador FBstab, cujo tempo computacional médio alcançou um valor

significativo de $1.8E - 4s$, Por outro lado, o solucionador conjunto ativo teve uma notória redução no tempo computacional, que foi de $2.E - 4s$ para $1.E - 4s$ sem afetar o comportamento do controlador no rastreamento de referência. Finalmente, o QPSWIT foi o melhor solucionador de QP com um tempo médio de $17.08\mu s$.

4.1.3 Análise de rastreamento de trajetória e do tempo de execução utilizando o controlador MPC com parametrização exponencial.

Para avaliar a segunda formulação de redução de carga, o objetivo foi obter um perfil do comando de controle semelhante ao das duas formulações MPC anteriores. Para a configuração dessa formulação, foi necessário configurar três parâmetros adicionais, foi utilizado o tempo de resposta exigido por nossa planta. (t_r), em seguida, os graus de liberdade da formulação (NoE) e, por fim, o fator alfa são configurados (α) que permite mudar o tempo de resposta e a formulação para a expressão da combinação linear de exponenciais, a configuração de cada solucionador de QP foi realizada de forma iterativa, no final verificou-se que essa formulação permite uma maior suavidade no perfil de comando sem uma perda considerável de desempenho, conforme exposto em (ALAMIR, 2013). Os outros parâmetros foram configurados da mesma forma que as formulações anteriores, os parâmetros utilizados nessa formulação são apresentados na tabela 6.

Tabela 6 – Dados de entrada de QPs o controlador MPC com parametrização exponencial.

Parâmetro	QP[1]	QP[2]	QP[3]	QP[4]	QP[5]
Qy	1E+02	1E+01	1E+02	1E+02	2.E-01
Qu	1E+02	1E+01	1E+02	1E+01	1.E+1
NoE	2	2	2	2	2
α	25	150	15	15	125
t_r	0.01	0.01	0.05	0.05	0.05
N	10	10	10	10	10
Max Iter	50	100	100	150	80
Tolerância	1E-08	1E-06	1.E-09	1E-06	1.2E-04
Tempo médio (s)	1.4E-05	9.1E-05	4.0-05	4.3E-05	1E-08

Solucionador: **QP[1]** - QPSWIFT; **QP[2]** - Conjunto ativo;
QP[3] - QPOASES; **QP[4]** - OSQP e **QP[5]** - Fbstab

Os resultados obtidos da parametrização exponencial são mostrados na figura 14. Se evidencia um comando mais suave do ângulo do volante, o que permite uma menor taxa de variação do ângulo do volante, para este caso de simulação a variável de comando não atinge essas restrições, o que pode resultar em um cálculo mais rápido para encontrar o comando do ângulo da direção, pois foi evidenciado que no momento em que uma variável está atingindo uma restrição imposta o tempo de cálculo aumenta.

Os resultados obtidos para este experimento com todos os solucionador QPs mostraram um perfil de comando semelhante dos experimentos, em que o ângulo de direção e sua

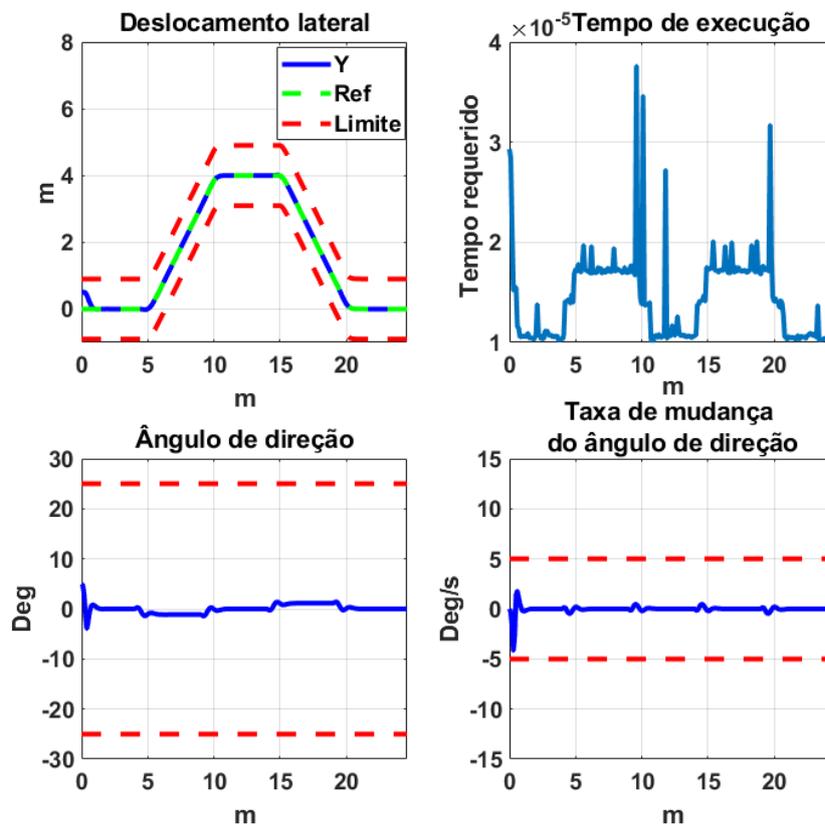


Figura 14 – Resultados de simulação obtidos com o teste de desempenho de rastreamento MPC com parametrização exponencial.

taxa de variação permaneceram dentro das restrições previamente estabelecidas. Além disso, para este último experimento, não foi obtido nenhum erro significativo no rastreamento de referência; para o solucionador FBstab e QPoases, houve um erro próximo a 2,5 mm, o que não representa perigo ou baixo desempenho do controlador. Com os resultados obtidos para esses modelos de rastreamento de referência com a configuração e a escolha adequadas do solucionador QPs, a formulação clássica do MPC pode ser substituída pela parametrização exponencial sem uma diminuição significativa no desempenho do controlador. O comportamento geral dos solucionador QPs quadráticos é apresentado na Figura 15, onde são mostrados o comando do perfil, o erro de deslocamento lateral e o tempo computacional.

Os solucionadores QPs com o controlador MPC com parametrização exponencial apresentaram um menor tempo de execução, sem diferenças significativas no comando de entrada (u) e no rastreamento de referência. Além disso, os picos tempo computacional ocorrem com menos frequência em comparação com as soluções anteriores. O resultado do experimento demonstra uma redução notável no tempo médio de computação em comparação com as formulações mencionadas acima.

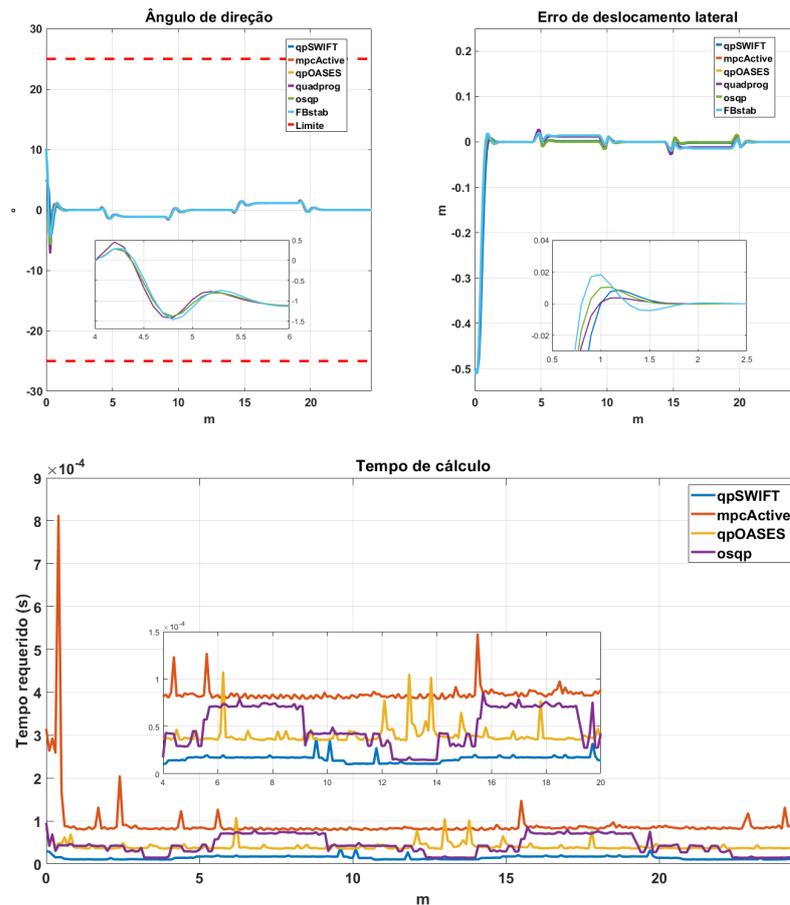


Figura 15 – Comando de entrada e erro de deslocamento lateral, parte superior, tempo computacional exigido pelo MPC com parametrização exponencial, parte inferior.

4.1.4 Análise comparativa das soluções

Foi realizada uma análise comparativa para avaliar as soluções de controladores MPCs baseados nas formulações clássica, trivial e exponencial. Esta análise comparativa se baseia na redução de custo computacional para os controladores durante o ciclo de teste e verificação MIL. Na figura 16 o tempo médio de computação para cada formulação e solucionador de QP é apresentado, e destacado em vermelho está o solucionador QPSWIFT, que obteve o melhor tempo médio de execução para as três formulações. Verificou-se que a formulação exponencial permite o melhor tempo médio para cada solucionador QP e, como resultado, o solucionador QPSWIFT obteve um tempo médio de execução do controlador MPC com formulação exponencial igual a $(14,26\mu s)$.

Na figura 17 apresenta os três melhores solucionadores com cada uma das formulações. Como já mencionado, a formulação MPC com parametrização exponencial utilizando o QPSWIFT obteve o menor tempo computacional médio, com um desempenho de $14,26\mu s$. Isso sugere que essa combinação específica de formulação de controle e solucionador de problemas quadráticos é altamente eficiente em termos de velocidade de execução e rastreamento da trajetória.

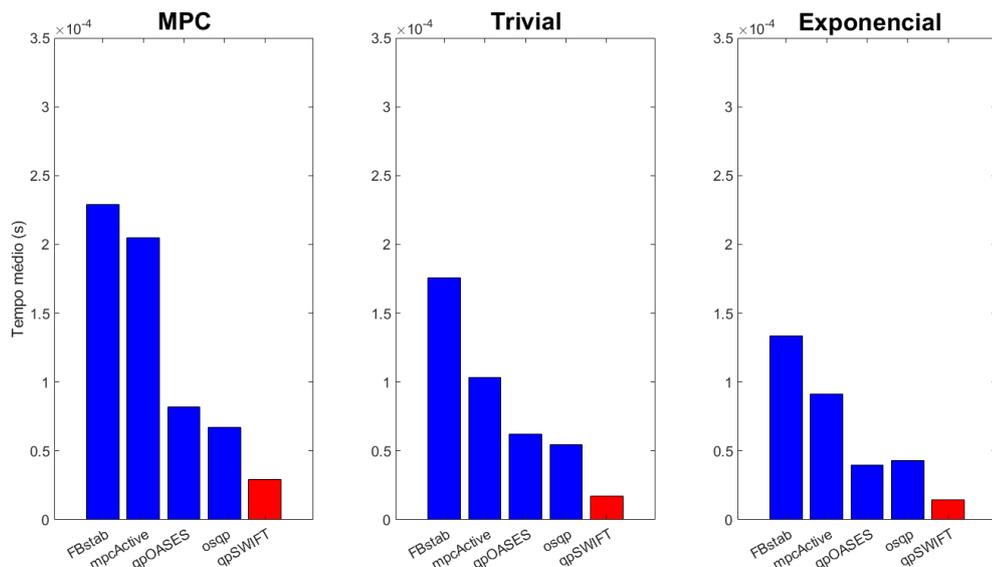


Figura 16 – Comparação do tempo de execução de todos os solucionador QPs quadráticos

Por outro lado, é importante observar a disponibilidade do QPSWIFT para implementação em uma ECU específica não pode ser garantida nessa aplicação para função LKAS utilizando a ECU (Baseline). Limitações técnicas, restrições de hardware ou software podem afetar a escolha do solucionador de problemas quadráticos a ser utilizado em aplicações em tempo real.

É por isso que o estudo também apresenta o QPOASES como a próxima opção mais promissora, com um tempo de execução de $39.90\mu s$, o que ainda é bastante eficiente. O OSQP alcançou a terceira posição em termos de tempo de execução, com $43.24\mu s$. Na tabela 7 mostra a redução percentual do tempo computacional de cada uma das formulações de MPC com cada solução, o que permite comparar a redução percentual entre as três formulações de controladores MPCs estudada.

Tabela 7 – Comparação das diferentes formulações na redução do tempo médio de computação.

Solucionador QP	Clássico vs Trivial	Trivial vs Exponencial	Clássico vs Exponencial
QPSWIFT	42%	17%	52%
Conjunto ativo	49%	12%	56%
QPOASES	24%	36%	51%
OSQP	19%	21%	36%
FBstab	23%	30%	46%

Conforme os dados apresentados na tabela 7, pode-se observar que o solucionador com a maior redução no tempo computacional para a formulação clássica é o que apresenta a maior redução no tempo computacional para a formulação exponencial. Nesse caso de estudo o solucionador de conjunto ativo alcançou um (56%) na porcentagem de redução de tempo desde a formulação clássica para a formulação exponencial. Por outro lado, o

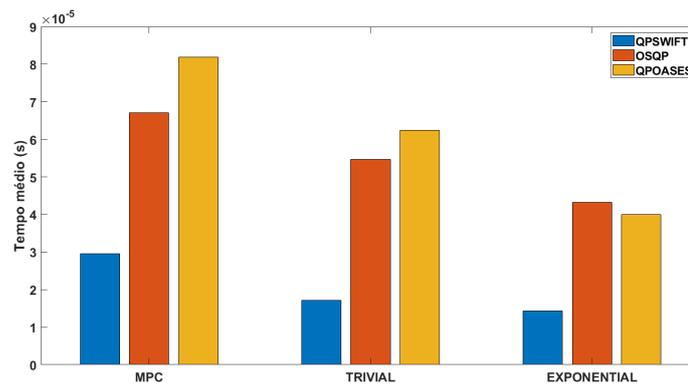


Figura 17 – Maiores alternativas de candidatos a QPs.

QP com a menor redução (19%) foi o OSQP ao mudar do MPC clássico para o MPC com parametrização trivial. Em todos os cenários, foi verificada uma redução significativa no tempo computacional sempre que foram usadas técnicas de parametrização triviais ou exponenciais.

4.2 Análise comparativa do erro de posição lateral e do erro de posição de guinada em *Model-In-the-Loop*

O modelo referente ao erro de posição lateral e ao erro de posição de guinada, apresentado na equação 2.13, é denominado assim devido ao fato de suas variáveis de estado serem representadas pelo erro atual do veículo em relação ao centro da faixa. Esta seção descreverá os resultados obtidos no ciclo de teste, verificação e validação MIL e HIL, enquanto ao ciclo de teste HIL foi desenvolvido na plataforma em tempo real Simulink com a plataforma de simulação em tempo real Speedgoat. O modelo de erro obtido no ciclo de teste MIL e HIL (RAJAMANI, R., 2011) foi implementado por meio da criação de cenários no *driving scenario create* do Matlab. Os parâmetros utilizados para o modelo do veículo são apresentados na tabela ??, esses parâmetros foram utilizados para os três modelos diferentes.

O modelo de erro e sua formulação oferecem uma vantagem sobre os modelos de rastreamento de trajetória, pois permitem incluir uma previsão da perturbação no sistema (curvatura desejada). Além disso, a formulação da matriz ampliada permite inserir a mudança do ângulo como variável de controle no processo de otimização proporcionando um comando mais suave quando comparado aos modelos anteriores.

4.2.1 Controlador MPC com formulação clássico em *Model-In-the-Loop*

Para a simulação do controlador MPC clássico em *Model-In-the-Loop*, inicialmente foram definidas as restrições usadas e para esse caso o tamanho do horizonte de previsão $N=20$ foi aumentado em relação aos modelos anteriores 2.6 e 2.7, que permite uma melhor

rejeição de perturbações (curvatura da pista). Além disso, para esse modelo a restrição de deslocamento lateral foi estabelecido como sendo $20cm$, em geral, os parâmetros para esse controlador são apresentados na tabela 8, em que são apresentados os parâmetros de restrição foram alterados para uma formulação diferente no que respeita à utilização das informações recolhidas na faixa de rodagem.

Tabela 8 – Parâmetros do controlador MPC nos ciclos de testes MIL e HIL para o modelo do erro

Descrição	Parâmetro	Valor	Unidade
Controle MPC-trivial- exponencial	N	20	ms
	u^{max}	23	deg
	u^{min}	-23	deg
	y_c^{max}	0.2	m
	y_c^{min}	0.2	m
	Δu^{max}	11.45	deg/s
	Δu^{min}	11.45	deg/s

A figura 18 mostra os resultados de dois controladores diferentes para a função LKAS, no lado esquerdo um controlador MPC e no lado direito um controlador LQR. Em geral, as quatro variáveis consideradas para a comparação de desempenho dos controladores são o deslocamento lateral, o tempo de computacional em cada iteração, o perfil do ângulo da roda e a taxa de variação do ângulo de roda.

O controlador MPC é capaz de controlar o sistema de direção sem desrespeitar as restrições impostas, isto é, o veículo não deve deixar a faixa de rodagem. Neste experimento, o tempo médio de computação é de $0,403ms$ e tem picos que chegam a $1,4ms$. Isto sugere que o controlador está abaixo da taxa de amostragem do sistema controlado que nesse cenário é igual a $100ms$. Por outro lado, o perfil do ângulo da roda mostra que o controlador adapta o ângulo conforme a curvatura da estrada, evitando movimentos bruscos graças à restrição em sua taxa de variação. Embora a taxa de variação do ângulo seja maior do que a inicial devido à condição inicial de ângulo de guinada igual $0,012467$ rad, quando o veículo se ajusta à trajetória, ele se torna mais suave e estável.

O controlador LQR também consegue manter o veículo dentro da faixa, mas excede os limites de deslocamento lateral, o que pode ser perigoso em situações de trânsito, pois faz com que o veículo se aproxime das margens da faixa. O tempo de cálculo é menor quando comparado ao do controlador MPC, o que pode ser uma vantagem em termos de desempenho, mas também pode indicar menor precisão ou robustez, já que permite que o veículo saia dos limites de deslocamento lateral da zona de segurança estabelecidos.

O perfil do ângulo da roda mostra que o controlador tem um comportamento mais oscilante durante todo o experimento, o que pode causar problemas em curvas fechadas ou mudanças de direção. A taxa de variação do ângulo é maior do que a do controlador MPC, o

que significa que o controlador faz mais ajustes e correções, causando maior desgaste no atuador e levando a uma redução do tempo de vida do atuador.

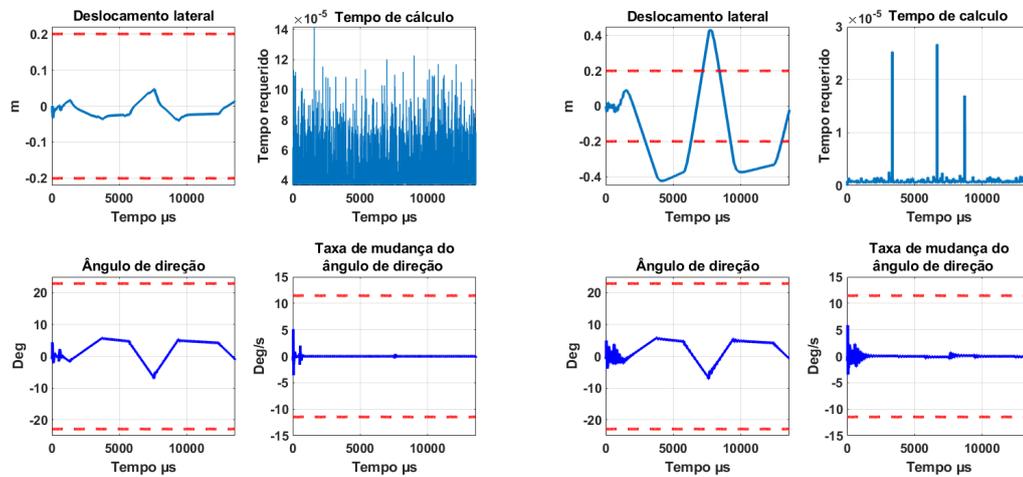


Figura 18 – Desempenho do controlador MPC e Desempenho do controlador LQR para LKAS.

A Figura 19 apresenta os resultados de desempenho de três solucionadores QP para o controlador MPC da função LKAS. Na parte superior da figura 19, o perfil de comando dos três solucionadores QP é mostrado, sendo possível verificar que todos os comandos gerados por esses solucionadores se sobrepõem. Além disso, durante todo o experimento, os comandos de todos os três solucionadores permanecem dentro dos limites definidos pelas restrições. Na segunda parte da figura, parte superior direita, o deslocamento lateral do centro da pista é abordado. Aqui, novamente, observa-se que os três solucionadores QP apresentam resultados consistentes, com os deslocamentos laterais seguindo uma trajetória quase idêntica. Na parte inferior da figura é possível observar que o método do conjunto ativo (MCA) atinge o menor tempo médio de computação, seguido pelo método do ponto interior, enquanto o solucionador FBstab alcança o terceiro melhor desempenho. É importante observar que, por meio de experimentos, foram detectados picos de tempo computacional para o FBstab, chegando a quase 12 milissegundos. Esses resultados destacam a importância de considerar tanto a precisão dos comandos gerados quanto o desempenho computacional ao selecionar um solucionador de QP na função LKAS.

A tabela 9 mostra os parâmetros de penalização e, principalmente, os tempos médios de computação para cada solução. Os resultados acima mostram uma correlação direta com a penalidade de cada variável, já que, nesse caso específico, todos usaram a mesma penalidade.

4.2.2 Controlador MPC com parametrização trivial em *Model-In-the-Loop*

Os resultados mostrados na figura 20 correspondem ao desempenho do controlador MPC com parametrização trivial com o solucionador de conjunto ativo. Observa-se que o

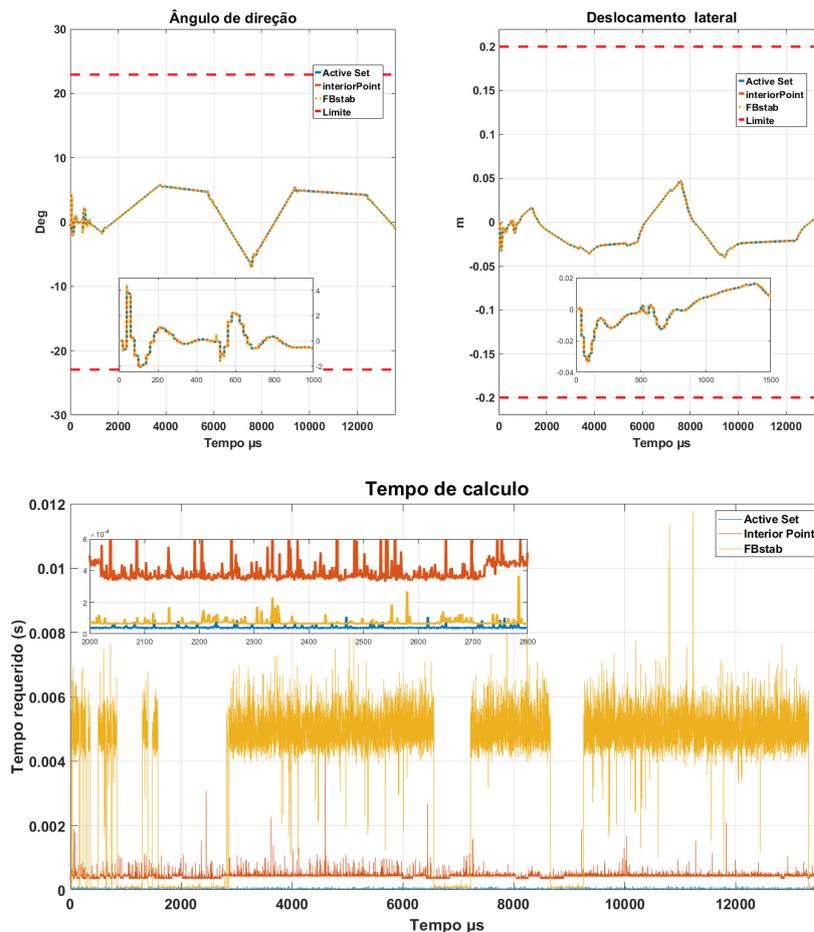


Figura 19 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC clássico

Tabela 9 – Parâmetros para controlador MPC-MIL

Parâmetros	Conjunto ativo QP	Interior Point QP	Fbstab QP
Qy	[1170,390,8000,24200,260]	[1170,390,8000,24200,260]	[1170,390,8000,24200,260]
Qu	140	140	140
N	20	20	20
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	4.03E-05	4.33E-04	3.7.E-03

controlador MPC com parametrização trivial consegue manter o veículo dentro das restrições de deslocamento lateral e ângulo da roda, sem oscilações ou ultrapassagens. Além disso, o deslocamento lateral do veículo com o controlador MPC com parametrização trivial é ligeiramente maior em comparação com o controlador MPC clássico, mas não representa uma diminuição no desempenho do mesmo, pois não atinge as restrições definidas, sugerindo que o controlador MPC com parametrização trivial tem uma margem de segurança adequada.

O perfil de comando do controlador MPC com parametrização trivial é semelhante ao do controlador MPC clássico, sendo mais suave e menos variável. Isso também se reflete na taxa de variação do ângulo da roda, menor com o controlador MPC com parametrização

trivial. Isso implica que o controlador MPC com parametrização trivial gera menos desgaste e consumo de energia do que o controlador MPC clássico.

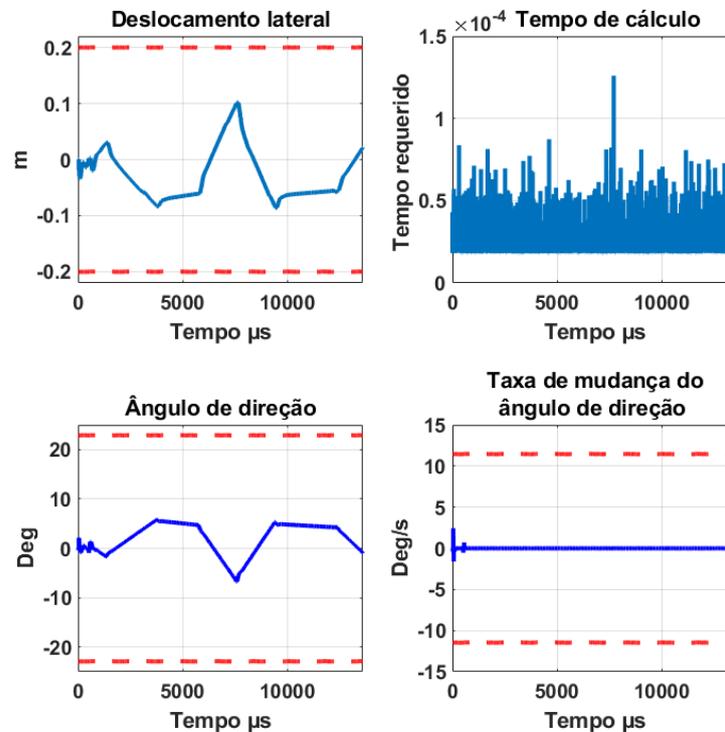


Figura 20 – Desempenho do controlador MPC com parametrização trivial para LKAS.

A figura 21 mostra os resultados obtidos a partir de experimentos com o controlador MPC trivial considerando três solucionadores QPs distintos. Na parte superior da figura 21, é possível observar que os três solucionadores seguem um padrão semelhante no perfil de comando. O método do ponto interior demonstrou uma resposta mais suave. Além disso, todos os três solucionadores respeitaram as restrições definidas para o ângulo da roda, e o valor máximo do ângulo no início do experimento para a parametrização trivial é menor do que o valor máximo do ângulo da roda no início do experimento 2.2 graus em comparação com a formulação clássica do controlador MPC 4.3 graus. Na segunda parte do deslocamento do centro da pista para cada solucionador QP, nota-se que o FBstab atinge o menor deslocamento, enquanto o método de ponto interior atinge um valor mais alto, superior a 10 cm. Além disso, observa-se que os três solucionadores mantêm um intervalo de deslocamento semelhante durante os primeiros 2 segundos do experimento. Na parte inferior da figura 21 são mostrados os tempos de execução do experimento para cada solucionador de QP. É notável que o FBstab apresenta um número maior de picos no tempo computacional.

No entanto, o tempo médio de cálculo mais elevado para essa formulação foi obtido com o solucionador de ponto interior, com $0,4ms$, enquanto o método do conjunto ativo mostra um comportamento mais constante e um valor médio mais baixo, com $20,0ms$.

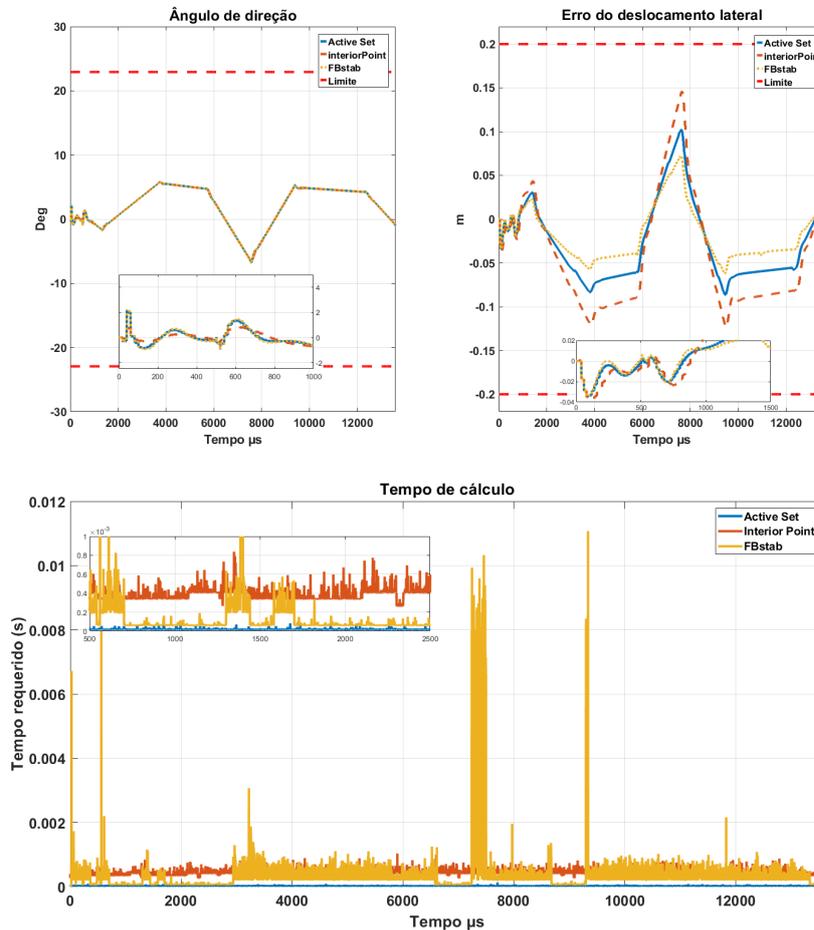


Figura 21 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC com parametrização trivial

Os resultados anteriores obtidos são influenciados por cada penalidade e parâmetro calibrado para cada solucionador quadrático. Para o controlador MPC clássico, os valores de penalidade foram constantes e, para essa formulação de parametrização, os resultados também mudaram para cada QP. Vale observar que os graus de liberdade também mudaram para o QP do ponto interno, com os pontos 1 e 5 como pontos de graus de liberdade, o que representou uma penalidade maior do deslocamento lateral e uma diminuição representativa na penalidade do valor do ângulo do volante e sua taxa de variação, conforme mostrado na tabela 10.

4.2.3 Controlador MPC com parametrização exponencial em *Model-In-the-Loop*

Os resultados obtidos com a implementação da formulação de redução da carga computacional do MPC com parametrização exponencial com o solucionador de conjunto

Tabela 10 – Parâmetros controle MPC com parametrização trivial MIL

Parâmetros	Conjunto ativo QP	Ponto interior QP	Fbstab QP
Qy	[1170,390,8000,24200,260]	[25000,390.51,8014,24253,1E-20]	[2000,500,8000,25000,1E-2]
Qu	1,40E+02	1,00E-45	1,00E-02
DOF	[3;5]	[5]	[3;5]
N	20	20	20
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	2.01E-05	4.1E-04	3.6-04

ativo são apresentados na figura 22. Em primeiro lugar, o deslocamento lateral do veículo segue uma trajetória semelhante à obtida com a formulação trivial, com um valor máximo de $9,2\text{cm}$, o que evidencia que o seguimento da referência não é afetado pela redução da carga computacional dessa formulação. Além disso, observa-se que o perfil de comando do ângulo da roda gerado pelo MPC com parametrização exponencial tem maior suavidade em comparação com a formulação trivial, o que leva a uma melhor experiência de conforto.

Uma vez que a taxa de variação de comando permanece quase constante para a parametrização exponencial e é muito menor em comparação com os dois controladores anteriores no estágio inicial do experimento, com valores de $5,12\text{ deg/s}$, $2,143\text{ deg/s}$ e $0,79\text{ deg/s}$ para MPC com parametrização clássica, trivial e exponencial, respectivamente, o restante do experimento para a última formulação é mantido próximo de zero, evitando mudanças abruptas e oscilações indesejadas no controle do veículo.

Em termos de eficiência computacional, o tempo de execução do solucionador permanece constante durante todo o experimento e tem uma média de 15ms . O tempo médio de computação é significativamente reduzido em comparação com a formulação trivial, indicando uma melhoria substancial na eficiência do processo de controle. Além disso, é importante observar que nenhuma restrição é atingida durante a simulação, o que confirma que o MPC com parametrização exponencial consegue respeitar os limites físicos e operacionais do veículo e do sistema de controle.

Os resultados da comparação entre os três solucionadores QP utilizados na formulação de parametrização exponencial para a redução da carga computacional são mostrados na figura 23. No canto superior esquerdo é possível observar que o perfil de comando dos ângulos da roda para esse caso o método FBstab mostra uma resposta mais ondulada e com um valor mais elevado nos primeiros instantes do experimento. Ao contrário, os métodos conjunto ativo e ponto interior apresentam perfis mais suaves e se interpolam, o que pode ser devido ao uso dos mesmos parâmetros de penalidade, conforme mostrado na tabela 11. Isto porque o FBstab é usada outra penalidade e parâmetros de tempo de assentamento e alfa, porque com a mesma penalidade não consegui encontrar uma simulação com um perfil de comando semelhante aos anteriores.

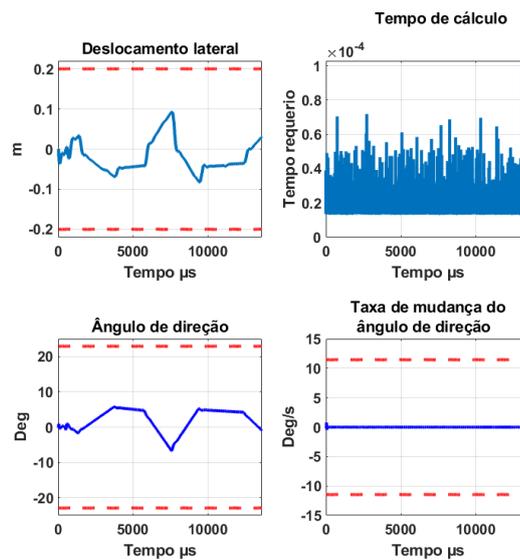


Figura 22 – Desempenho do controlador MPC com parametrização exponencial para LKAS.

Como na parte anterior, o método FBstab mostrou um comportamento mais inicialmente e depois, também atinge valores de deslocamento mais altos em comparação com os outros dois solucionadores, chegando a 10,82 cm. Em geral, o comportamento dos três solucionadores permanece dentro das restrições impostas, mas os métodos do conjunto ativo e do ponto interior oferecem melhor desempenho e menor ondulação.

Na última parte, são mostrados os tempos de execução da formulação de parametrização exponencial, sendo possível verificar que o método FBstab reduz muito os picos de tempo computacional, o que implica maior eficiência e menor demanda computacional. Todos os três solucionadores QP apresentaram uma redução no tempo em comparação com as duas formulações anteriores, mas o método do conjunto ativo com 15μ foi o mais rápido, seguido pelo FBstab com uma média de $25\mu s$ e o método do ponto interior com $39\mu s$.

A tabela 11 mostra os parâmetros utilizados para a formulação de MPC com parametrização. O número de graus de liberdade para essa formulação era, em geral, 2, mas o número de alfa mudou para cada solucionador de QP. Esse parâmetro, combinado com o tempo de estabilização t_r , permitiu a calibração da suavidade e do tempo médio de computação.

Tabela 11 – Parâmetros controle MPC com parametrização exponencial MIL

Parâmetros	Conjunto ativo QP	Ponto interior QP	Fbstab QP
Qy	[2E4,2E3,16E3,5E4,1E5]	[2E4,2E3,16E3,5E4,1E5]	[2E4,15E2,16E3,25E3,5E2]
Qu	1,00E+07	1,00E+07	5,00E+01
NoE	2	2	2
α	25	25	225
t_r	0.01	0.01	0.005
N	20	20	20
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	1.5E-05	3.9E-04	2.5-05

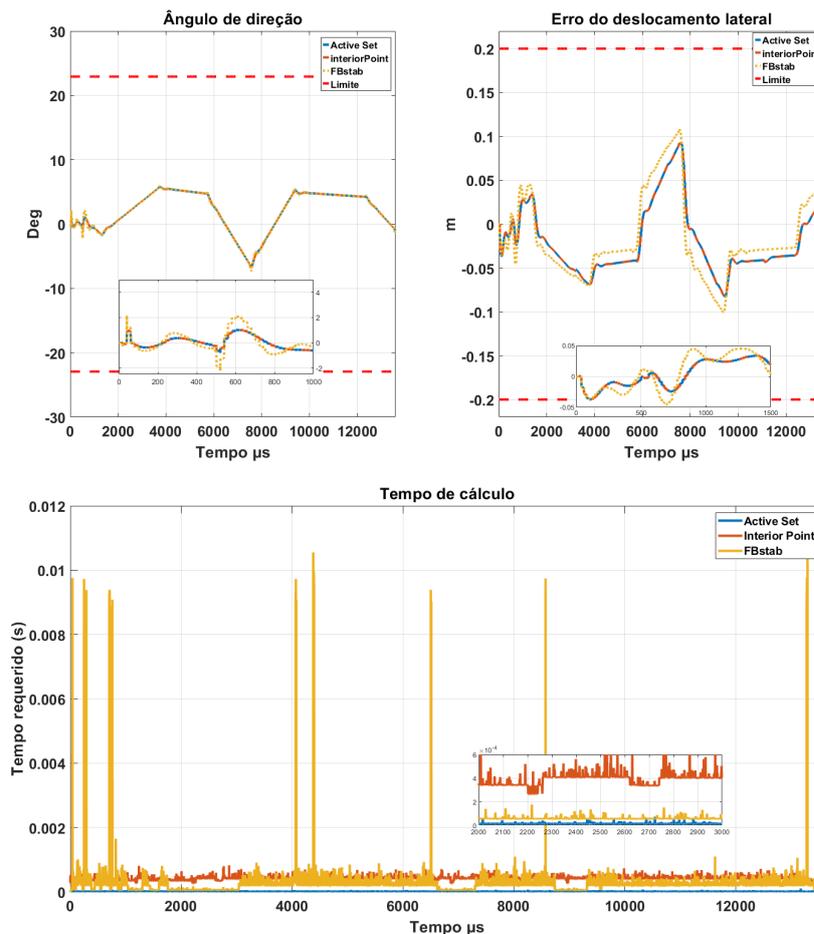


Figura 23 – Comando de entrada e tempo computacional dos solucionadores quadráticos do controlador MPC com parametrização exponencial

4.2.4 Análise comparativa dos controladores MPCs em *Model-In-the-Loop*.

Uma análise comparativa dos controladores MPC clássico, com parametrização trivial e com parametrização exponencial em combinação com três solucionadores QP do método do conjunto ativo, do método do ponto interior e do FBstab é mostrada nesta seção. A combinação revela resultados significativos em termos de tempo computacional dos solucionadores e seu desempenho na reação e no comando do ângulo do volante. Entre as várias combinações avaliadas, é importante destacar que o solucionador que obteve o melhor tempo computacional foi o método do conjunto ativo, quando a formulação de redução de custo de parametrização exponencial é aplicada. Essa abordagem atinge um tempo médio de $15\mu s$, indicando uma notória eficiência computacional. Em segundo lugar, o solucionador FBstab também demonstra um sólido desempenho em termos de tempo computacional, com um tempo médio de $25\mu s$. Essas duas combinações são excelentes em termos de velocidade na solução dos cenários propostos para essa simulação da função LKAS.

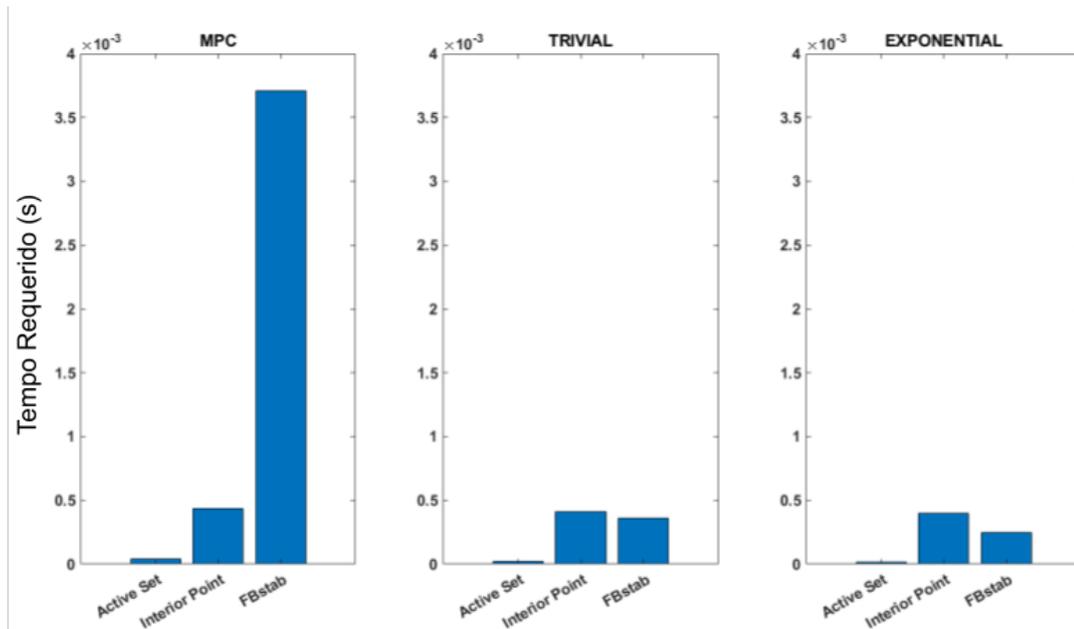


Figura 24 – Comparação de todos os solucionadores QP.

Tabela 12 – Diminuição percentual no tempo médio de cálculo em MIL

Solucionador QP	Clássico vs Trivial	Trivial vs Exponencial	Clássico vs Exponencial
Conjunto ativo QP	50%	26%	63%
Ponto interior QP	5%	3%	8%
Fbstab QP	90%	31%	93%

Por outro lado, o solucionador com o pior tempo médio é o método do ponto interior. Essa abordagem é caracterizada por um desempenho mais lento em comparação com os outros solucionadores avaliados. Esses resultados destacam a importância da escolha do solucionador e da formulação de parametrização sobre a eficiência do controlador MPC, o que pode ter implicações significativas em aplicativos e sistemas em tempo real que exigem respostas rápidas. Na figura 24 é evidenciado de forma gráfica como é a redução do tempo de cálculo para cada formulação de redução e solucionador quadrático, esses 3 gráficos estão na mesma escala de tempo para evidenciar de forma mais clara a redução de tempo.

Na tabela 12 são apresentadas as reduções percentuais de cada solucionador com cada formulação do controlador MPC. Neste experimento, a maior redução de tempo foi para o solucionador fbstab, comparando o MPC clássico com a formulação exponencial, com um valor de 93% seguido pelo método do conjunto ativo, em geral, também é evidente que as piores porcentagens de redução de tempo foram para o método do ponto interior com um valor de 8%. Além disso, vale a pena mencionar que, embora o FBstab tenha obtido a melhor porcentagem de redução de tempo e se tornado um potencial para implementação em tempo real, o solucionador de conjunto ativo obteve a melhor média computacional.

Na figura 25 são apresentados os tempos médios para cada uma das formulações do controlador MPC com o solucionador conjunto ativo apresentado como a melhor opção para

a execução na função LKAS. Além disso, é importante mencionar que nem sempre é encontrada a disponibilidade dos solucionadores quadráticos QP nas linguagens ou plataformas de hardware; por isso, é importante ter diferentes solucionadores quadráticos e formulações que permitam uma maior flexibilidade na execução do controlador MPC em tempo real.

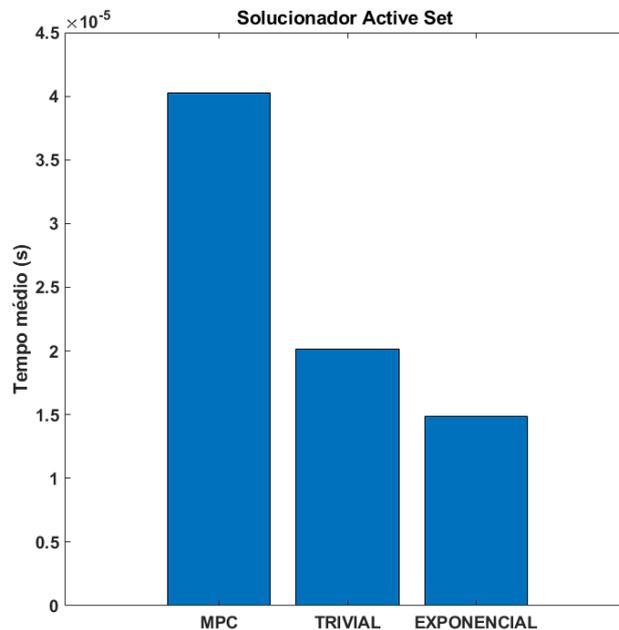


Figura 25 – Melhor tempo encontrado pela combinação do solucionador e da formulação.

4.3 Análise comparativa do erro de posição lateral e do erro de posição de guinada em *Hardware-In-the-Loop*

Nesta seção é apresentado os resultados obtidos para o controlador MPC na função LKAS em *Hardware-In-the-Loop*, para a execução da aplicação em tempo real utilizamos a plataforma Speedgoat com *Simulink-Real time*, foram utilizadas duas máquinas de simulação em tempo real, a primeira máquina (Performance) foi utilizada principalmente para a simulação da dinâmica veicular e seu comportamento na faixa, toda a data dessa simulação são enviadas via Ethernet para a segunda máquina Baseline (ECU). Essas informações são enviadas via Ethernet para a segunda máquina Baseline (ECU), responsável pela execução do controlador em tempo real visando encontrar o valor ideal do ângulo para manter o veículo no meio da pista.

Os solucionadores conjunto ativo e ponto interior são utilizados porque permitem a geração de código por meio do Simulink, o solucionador FBstab foi implementado utilizando a interface desenvolvida para o Matlab, para os 3 solucionadores foi utilizada uma função

Matlab no Simulink. Os parâmetros utilizados no modelo do veículo são os apresentados na tabela 13, e as restrições impostas são mostradas na tabela 9.

Tabela 13 – Parâmetros para o modelo do erro do veículo com perturbação em relação à pista

Parâmetro	Valor	Unidade
m	1.575	Kg
I_z	2.875	Kg.m ²
a	1,2	m
b	1,6	m
$C_{\alpha f}$	19.000	N/rad
$C_{\alpha r}$	33.000	N/rad
D	1,795	m
V_x	14	m/s

4.3.1 Análise do controlador MPC clássico em *Hardware-In-the-Loop*

Os resultados mostrados na figura 26 comparam o desempenho dos controladores MPC e LQR para a função de manutenção de faixa em *Hardware-In-the-Loop*. O controlador MPC clássico tem um perfil de comando de ângulo de roda suave em como é mostrado na figura 26 onde pode se observar o desempenho dos controladores MPC e LQR para a função de manutenção de faixa em HIL. O controlador MPC clássico tem um perfil de comando de ângulo de roda suave, o que é evidente no perfil da taxa de mudança de ângulo próximo a zero durante o experimento, o tempo computacional do controlador MPC clássico tem um pico máximo de 355 μ s. Além disso, nenhuma das três variáveis com restrições impostas excede seu limite, mantendo o veículo dentro da zona de segurança definida para essa função. O controlador LQR tem um comportamento mais oscilante no perfil do ângulo da roda, com uma taxa de variação maior, o que pode afetar o conforto e a estabilidade do veículo (68cm). No entanto, é importante observar que o tempo de cálculo do controlador LQR tem um pico máximo de 131 μ s, que é menor do que o do controlador MPC clássico, além de ter um tempo de média digital de 77 μ s.

A parte superior esquerda da figura 27 mostra os perfis de comando do ângulo da roda, em que um perfil mais suave do ângulo da roda é notável para o solucionador de conjunto ativo. Já para os solucionadores quadráticos de ponto interior e FBstat, eles se sobrepõem em muitos pontos dentro do experimento. Também uma maior oscilação é mostrada no início do experimento devido à condição inicial do ângulo de guinada igual a 0,012467rad, O lado superior direito da figura 27 mostra o perfil de deslocamento de cada solucionador, o perfil de deslocamento do solucionador de conjunto ativo é menor em comparação com os outros dois solucionadores quadráticos, como é mostrado no início do experimento, para os solucionadores quadráticos de ponto interior e FBstab o deslocamento máximo atingido é de 10cm, Finalmente, a parte inferior mostra o tempo de execução de cada solucionador de QP.

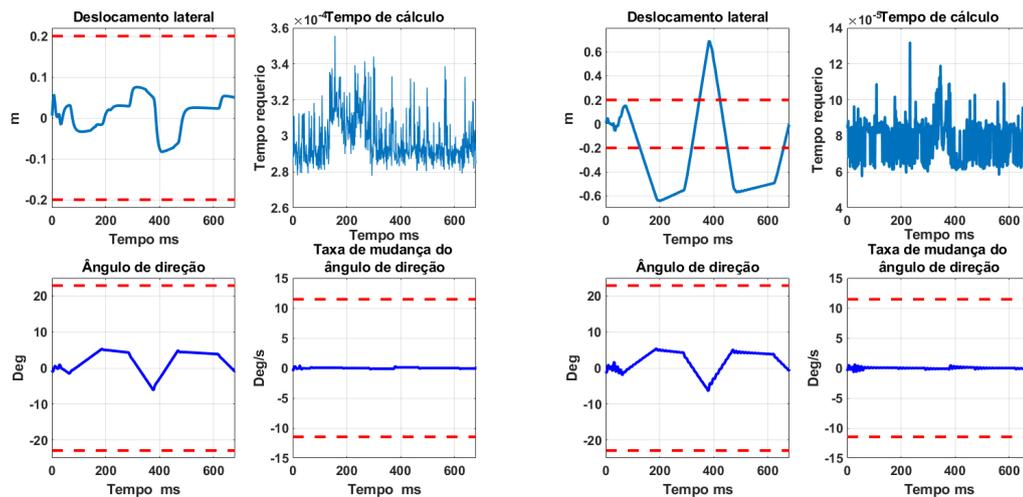


Figura 26 – Desempenho do controlador MPC lado esquerdo e Desempenho do controlador LQR lado direito para LKAS.

Tabela 14 – Parâmetros para controlador MPC HIL

Parâmetros	Conjunto ativo QP	Ponto interior QP	Fbstab QP
Qy	[2E7,1E7,40.070,121.265,1E2]	[2E7,1E7,40.070,121.265,1E2]	[2E7,1E7,40.070,121.265,1E5]
Qu	1,00E+10	1,00E+10	1,00E+10
N	20	20	15
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	2,97E-04	3,40E-03	3,50E-03

O comportamento mais diferenciado é o do controlador FBstab, que apresenta diferentes picos que podem estar relacionados ao momento de giro ou à mudança de curvatura no cenário do experimento. É importante notar que nenhum dos solucionadores atingiu as restrições de estabilidade neste experimento.

Já no experimento utilizando o controlador MPC com parametrização trivial, os parâmetros de penalidade foram definidos para cada um dos solucionadores. Em geral, todos os três solucionadores foram penalizados de forma semelhante, mas o controlador FBstab não foi possível simular em tempo real com um valor mais alto no horizonte de previsão em 1.5s, em todos esses casos, o tempo do processo de otimização excedeu o tempo de amostragem definido em 100ms que gerou uma violação temporal dentro da ECU Baseline, para esse solucionador foi utilizada uma penalidade maior no comando de endereço em comparação com os outros dois solucionadores quadráticos, conforme mostrado na tabela 14.

Os tempos alcançados no MPC clássico mostrados na tabela acima mostram que o melhor tempo médio obtido é com o método do conjunto ativo com $249\mu s$, seguido pelo método do ponto interior com um tempo médio igual a $3,5ms$ e, por último, o FBstab, que tem um horizonte de previsão menor $N = 15$ e ainda tem um tempo médio $3,5ms$ melhor do que os outros dois solucionadores.

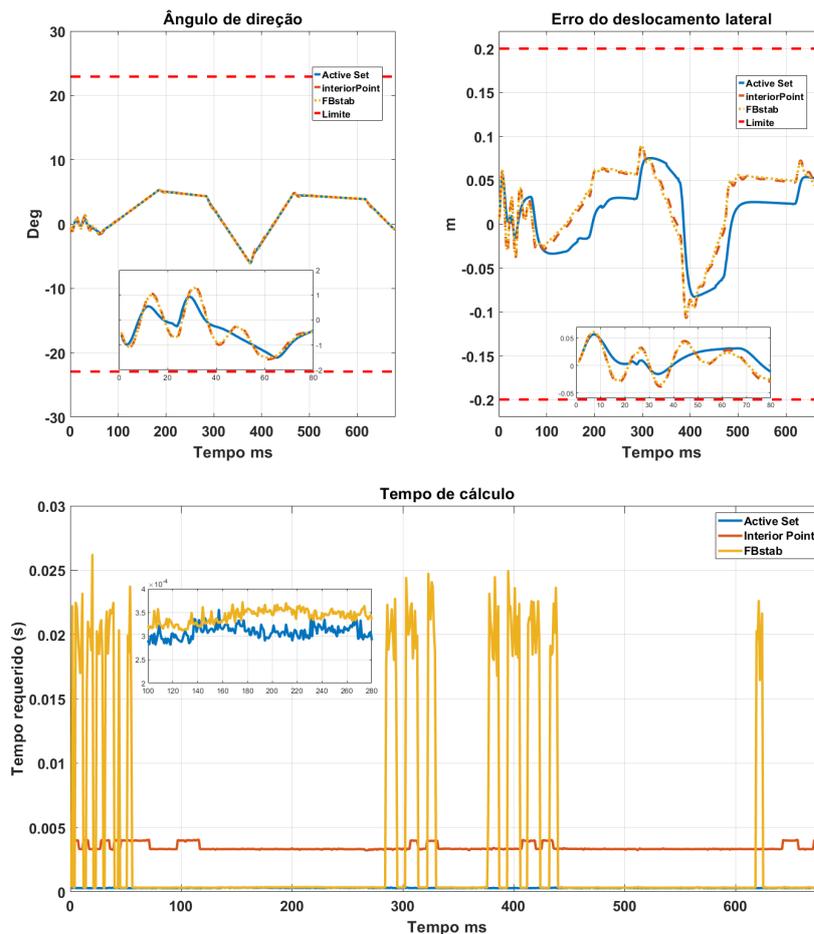


Figura 27 – Comando de entrada e tempo de cálculo para solucionadores quadráticos de controlador MPC

4.3.2 Análise do controlador MPC com parametrização Trivial em *Hardware-In-the-Loop*

Os resultados da formulação de parametrização trivial com o conjunto ativo solucionador A são apresentados na figura 28. Neste caso é importante observar que o deslocamento lateral aumentou em relação ao controlador MPC ao atingir um máximo de 11cm do centro da faixa. Por outro lado, o perfil de comando do ângulo da roda apresentou uma breve oscilação nas mudanças de direção do ângulo quando comparado à formulação MPC clássica, o que se reflete em uma taxa de mudança com valores oscilantes. No entanto, isso não representa um problema, pois cada variável de estado do sistema (deslocamento lateral e ângulo de guinada) é mantida dentro das restrições estabelecidas.

A Figura 29 apresenta uma visão detalhada dos resultados obtidos com a aplicação do controlador MPC com a formulação de redução de carga de parametrização trivial. Essa figura é dividida em três seções: na parte superior esquerda, é exibido o perfil de ângulo dos três solucionadores utilizados; na parte superior direita, são apresentados os deslocamentos do centro da pista; e na parte inferior, é ilustrado o tempo exigido por cada solucionador QP.

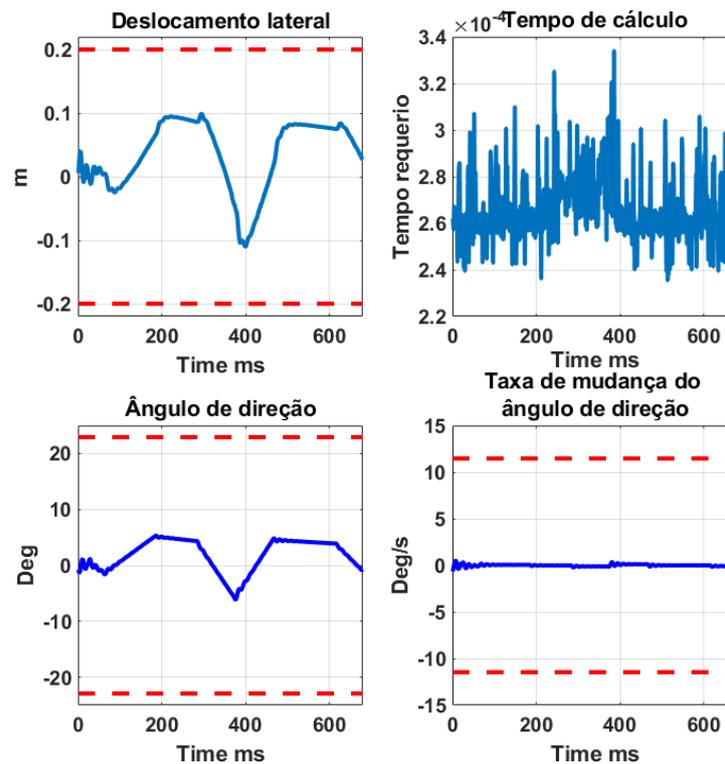


Figura 28 – Desempenho do controlador MPC com parametrização trivial para LKAS.

Com relação ao perfil de comando dos solucionadores quadráticos, observa-se que o solucionador FBstab apresenta um comportamento mais oscilatório em comparação com os solucionadores quadráticos de conjunto ativo e de ponto interior, que apresentam respostas semelhantes e se sobrepõem durante a maior parte do experimento. Em termos de deslocamento lateral, o solucionador FBstab apresenta uma oscilação inicial mais pronunciada, enquanto os outros dois solucionadores quadráticos apresentam resultados semelhantes. Na seção inferior da figura, destaca-se que o solucionador FBstab não apresenta mais os picos computacionais pronunciados que foram observados no controlador MPC clássico. Além disso, o perfil de tempo exigido por cada solucionador é claramente evidente, destacando que, para essa simulação, foi usado um horizonte de previsão $N=20$ para o solucionador FBstab, um valor que não permitiu a execução do controlador MPC clássico.

Na tabela 15 são apresentados os valores da penalidade para cada variável e o comando de controle, além disso, são observados os graus de liberdade utilizados em geral para a simulação, esses graus de liberdade afetam de forma direta a penalidade da variável de estado e o comando e esses valores são escolhidos em geral porque permitem a penalidade dos estados mais próximos no horizonte de previsão.

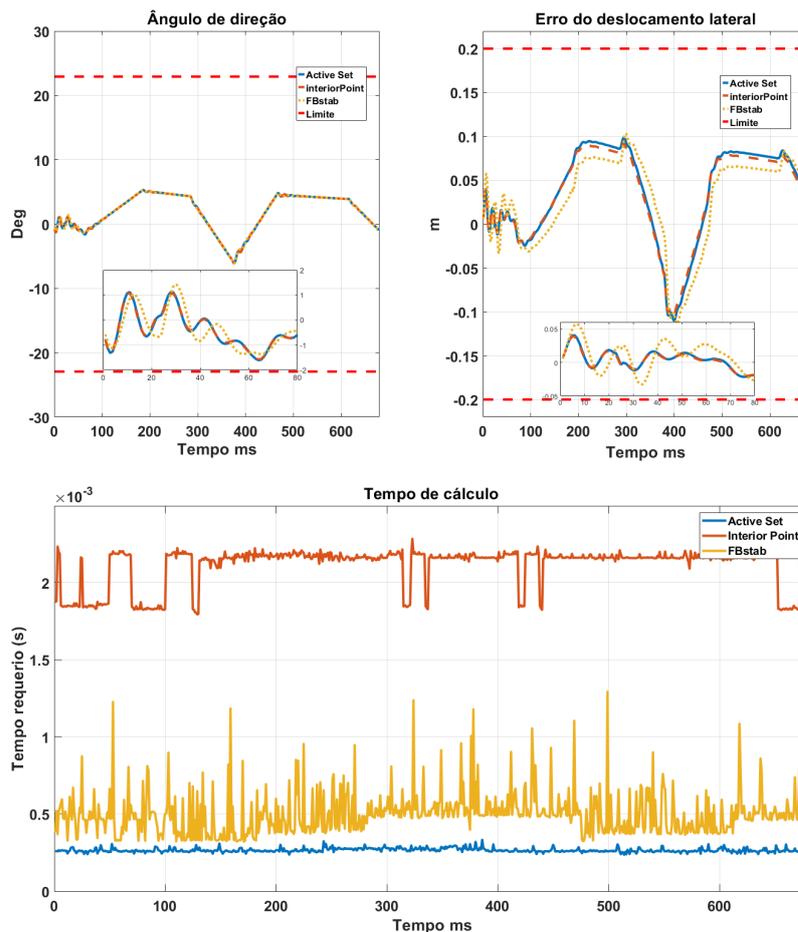


Figura 29 – Comando de entrada e tempo de cálculo do solucionador para o controlador trivial mpc

Por outro lado, conforme mostrado graficamente na figura 29, o solucionador de conjunto ativo obteve o melhor tempo médio com $265ms$, seguido pelo FBstab com $400ms$ e, por fim, o solucionador de ponto interior com um valor de $2,1ms$. Para essa formulação, foram encontrados diferentes valores de penalidade do solucionador e, em geral, a penalidade da taxa de variação foi superior a $1E10$ para todos os solucionadores.

Tabela 15 – Parâmetros para controlador Trivial HIL

Parâmetros	Conjunto ativo QP	Ponto interior QP	Fbstab QP
Qy	[4E7,5E7,40.070,121.265,1E5]	[4.5E7,5E7,40.070,121.265,1E5]	[2E9,2E9,40.070,121.265,1E10]
Qu	1,00E+10	1,00E+10	1,00E+12
DOF	[3;5]	[3;5]	[3;5]
N	20	20	20
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	2,65E-04	2,10E-03	4,99E-04

4.3.3 Análise do controlador MPC com parametrização exponencial em *Hardware-In-the-Loop*

Os resultados obtidos com a implementação da formulação de redução de carga computacional exponencial com o solucionador de conjunto ativo em *Hardware-In-the-Loop* são apresentados na figura 30. O perfil de deslocamento lateral do modelo segue uma trajetória semelhante à obtida com o controlador MPC com parametrização trivial, com um valor máximo de $8,8\text{cm}$ resultando em uma diminuição do valor máximo em comparação com a formulação MPC com parametrização trivial. Para o perfil de comando do ângulo da roda, mesmo no início é observada uma oscilação, no entanto, no restante do experimento, ele permanece dentro das restrições impostas. Durante o experimento, é observada uma diminuição do tempo, pois há menos picos de tempo de cálculo e o pico máximo atingido durante todo o experimento é de $300\mu\text{s}$. Em geral, durante o restante do experimento, o modelo do veículo permanece dentro da zona de segurança. Deve-se observar que o tempo computacional do solucionador é reduzido para uma média de $248\mu\text{s}$. O tempo médio de computação é reduzido em comparação com a formulação trivial, mantendo o perfil do ângulo de comando e reduzindo o deslocamento máximo obtido, o que indica uma melhoria na eficiência do processo de controle. Na figura 31 são apresentados os resultados para os

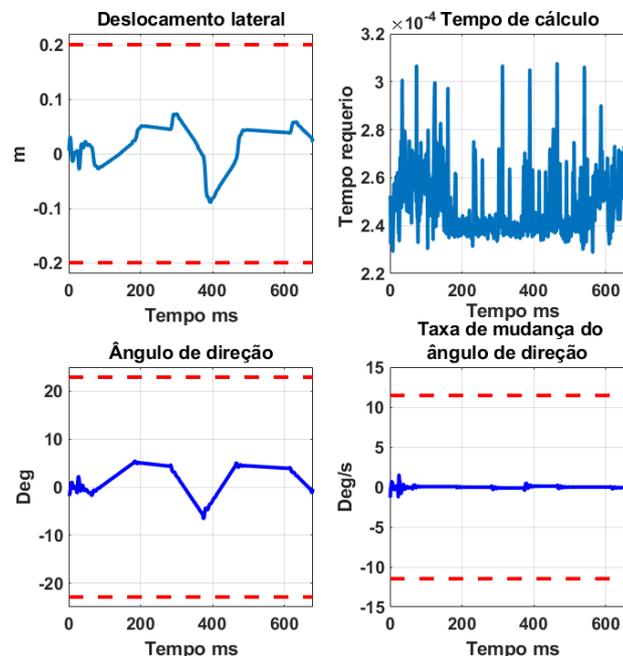


Figura 30 – Desempenho do controlador MPC com parametrização exponencial para LKAS.

três solucionadores QPs em utilizados em HIL com controlador MPC com parametrização exponencial, a parte superior esquerda mostra o perfil do comando de ângulo, um comportamento semelhante é observado, destacando que o solucionador de ponto interno apresenta uma oscilação maior em comparação com os outros solucionadores quadráticos. O valor máximo atingido pelos 3 solucionadores quadráticos foi de 6.6deg , Já no canto superior

direito apresenta o deslocamento lateral para cada solução, onde é possível observar uma maior oscilação do solucionador de ponto interno, que é o que, durante a simulação, atinge o maior deslocamento lateral com $8,1cm$ sem atingir a restrição imposta e, em geral, os três solucionadores obtiveram uma resposta muito semelhante. Por fim, na parte inferior da figura é apresentado o tempo de execução, onde se destaca a ausência de elevados picos computacionais apresentados pelo solucionador no FBstab, além de mostrar a diferença no tempo de cálculo dos três solucionadores QP.

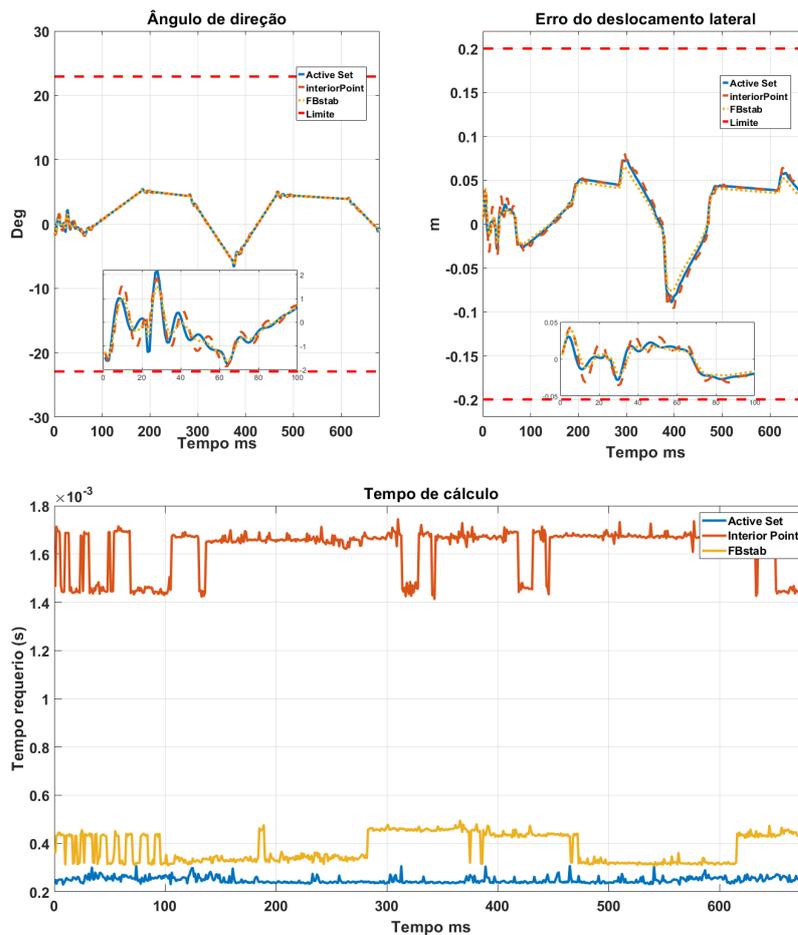


Figura 31 – Comando de entrada e tempo de cálculo dos solucionadores para controlador MPC com parametrização exponencial

Na tabela 16 são apresentados os parâmetros de penalidade para cada solucionador. Neste caso vale a pena observar que a penalidade da variável de controle dentro da função de custo diminuiu significativamente quando é comparada com a formulação trivial, o que é evidente em um comando brevemente mais oscilante do que o MPC com parametrização trivial, o que não representa uma diminuição no desempenho e na compreensão do controlador.

Além disso é possível observar uma diminuição no tempo médio obtido por cada solucionador, como os anteriores, o solucionador conjunto ativo obteve o melhor tempo com $248\mu s$ seguido por FBstab com $386\mu s$ e, por fim, o solucionador de ponto interior com $1,6ms$, o número de exponenciais (NoE) permaneceu constante para os 3 solucionadores.

Já para o parametro alfa (α) mudou para cada solucionador, pois o tempo proposto em que a variável de controle atinge a referência é diferente e, para essa formulação, é necessário encontrar um equilíbrio entre esses dois parâmetros, o que pode tornar mais sensível a penalidade de cada uma das variáveis dentro do sistema, conforme evidenciado na tabela 16.

Tabela 16 – Parâmetros para controlador MPC com parametrização exponencial em HIL

Parâmetros	Conjunto ativo QP	Ponto interior QP	Fbstab QP
Qy	[1.168E4,390.510,40070,121.265,1E4]	[1.168E3,390.510,8.014,24.253,1E1]	[1168*1E5,39.051E2,801400,24.253E2,1E5]
Qu	1,00E+05	1,00E+00	1,00E+06
NoE	2	2	2
α	1.250	15	4.525
t_r	0.001	0.1	0.001
N	20	20	20
Max Iter	100	100	80
Tolerância	1.E-04	1.E-04	1.E-06
Tempo médio (s)	2,48E-04	1,60E-03	3,86E-04

4.3.4 Análise comparativa dos controladores MPCs em *Hardware-In-the-Loop*

Na figura 32 é mostrada uma análise comparativa do experimento dos controladores em *Hardware-In-the-Loop* com três solucionadores quadráticos QPs do método do conjunto ativo, do método do ponto interior e do FBstab. A combinação revela resultados significativos em termos de tempo computacional dos solucionadores quadráticos e seu desempenho no rastreamento da refração e no comando do ângulo das rodas. Entre as várias combinações avaliadas, é importante destacar que o solucionador que obteve o melhor tempo computacional foi o método do conjunto ativo, quando a formulação de redução de custo de parametrização exponencial é aplicada. Essa abordagem atinge um tempo médio de $248\mu s$, o que indica uma notável eficiência computacional. Em segundo lugar, o solucionador FBstab também demonstra melhor desempenho em termos de tempo computacional, com um tempo computacional médio de $386\mu s$. Vale ressaltar que, ao encontrar o melhor tempo com a formulação MPC com parametrização exponencial, os solucionadores encontraram uma resposta semelhante, o que permite demonstrar uma configuração adequada para cada solucionador e observar seu bom desempenho no rastreamento da trajetória e na manutenção do modelo do veículo dentro das zonas de segurança.

Como no ciclo de teste MIL, o solucionador com o pior tempo médio é o método do ponto interior. Essa abordagem é caracterizada por um desempenho mais lento em comparação com os outros solucionadores avaliados. Esses resultados destacam a importância da escolha do solucionador e da formulação de parametrização sobre a eficiência do controlador

MPC, o que pode ter implicações significativas em aplicativos e sistemas em tempo real que exigem respostas rápidas.

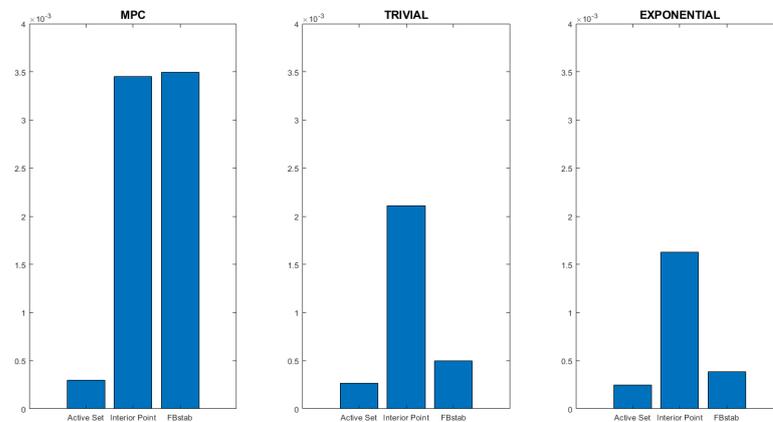


Figura 32 – Comparação de todos os solucionadores QP.

A tabela 17 são apresentados a taxa de redução de cada solucionador com cada formulação do controlador MPC. Neste experimento a maior redução de tempo foi obtida pelo solucionador Fbstab comparando desde a formulação Clássica com a formulação exponencial com um valor de 89%. Seguido pelo método do ponto interior com 53%. Em geral, também é evidente que as piores porcentagens de redução de tempo foram para o método do conjunto ativo com um valor de 16%. Também vale a pena mencionar que, embora o Fbstab tenha obtido o melhor percentual de redução de tempo e se tornado um potencial para implementação em tempo real, o conjunto ativo solucionador obteve a melhor média de computação para cada uma das formulações de MPC. Para essa simulação a medição do tempo foi obtida pelo bloco (SLRT Overload Options) do Simulink, esse bloco além de permitir configurar as opções de sobrecarga. Também permite o cálculo do tempo da tarefa em geral, isso é evidenciado nos tempos apresentados, já que em *Model-In-the-Loop* o tempo de cálculo é a média do tempo de execução do controlador MPC apresentando valores menores do que os apresentados para a simulação no HIL para esse exemplo, o tempo da tarefa leva em conta outros processadores, como a atribuição de parâmetros e as taxas de alteração do tempo total do bloco, mesmo assim a redução real de cada uma das formulações com cada solucionador é evidente.

Tabela 17 – Comparação das diferentes formulações na redução do tempo médio de computação

Solucionador QP	Clássico vs Trivial	Trivial vs Exponencial	Clássico vs Exponencial
Conjunto ativo QP	11%	6%	16%
Ponto interior QP	38%	24%	53%
Fbstab QP	86%	23%	89%

Na figura 33 são apresentados os tempos médios para cada uma das formulações do controlador MPC com o conjunto ativo solucionador que é apresentado como a melhor

Tabela 18 – Resumo de tabela de resultados de redução porcentual do tempo

	Solucionador QP	Clássico vs Trivial	Trivial vs Exponencial	Clássico vs Exponencial
Modelo do erro com ciclo de teste em <i>Model-In-the-Loop</i>	Conjunto ativo QP	50%	26%	63%
	Ponto interior QP	5%	3%	8%
	Fbstab QP	90%	31%	93%
Modelo do erro com ciclo de teste em <i>Hardware-In-the-Loop</i>	Conjunto ativo QP	11%	6%	16%
	Interior Point QP	38%	24%	53%
	Fbstab QP	86%	23%	89%

opção para a execução na função LKAS em tempo real do controlador. Essa formulação está disponível na interface do Matlab e Simulink e permite a opção de geração de código para C e C++, vale ressaltar que a solução FBstab poderia representar uma segunda opção fazendo uso da formulação de exponencial.

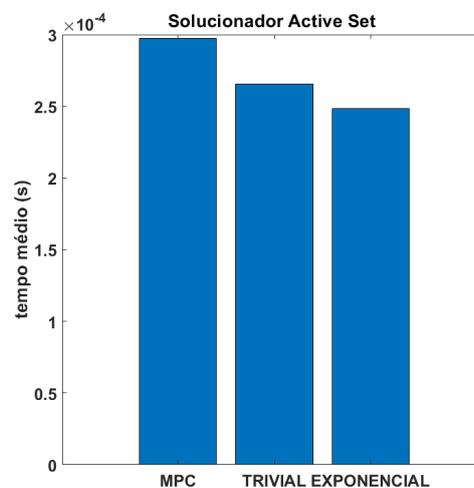


Figura 33 – Melhor tempo encontrado combinando o solucionador e a formulação

4.3.5 Análise da redução percentual

Nesta seção, será apresentada uma análise comparativa entre os ciclos de teste MIL (*Model-In-the-Loop*) e HIL (*Hardware-In-the-Loop*) para as formulações propostas do controlador MPC e os solucionadores quadráticos associados. O objetivo é evidenciar se existe uma correlação entre os resultados obtidos em cada um dos ciclos. Essa análise permitirá selecionar uma formulação e um solucionador quadrático adequado para sua possível aplicação em uma ECU real.

Na tabela 18, é possível avaliar a diferença entre os ciclo de testes em MIL e HIL. Destaca-se a similaridade entre os resultados do método FBstb para os dois cenários de simulação, que estão próximos nos valores de redução percentual. Por outro lado, para os solucionadores de ponto interior e conjunto ativo de respostas, observa-se uma notável diferença nos valores percentuais de redução do tempo computacional.

Com cada formulação de parametrização do controlador, há uma redução geral do tempo de cálculo, sendo que para a formulação de parametrização exponencial, a redução

percentual é maior. Esses resultados oferecem opções para a aplicação do controlador MPC em tempo real, considerando a possibilidade de não encontrar o código do solucionador para uma plataforma ou microcontrolador específico. Além disso, permitem uma escolha mais criteriosa do solucionador, formulação e hardware para a execução do controlador.

Essa tabela reforça a necessidade de uma boa escolha do solucionador e da formulação de formulação ideal para a função LKAS (*Lane Keeping Assist System*) na execução em tempo real. Vale ressaltar que, neste caso, o solucionador com o melhor tempo de cálculo foi o solucionador de conjunto ativo. Apesar de não representar o maior valor percentual de redução na tabela, destaca-se a redução percentual apresentada pelo solucionador FBstab na formulação exponencial, atingindo 89%. Isso equivale a uma redução de 3,5ms para 0,386ms, comparável ao melhor tempo do solucionador de conjunto ativo, que foi de 0,248ms.

4.3.6 Análise de robustez em *Hardware-In-the-Loop*

Realizar uma análise de robustez do controlador MPC em um ambiente HIL é crucial para validar seu desempenho em condições realistas. A capacidade de testar o controlador em um ambiente que simula o comportamento do veículo e as condições da estrada garante que ele possa se adaptar de maneira adequada a situações práticas. Além disso, a avaliação da robustez em um ambiente HIL permite identificar e abordar as limitações do controlador MPC. Ao introduzir diferentes níveis de variabilidade e perturbações no sistema, é possível identificar possíveis fraquezas e otimizar o projeto do controlador para melhorar sua capacidade de manter o veículo dentro da faixa em condições adversas.

Tabela 19 – Novos parâmetros para o modelo do erro do veículo em relação à pista

Parâmetro	Valor	Novo valor	Unidade
m	1575	2047,5	Kg
I_z	2875	3737,5	Kg.m ²
a	1,2	1,2	m
b	1,6	1,6	m
$C_{\alpha f}$	19000	13300	N/rad
$C_{\alpha r}$	33000	23100	N/rad
D	1,795	1,795	m
V_x	14	14	m/s

Para testar a robustez e a estabilidade da formulação de redução do tempo computacional e seu comportamento em situações de mudança no ambiente e no modelo, os valores da massa do veículo foram modificados aumentando a massa em um 30% e uma redução do coeficiente de atrito da roda em um 30%, com essas variações nos parâmetros do veículo, espera-se verificar se o controlador mantém as restrições estabelecidas para a segurança, o conforto e as limitações físicas do sistema, como o ângulo máximo das rodas do veículo, demonstrando sua capacidade de robustez e estabilidade em situações ambientais e parâmetros do modelo em constante mudança. Na figura 34 são apresentados os resultados do experimento no HIL com o solucionador de conjunto ativo com as modificações propostas.

No lado direito, é mostrado o controlador LQR, no qual há evidências de um afastamento da zona de segurança proposta para o veículo, aumentando o risco de partida do veículo; é importante observar que o tempo de cálculo desse controlador é menor quando comparado ao controlador MPC; por outro lado, o controlador MPC com a formulação exponencial atinge as restrições impostas e permanece dentro da zona de segurança. É evidente que, ao atingir as restrições, o ângulo das rodas e sua taxa de variação aumentam em magnitude. Nesse ponto, é importante destacar que a penalização dessas variáveis deve permitir uma leve reação em casos de emergência ou situações de risco, pois a penalização incorreta poderia não permitir que o controlador encontrasse um ângulo adequado para permanecer dentro das restrições impostas. Para a execução desse cenário, os parâmetros de penalidade foram mantidos em relação à tabela 16 e os parâmetros do veículo são mostrados na tabela 19.

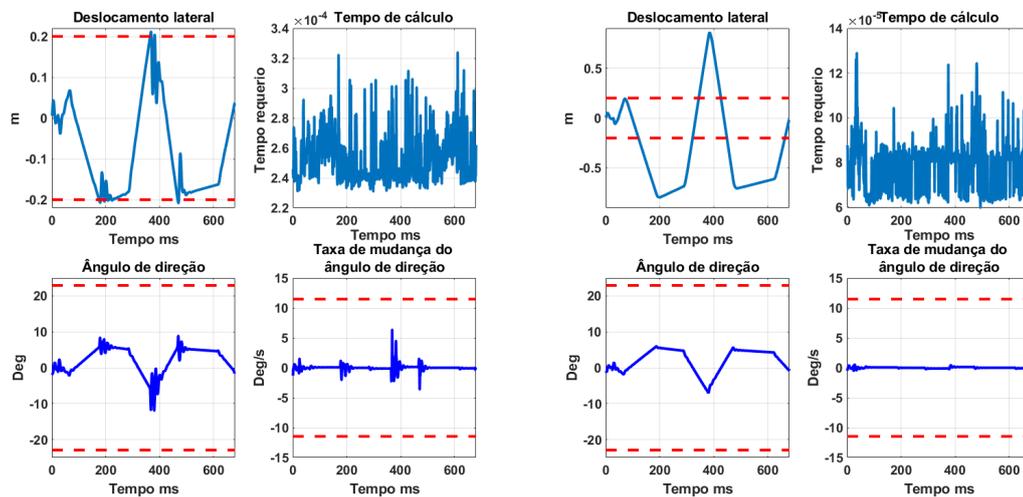


Figura 34 – Desempenho do controlador MPC e Desempenho do controlador LQR para LKAS com parâmetros diferentes do modelo (Aumento de massa em um 30% e uma redução do coeficiente de atrito em um 30%).

5 Conclusões

Esta pesquisa concentrou-se em conceber e implementar estratégias de MPC utilizando diferentes representações de modelos de dinâmica veicular. Foram exploradas formulações teóricas de parametrização para reduzir a carga computacional do controlador MPC, incluindo uma parametrização trivial e exponencial. Além disso, foi utilizado um controlador LQR para comparar seu desempenho com as formulações propostas de parametrização com o controlador MPC em termos de desempenho e eficiência computacional.

Os resultados obtidos dos experimentos e análises comparativas de vários solucionadores de Programação Quadrática (QPs) em combinação com as diferentes formulações de controle MPC com parametrização propostas para a função LKAS demonstraram que os objetivos estabelecidos foram alcançados. A eficiência do controlador MPC e o tempo de computação resultante com os diferentes modelos (rastreamento de trajetória e erro em relação ao centro da faixa) nos ciclos de teste MIL e HIL foram testados satisfatoriamente. Além disso, a aplicação de técnicas de parametrização em diferentes solucionadores QP permitiu realizar comparações abrangentes em termos de desempenho e redução percentual do tempo de cálculo do controlador MPC e carga computacional. O cenário de teste em HIL destacou a viabilidade e aplicabilidade do controlador MPC em uma ECU real, enfatizando sua capacidade de adaptação a condições operacionais diversas e variáveis.

Um dos aspectos mais significativos desta pesquisa foi a demonstração da robustez e estabilidade do controlador diante de mudanças nos parâmetros do cenário e do modelo no ciclo de teste HIL. Esta capacidade de adaptação e resposta do controlador MPC com parametrização exponencial ressalta sua eficácia e confiabilidade em ambientes dinâmicos e variáveis. A análise em tempo real das soluções baseadas em MPC para LKAS proporcionou uma compreensão mais profunda de seu desempenho em situações práticas. Essas avaliações em tempo real permitiram identificar áreas de melhoria e otimização para garantir um funcionamento ideal do sistema LKAS em condições reais de operação.

Os resultados alcançados da concepção e implementação das estratégias de modelo de controle preditivo baseado em modelos (MPC) para LKAS, utilizando diferentes representações de modelos de dinâmica veicular, demonstram que as estratégias de redução de custos computacionais foram eficazes na correção da trajetória do veículo, além de manter as restrições impostas para a segurança, conforto e físicas do sistema. O desenvolvimento de técnicas destinadas a reduzir a carga computacional do MPC através da parametrização adequada das sequências de controle demonstram que foi possível atingir este objetivo. As avaliações comparativas das abordagens teóricas de parametrização revelaram uma redução significativa no tempo de computação sem comprometer a precisão do controle, obtendo, no

caso dos modelos de rastreamento, o solucionador quadrático QSWIFT com a formulação do controlador com parametrização exponencial como o melhor tempo médio de execução no cenário de teste, e permitindo que outros solucionadores alcancem tempos mais viáveis para execução, como no FBstab. Por outro lado, na execução do cenário de teste para o modelo de erro, foi encontrado que o solucionador de conjunto ativo, em conjunto com a formulação exponencial, obteve o melhor tempo de execução tanto no ciclo de teste MIL quanto no HIL.

Também a realização de experimentos e análise comparativa de vários solucionadores de Programação Quadrática (QP) para avaliar a eficiência do controlador e o tempo de computação resultante evidenciam que determinadas formulações do MPC apresentaram desempenho superior em termos de eficiência e tempo de computação, para este caso a formulação de controle MPC com parametrização exponencial.

Os resultados da aplicação de técnicas de parametrização em vários solucionadores QP estabelecidos, efetuando comparações exaustivas em termos de desempenho do controlador e carga computacional entre diferentes formulações do MPC, mostram as vantagens e limitações de cada formulação do MPC, fornecendo valiosos pontos para a seleção do método mais adequado para diferentes cenários de aplicação da função LKAS.

5.0.1 Limitações da pesquisa

Os cenários de testes desenvolvidos nesta pesquisa envolvem o comportamento real do veículo. No entanto, no desenho e execução dos experimentos com os controladores, considerou-se só a dinâmica lateral do veículo, mantendo a velocidade longitudinal constante ao longo de todo o percurso. Isso pode diferir do modelo real do veículo. Além disso, a comunicação utilizada na execução do controlador em tempo real foi Ethernet, o que poderia representar um cenário ideal, visto que em um veículo real podem existir latências que não foram consideradas nesta pesquisa. Também na execução dos cenários de teste, foi utilizado um modelo de câmera simulada dentro do ambiente do Simulink, o que pode não representar de forma adequada uma câmera real com a latência que esta possui intrinsecamente.

Por outro lado, a utilização de uma plataforma de prototipagem rápida em tempo real, como a máquina Baseline da empresa Speedgoat, pode simular uma Unidade de Controle Eletrônico (ECU) com capacidades superiores de uma ECU real de um veículo. Além disso, a utilização de solucionadores quadráticos poderia ser ampliada se houvesse uma interface que permitisse a codificação e geração de código a partir do MATLAB/Simulink.

5.0.2 Trabalhos futuros

Existem diversas melhorias para o sistema LKAS e controlador MPC proposto nesta dissertação, começando com o modelo em que o torque pode ser incorporado como variável de comando, o que permitiria uma implementação mais rápida dentro do veículo

real, a implementação do controlador MPC mostrou um comportamento adequado para a aplicação em tempo real, por essa razão, o próximo passo é a aplicação em um veículo real, também para a aplicação em tempo real deve ser desenvolvida a função para enviar o sinal de comando por meio do protocolo de comunicação CAN. Outro trabalho futuro é a integração de soluções como o QPswift para execução em tempo real, pois ele mostrou um entendimento adequado para a aplicação do controlador com estratégia de formulação exponencial. o desenvolvimento de um aplicativo para o reconhecimento das principais características do veículo para uma integração mais completa em um veículo real, também deve ser buscada a adição da dinâmica longitudinal ao controlador, já que neste trabalho a velocidade constante é considerada.

Referências

- ADLER, I.; HU, Z. T.; LIN, T. New proximal newton-type methods for convex optimization. In: IEEE. 2020 59th IEEE Conference on Decision and Control (CDC). 2020. P. 4828–4835. Citado na p. 48.
- ALAMIR, M. A. Pragmatic Story of Model Predictive Control: Self Contained Algorithms and Case-studies. (CreateSpace Independent Publishing Platform, 2013. Citado nas pp. 17, 32, 34, 36, 39, 40, 43, 59, 72.
- AN, Q.; CHENG, S.; LI, L.; PENG, H. Novel dual-layer-oriented strategy for fully automated vehicles' lane-keeping system. **IET Intelligent Transport Systems**, Wiley Online Library, v. 14, n. 13, p. 1778–1787, 2020. Citado nas pp. 16, 57.
- BAR-SHALOM, Y.; LI, X. R.; KIRUBARAJAN, T. **Estimation with applications to tracking and navigation: theory algorithms and software**. John Wiley & Sons, 2004. Citado na p. 24.
- BENINE-NETO, A.; MAMMAR, S. Piecewise affine output feedback controller for vehicle lane keeping. In: IEEE. 2012 American Control Conference (ACC). 2012. P. 6533–6538. Citado nas pp. 57, 58.
- BIAN, C.; YIN, G.; ZHANG, N.; XU, L. Takagi-sugeno fuzzy model predictive controller design for combining lane keeping and speed tracking of four wheels steering and four wheels drive electric vehicle. In: IEEE. 2017 29th Chinese Control And Decision Conference (CCDC). 2017. P. 4067–4072. Citado na p. 58.
- BOJARSKI, M.; DEL TESTA, D.; DWORAKOWSKI, D.; FIRNER, B.; FLEPP, B.; GOYAL, P.; JACKEL, L. D.; MONFORT, M.; MULLER, U.; ZHANG, J. et al. End to end learning for self-driving cars. **arXiv preprint arXiv:1604.07316**, 2016. Citado na p. 55.
- BORKAR, A.; HAYES, M.; SMITH, M. T. A novel lane detection system with efficient ground truth generation. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 13, n. 1, p. 365–374, 2011. Citado na p. 24.
- CARMAKER. **Carmaker**. Disponível em: <<https://ipg-automotive.com/en/products-solutions/software/carmaker/>>. Acesso em: 6 nov. 2023. Citado na p. 60.
- CARSIM. **Carsim - Trucksim**. Disponível em: <<https://www.carsim.com/>>. Acesso em: 6 nov. 2023. Citado na p. 60.
- CHEN, C.; JIA, Y.; DU, J.; YU, F. Lane keeping control for autonomous 4WS4WD vehicles subject to wheel slip constraint. In: IEEE. 2012 American Control Conference (ACC). 2012. P. 6515–6520. Citado na p. 57.

- CHEN, B.-C.; LUAN, B.-C.; LEE, K. Design of lane keeping system using adaptive model predictive control. In: IEEE. 2014 IEEE International conference on automation science and engineering (CASE). 2014. P. 922–926. Citado nas pp. 16, 59.
- CHEN, J.; SUN, D.; ZHAO, M.; LI, Y.; LIU, Z. A new lane keeping method based on human-simulated intelligent control. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 23, n. 7, p. 7058–7069, 2021. Citado nas pp. 53, 61.
- CHEN, Y.; SCARABOTTOLO, N.; BRUSCHETTA, M.; BEGHI, A. Efficient move blocking strategy for multiple shooting-based non-linear model predictive control. **IET Control Theory & Applications**, v. 14, p. 343–351, 2020. Citado na p. 39.
- CHEN, Z.; LI, L.; HUANG, X. Building an autonomous lane keeping simulator using real-world data and end-to-end learning. **IEEE Intelligent Transportation Systems Magazine**, IEEE, v. 12, n. 1, p. 47–59, 2018. Citado nas pp. 20, 24, 52, 53, 55, 60, 62.
- CHU, Z.; SUN, Y.; WU, C.; SEPEHRI, N. Active disturbance rejection control applied to automated steering for lane keeping in autonomous vehicles. **Control Engineering Practice**, Elsevier, v. 74, p. 13–21, 2018. Citado nas pp. 20, 57.
- DAI, S.; KOUTSOUKOS, X. Safety analysis of integrated adaptive cruise and lane keeping control using multi-modal port-Hamiltonian systems. **Nonlinear Analysis: Hybrid Systems**, Elsevier, v. 35, p. 100816, 2020. Citado na p. 60.
- DIAB, M. K.; AMMAR, H. H.; SHALABY, R. E. Self-driving car lane-keeping assist using pid and pure pursuit control. In: IEEE. 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT). 2020. P. 1–6. Citado na p. 62.
- DING, Y.; XU, Z.; ZHANG, Y.; SUN, K. Fast lane detection based on bird’s eye view and improved random sample consensus algorithm. **Multimedia Tools and Applications**, Springer, v. 76, p. 22979–22998, 2017. Citado na p. 22.
- DOSOVITSKIY, A.; ROS, G.; CODEVILLA, F.; LOPEZ, A.; KOLTUN, V. CARLA: An Open Urban Driving Simulator. In: PROCEEDINGS of the 1st Annual Conference on Robot Learning. 2017. P. 1–16. Citado na p. 60.
- DSPACE. **Dspace**. 2023. Disponível em: <<https://www.dspace.com/en/pub/home.cfm>>. Acesso em: 7 nov. 2023. Citado na p. 60.
- FARAG, W. Real-time NMPC path tracker for autonomous vehicles. **Asian Journal of Control**, Wiley Online Library, v. 23, n. 4, p. 1952–1965, 2021. Citado na p. 56.
- FARAG, W. A. Model-predictive-control complex-path tracking for self-driving cars. **International Journal of Modelling, Identification and Control**, Inderscience Publishers (IEL), v. 34, n. 3, p. 265–277, 2020. Citado nas pp. 30, 31, 58.

- FARONI, M.; BESCHI, M.; BERENGUEL, M.; VISIOLI, A. Fast MPC with staircase parametrization of the inputs: Continuous input blocking. **2017 22nd IEEE International Conference On Emerging Technologies And Factory Automation (ETFA)**, p. 1–8, 2017. Citado nas pp. [39](#), [59](#).
- FERREAU, H.; KIRCHES, C.; POTSCSKA, A.; BOCK, H.; DIEHL, M. q. A. parametric active-set algorithm for quadratic programming. **Mathematical Programming Computation**, v. 6, p. 327–363, 2014. Citado na p. [47](#).
- FLORES, C.; LU, X.-Y. Robust preview-based tractor-trailer lateral control for lane keeping. In: IEEE. 2020 American Control Conference (ACC). 2020. P. 1229–1234. Citado na p. [62](#).
- GE, Z.; WANG, Z.; BAI, X.; WANG, X. Robust Sliding Mode-Based Learning Control for Lane-Keeping Systems in Autonomous Vehicles. In: IEEE. 2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA). 2021. P. 1856–1861. Citado nas pp. [53](#), [62](#).
- GUO, C.; SENTOUH, C.; POPIEUL, J.; HAUÉ, J. MPC-based shared steering control for automated driving systems. **2017 IEEE International Conference On Systems, Man, And Cybernetics (SMC)**, p. 129–134, 2017. Citado na p. [59](#).
- GUO, J.; WANG, J.; LUO, Y.; LI, K. Takagi–sugeno fuzzy-based robust H integrated lane-keeping and direct yaw moment controller of unmanned electric vehicles. **IEEE/ASME Transactions on Mechatronics**, IEEE, v. 26, n. 4, p. 2151–2162, 2020. Citado nas pp. [20](#), [54](#), [57](#).
- HAN, Y.; LEE, H. Comparison and Analysis of Convex Optimization Methods for Potential Field-based Path Planning. In: IEEE. 2021 21st International Conference on Control, Automation and Systems (ICCAS). 2021. P. 1447–1450. Citado na p. [47](#).
- HTET, K. K. K.; KIONG, T. K.; XINXIN, D. Comprehensive lane keeping system with mono camera. In: IEEE. 2015 10th Asian Control Conference (ASCC). 2015. P. 1–6. Citado nas pp. [23](#), [25](#), [53–55](#).
- HU, C.; QIN, Y.; CAO, H.; SONG, X.; JIANG, K.; RATH, J. J.; WEI, C. Lane keeping of autonomous vehicles based on differential steering with adaptive multivariable super-twisting control. **Mechanical Systems and Signal Processing**, Elsevier, v. 125, p. 330–346, 2019. Citado nas pp. [54](#), [57](#).
- HUANG, J.; TAN, H.-S. Development and validation of an automated steering control system for bus revenue service. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 13, n. 1, p. 227–237, 2015. Citado nas pp. [44](#), [53](#), [63](#).

- IQBAL, M.; HAN, J. C.; ZHOU, Z. Q.; TOWEY, D. Enhancing Euro NCAP standards with metamorphic testing for verification of advanced driver-assistance systems. In: IEEE. 2021 IEEE/ACM 6th International Workshop on Metamorphic Testing (MET). 2021. P. 37–41. Citado na p. 21.
- JIANG, J.; ASTOLFI, A. Lateral control of an autonomous vehicle. **IEEE Transactions on Intelligent Vehicles**, IEEE, v. 3, n. 2, p. 228–237, 2018. Citado nas pp. 57, 62.
- JIANG, R.; KLETTE, R.; VAUDREY, T.; WANG, S. New lane model and distance transform for lane detection and tracking. In: SPRINGER. COMPUTER Analysis of Images and Patterns: 13th International Conference, CAIP 2009, Münster, Germany, September 2-4, 2009. Proceedings 13. 2009. P. 1044–1052. Citado na p. 55.
- JÚNIOR, Z.; MURILO, A.; LOPES, R. Vehicle stability upper-level-controller based on parameterized model predictive control. **IEEE Access**, v. 10, p. 21048–21065, 2022. Citado nas pp. 33, 40.
- KAMAT, S. Model Predictive Control Approaches for Lane Keeping of Vehicle. **IFAC-PapersOnLine**, Elsevier, v. 53, n. 1, p. 176–182, 2020. Citado nas pp. 30, 52, 58.
- KANG, C. M.; KIM, W.; CHUNG, C. C. Observer-based backstepping control method using reduced lateral dynamics for autonomous lane-keeping system. **ISA transactions**, Elsevier, v. 83, p. 214–226, 2018. Citado na p. 56.
- KANG, C. M.; LEE, S.-H.; CHUNG, C. C. Lane estimation using a vehicle kinematic lateral motion model under clothoidal road constraints. In: IEEE. 17TH International IEEE Conference on Intelligent Transportation Systems (ITSC). 2014. P. 1066–1071. Citado nas pp. 20, 25, 60, 63.
- KANG, C. M.; LEE, S.-H.; CHUNG, C. C. Lane keeping system based on kinematic model with road friction coefficient adaptation. In: IEEE. 2016 IEEE Intelligent Vehicles Symposium (IV). 2016. P. 552–557. Citado nas pp. 57, 63.
- KIM, D. J.; KANG, C. M.; LEE, S.-H.; CHUNG, C. C. Discrete-time integral sliding model predictive control for dynamic lateral motion of autonomous driving vehicles. In: IEEE. 2018 Annual American Control Conference (ACC). 2018. P. 4757–4763. Citado nas pp. 15, 31, 53, 54, 59.
- KIM, W.; SON, Y. S.; CHUNG, C. C. Torque-overlay-based robust steering wheel angle control of electrical power steering for a lane-keeping system of automated vehicles. **IEEE Transactions on Vehicular Technology**, IEEE, v. 65, n. 6, p. 4379–4392, 2015. Citado nas pp. 16, 53, 57.
- KITAJIMA, S.; CHOUCANE, H.; ANTONA-MAKOSHI, J.; UCHIDA, N.; TAJIMA, J. A nationwide impact assessment of automated driving systems on traffic safety using multiagent traffic simulations. **IEEE Open Journal of Intelligent Transportation Systems**, IEEE, v. 3, p. 302–312, 2022. Citado na p. 61.

- KUO, C.; LU, Y.; YANG, S. On the image sensor processing for lane detection and control in vehicle lane keeping systems. **Sensors**, MDPI, v. 19, n. 7, p. 1665, 2019. Citado nas pp. 21, 54, 61.
- LEE, J.; CHOI, J.; YI, K.; SHIN, M.; KO, B. Lane-keeping assistance control algorithm using differential braking to prevent unintended lane departures. **Control Engineering Practice**, Elsevier, v. 23, p. 1–13, 2014. Citado nas pp. 16, 55, 57.
- LEE, K.; LI, S. E.; KUM, D. Synthesis of robust lane keeping systems: impact of controller and design parameters on system performance. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 20, n. 8, p. 3129–3141, 2018. Citado nas pp. 52, 54, 58, 62.
- LI, R.; LI, Y.; LI, S.; ZHANG, C.; BURDET, E.; CHENG, B. Indirect shared control for cooperative driving between driver and automation in steer-by-wire vehicles. **IEEE Transactions On Intelligent Transportation Systems**, v. 22, p. 7826–7836, 2020. Citado na p. 31.
- LI, S.; JIA, Z.; LI, K.; CHENG, B. Fast online computation of a model predictive controller and its application to fuel economy-oriented adaptive cruise control. **IEEE Transactions On Intelligent Transportation Systems**, v. 16, p. 1199–1209, 2014. Citado na p. 59.
- LI, W.; XIE, Z.; ZHAO, J.; GAO, J.; HU, Y.; WONG, P. K. Human-machine shared steering control for vehicle lane keeping systems via a fuzzy observer-based event-triggered method. **IEEE Transactions on Intelligent Transportation Systems**, IEEE, v. 23, n. 8, p. 13731–13744, 2021. Citado na p. 58.
- LIANG, J.; YANG, Y.; ZHU, X.; SHENG, K. Realization of Emergency Lane Keeping System by Adaptive Control based on the Finite State Machine. In: IEEE. 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI). 2021. P. 1–6. Citado nas pp. 15, 25, 52, 53, 61.
- LIANG, Y.; YIN, Z.; NIE, L. Shared steering control for lane keeping and obstacle avoidance based on multi-objective MPC. **Sensors**, v. 21, p. 4671, 2021. Citado nas pp. 59, 62.
- LIAO-MCPHERSON, D.; KOLMANOVSKY, I. F. A. proximally stabilized semismooth algorithm for convex quadratic programming. **Automatica**, v. 113, 2020. Citado na p. 47.
- LITMAN, T. Autonomous vehicle implementation predictions. Victoria Transport Policy Institute Victoria, BC, Canada, 2017. Citado na p. 63.
- LIU, C.; CARVALHO, A.; SCHILDBACH, G.; HEDRICK, J. K. Stochastic predictive control for lane keeping assistance systems using a linear time-varying model. In: IEEE. 2015 American Control Conference (ACC). 2015. P. 3355–3360. Citado nas pp. 55, 56.

- LIU, X.; LIANG, J.; XU, B. A deep learning method for lane changing situation assessment and decision making. **IEEE Access**, IEEE, v. 7, p. 133749–133759, 2019. Citado na p. 55.
- MATWORKS. **MatWorks - mpcActiveSetSolver**. Disponível em: <<https://www.mathworks.com/help/mpc/ref/mpcactivesetsolver.html>>. Acesso em: 3 ago. 2023. Citado na p. 60.
- MERAH, A.; HARTANI, K.; DRAOU, A. A new shared control for lane keeping and road departure prevention. **Vehicle System Dynamics**, Taylor & Francis, v. 54, n. 1, p. 86–101, 2016. Citado na p. 53.
- NGUYEN, A.; CHEVREL, P.; CLAVEAU, F. On the effective use of vehicle sensors for automatic lane keeping via LPV static output feedback control. **IFAC-PapersOnLine**, v. 50, p. 13808–13815, 2017. Citado na p. 63.
- NGUYEN, A.-T.; SENTOUH, C.; POPIEUL, J.-C. Fuzzy steering control for autonomous vehicles under actuator saturation: Design and experiments. **Journal of the Franklin Institute**, Elsevier, v. 355, n. 18, p. 9374–9395, 2018. Citado nas pp. 53, 58.
- NHTSA. **National Highway Traffic Safety Administration-Human Factors in Road Safety**. 2024. Disponível em: <<https://www.nhtsa.gov/research-data/human-factors>>. Citado na p. 15.
- ORGANIZATION, W. H. **Global Status Report on Road Safety 2018**. 2018. Disponível em: <<https://www.who.int/publications/i/item/9789241565684>>. Citado na p. 19.
- PANDALA, A.; DING, Y.; PARK, H. q. A real-time sparse quadratic program solver for robotic applications. **IEEE Robotics And Automation Letters**, v. 4, p. 3355–3362, 2019. Citado na p. 46.
- PARKASH, A.; SWARUP, A. Control of autonomous vehicle for lateral dynamics using sliding mode and input-to state stability methods. In: IEEE. 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE). 2020. P. 1–7. Citado na p. 57.
- PFEFFER, P.; STUBLER, T.; F"ORSTER, M.; NIEGL, M.; ABHISHEK, R. Model-based Calibration of the Lateral Controllers ‘Steer-by-Angle’and ‘Steer-by-Torque’of a Lane Keeping Assistance System. **FISITA 2018 World Automot. Congr**, p. 1459–1468, 2018. Citado nas pp. 16, 60.
- RAJAMANI; RAJESH. Longitudinal control for vehicle platoons. **Vehicle Dynamics and Control**, Springer, p. 171–200, 2012. Citado nas pp. 26, 29, 57.
- RAJAMANI, R. **Vehicle dynamics and control**. Springer Science & Business Media, 2011. Citado nas pp. 52, 56, 57, 76.

- RATHAI, K. M. M.; AMIRTHALINGAM, J.; JAYARAMAN, B. Robust tube-MPC based lane keeping system for autonomous driving vehicles. In: PROCEEDINGS of the Advances in Robotics. 2017. P. 1–6. Citado nas pp. [15](#), [58](#).
- RAWLINGS, J. B.; MAYNE, D. Q.; DIEHL, M. **Model predictive control: theory, computation, and design**. Nob Hill Publishing Madison, WI, 2017. v. 2. Citado na p. [43](#).
- REDDY, N.; FARAH, H.; HUANG, Y.; DEKKER, T.; VAN AREM, B. Operational design domain requirements for improved performance of lane assistance systems: A field test study in The Netherlands. **IEEE Open Journal of Intelligent Transportation Systems**, IEEE, v. 1, p. 237–252, 2020. Citado nas pp. [55](#), [61](#), [63](#).
- RODRIGUES, R.; MURILO, A.; LOPES, R.; DE SOUZA, L. Hardware in the loop simulation for model predictive control applied to satellite attitude control. **IEEE Access**, v. 7, p. 57401–15741, 2019. Citado na p. [40](#).
- ROSEN, H. E.; BARI, I.; PAICHADZE, N.; PEDEN, M.; KHAYESI, M.; MONCLÚS, J.; HYDER, A. A. Global road safety 2010–18: an analysis of global status reports. **Injury**, Elsevier, 2022. Citado na p. [19](#).
- RUI, R. Lane line detection technology based on machine vision. In: IEEE. 2022 4th International Conference on Artificial Intelligence and Advanced Manufacturing (AIAM). 2022. P. 562–566. Citado na p. [55](#).
- SALT DUCAJÚ, J. M.; SALT LLOBREGAT, J. J.; CUENCA, Á.; TOMIZUKA, M. Autonomous ground vehicle lane-keeping LPV model-based control: Dual-rate state estimation and comparison of different real-time control strategies. **Sensors**, MDPI, v. 21, n. 4, p. 1531, 2021. Citado nas pp. [16](#), [57](#), [58](#).
- SAMUEL, M.; MOHAMAD, M.; HUSSEIN, M.; SAAD, S. M. Lane keeping maneuvers using proportional integral derivative (PID) and model predictive control (MPC). **Journal of Robotics and Control (JRC)**, v. 2, n. 2, p. 78–82, 2021. Citado nas pp. [57](#), [58](#).
- SARAÇOĞLU, K.; ÜLEŞ, B.; SCHMIDT, K. W. A lane keeping system with a weighted preview measurement. In: IEEE. 2018 26th Signal Processing and Communications Applications Conference (SIU). 2018. P. 1–4. Citado nas pp. [20](#), [53](#), [58](#).
- SATRIA, M. A.; INDRIAWATI, K.; WIDJANTORO, B. L.; HIJA, A. I.; NURHADI, H. Lane Keeping Control Using Nonlinear Model Predictive Control on Constant Speed Autonomous Car. In: IEEE. 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom). 2022. P. 12–16. Citado na p. [57](#).
- SENTOUH, C.; NGUYEN, A.; BENLOUCIF, M.; POPIEUL, J. Driver-automation cooperation oriented approach for shared control of lane keeping assist systems. **IEEE Transactions On Control Systems Technology**, v. 27, p. 1962–1978, 2018. Citado nas pp. [61](#), [62](#).

- SETIAWAN, J. D.; SAFARUDIN, M.; SINGH, A. Modeling, simulation and validation of 14 DOF full vehicle model. In: IEEE. INTERNATIONAL Conference on Instrumentation, Communication, Information Technology, and Biomedical Engineering 2009. 2009. P. 1–6. Citado na p. 57.
- SHERIF, M. M.; AHMED, A. M.; MOUSTAFA, A. M.; MONESS, M. Optimal control of lane keeping system using simulated annealing and linear quadratic regulator. In: IEEE. 2019 15th International Computer Engineering Conference (ICENCO). 2019. P. 1–5. Citado na p. 58.
- SHU, P.; SAGARA, S.; WANG, Q.; OYA, M. Improved adaptive lane-keeping control for four-wheel steering vehicles without lateral velocity measurements. **International Journal of Robust and Nonlinear Control**, Wiley Online Library, v. 27, n. 17, p. 4154–4168, 2017. Citado na p. 57.
- SON, Y. S.; KIM, W.; LEE, S.-H.; CHUNG, C. C. Robust multirate control scheme with predictive virtual lanes for lane-keeping system of autonomous highway driving. **IEEE Transactions on Vehicular Technology**, IEEE, v. 64, n. 8, p. 3378–3391, 2014. Citado nas pp. 56–58.
- SON, Y. S.; LEE, S.-H.; CHUNG, C. C. Predictive virtual lane using relative motions between a vehicle and lanes. In: IEEE. 2013 IEEE Intelligent Vehicles Symposium (IV). 2013. P. 771–776. Citado nas pp. 15, 54, 61.
- STELLATO, B.; BANJAC, G.; GOULART, P.; BEMPORAD, A.; BOYD, S. O. An operator splitting solver for quadratic programs. **Mathematical Programming Computation**, v. 12, p. 637–672, 2020. Citado na p. 47.
- TIAN, Y.; ZHAO, Y.; SHI, Y.; CAO, X.; YU, D.-L. The indirect shared steering control under double loop structure of driver and automation. **IEEE/CAA Journal of Automatica Sinica**, IEEE, 2020. Citado na p. 59.
- ULSOY, A.; PENG, H. a. **Automotive control systems. (Cambridge University Press.** 2012. Citado na p. 28.
- WANG, H.; WANG, Y.; ZHAO, X.; WANG, G.; HUANG, H.; ZHANG, J. Lane detection of curving road for structural highway with straight-curve model on vision. **IEEE Transactions on Vehicular Technology**, IEEE, v. 68, n. 6, p. 5321–5330, 2019. Citado nas pp. 23, 54, 62.
- WANG, L. **Model predictive control system design and implementation using MATLAB®.** Springer Science & Business Media, 2009. Citado nas pp. 17, 36.
- WEI, S.; PFEFFER, P. E.; EDELMANN, J. State of the Art: Ongoing Research in Assessment Methods for Lane Keeping Assistance Systems. **IEEE Transactions on Intelligent Vehicles**, IEEE, 2023. Citado na p. 62.

- WU, C.-B.; WANG, L.-H.; WANG, K.-C. Ultra-low complexity block-based lane detection and departure warning system. **IEEE Transactions on Circuits and Systems for Video Technology**, IEEE, v. 29, n. 2, p. 582–593, 2018. Citado nas pp. [22](#), [55](#).
- XING, Y.; LV, C.; CHEN, L.; WANG, H.; WANG, H.; CAO, D.; VELENIS, E.; WANG, F.-Y. Advances in vision-based lane detection: Algorithms, integration, assessment, and perspectives on ACP-based parallel vision. **IEEE/CAA Journal of Automatica Sinica**, IEEE, v. 5, n. 3, p. 645–661, 2018. Citado na p. [25](#).
- XU, L. H.; HU, S. G.; LUO, Q. et al. A new lane departure warning algorithm considering the driver's behavior characteristics. **Mathematical Problems in Engineering**, Hindawi, v. 2015, 2015. Citado nas pp. [19](#), [53](#), [62](#).
- XU, S.; PENG, H.; LU, P.; ZHU, M.; TANG, Y. Design and experiments of safeguard protected preview lane keeping control for autonomous vehicles. **IEEE Access**, IEEE, v. 8, p. 29944–29953, 2020. Citado na p. [55](#).
- XU, X.; GRIZZLE, J. W.; TABUADA, P.; AMES, A. D. Correctness guarantees for the composition of lane keeping and adaptive cruise control. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 15, n. 3, p. 1216–1229, 2017. Citado na p. [52](#).
- XU, X.; WATERS, T.; PICKEM, D.; GLOTFELTER, P.; EGERSTEDT, M.; TABUADA, P.; GRIZZLE, J. W.; AMES, A. D. Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds. In: IEEE. 2017 IEEE conference on control technology and applications (CCTA). 2017. P. 1769–1775. Citado na p. [20](#).
- YAKSHA, S.; AKASH, E.; SANJAY, U.; SHETTY, S.; BABALI, B. Lateral and longitudinal control of self-driving car with enhanced trajectory tracking using direct yaw stability control. In: IEEE. 2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT). 2021. P. 1–10. Citado nas pp. [56](#), [57](#), [60](#).
- YAN, M.; CHEN, W.; WANG, Q.; ZHAO, L.; LIANG, X.; CAI, B. Human–Machine Cooperative Control of Intelligent Vehicles for Lane Keeping—Considering Safety of the Intended Functionality. In: MDPI, 9. ACTUATORS. 2021. v. 10, p. 210. Citado nas pp. [21](#), [61](#).
- YAN, N.; LIU, C.; SUN, Q. A novel incremental control method based on h-infinity control for lane keeping assist. In: IEEE. 2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI). 2021. P. 1–6. Citado nas pp. [53](#), [58](#).
- YANG, J.; KIM, S.; HUH, K. Development of a unified lane-keeping and collision avoidance system for semi-trailer truck. **IEEE Access**, IEEE, v. 8, p. 149751–149763, 2020. Citado nas pp. [56](#), [58](#), [59](#), [62](#).

YOFFIE, D. B. Mobileye: The future of driverless cars. **Harvard Business School Case**, Business History, p. 715–421, 2014. Citado na p. [55](#).