

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Simulação e Controle Preditivo de um Ciclista Virtual

Vítor Agostinho Mouro

JUIZ DE FORA
SETEMBRO, 2024

Simulação e Controle Preditivo de um Ciclista Virtual

VÍTOR AGOSTINHO MOURO

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Orientador: Marcelo Caniato Renhe

JUIZ DE FORA
SETEMBRO, 2024

SIMULAÇÃO E CONTROLE PREDITIVO DE UM CICLISTA VIRTUAL

Vítor Agostinho Mouro

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DO INSTITUTO DE CIÊNCIAS
EXATAS DA UNIVERSIDADE FEDERAL DE JUIZ DE FORA, COMO PARTE INTE-
GRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
BACHAREL EM CIÊNCIA DA COMPUTAÇÃO.

Aprovada por:

Marcelo Caniato Renhe
Doutor em Engenharia de Sistemas e Computação

Heder Soares Bernardino
Doutor em Modelagem Computacional

Rodrigo Luis de Souza da Silva
Doutor em Engenharia

JUIZ DE FORA
06 DE SETEMBRO, 2024

Resumo

A complexidade inerente aos veículos de duas rodas, que dependem de equilíbrio contínuo e orientação precisa para garantir sua estabilidade, é um dos principais desafios abordados neste trabalho. Para enfrentar essa dificuldade, foi desenvolvido um controlador em tempo real utilizando o método de Controle Preditivo Baseado em Modelo (MPC), que permite prever e ajustar o comportamento do veículo em tempo real, com base em um modelo. A simulação física do sistema é realizada por meio do motor MuJoCo, que permite modelar as interações dinâmicas do veículo, proporcionando um ambiente para testar e validar as estratégias de controle propostas. O controlador proposto visa alcançar estabilidade e eficiência em cenários simulados, com um enfoque especial na aplicação prática para jogos eletrônicos. Além disso, este estudo busca contribuir para a área de controle de veículos de duas rodas, explorando a viabilidade e os desafios do MPC em ambientes de simulação complexos.

Palavras-chave: Controle Preditivo Baseado em Modelo, veículos de duas rodas, simulação física, MuJoCo.

Abstract

The inherent complexity of two-wheeled vehicles, which rely on continuous balance and precise orientation to ensure their stability, is one of the main challenges addressed in this work. To tackle this difficulty, a real-time controller was developed using the Model Predictive Control (MPC) method, which allows for predicting and adjusting the vehicle's behavior in real time, based on a model. The physical simulation of the system is carried out using the MuJoCo engine, which models the dynamic interactions of the vehicle, providing a environment for testing and validating the proposed control strategies. The proposed controller aims to achieve stability and efficiency in simulated scenarios, with a particular focus on practical application for electronic games. Furthermore, this study seeks to contribute to the field of two-wheeled vehicle control by exploring the feasibility and challenges of MPC in complex simulation environments.

Keywords: Model Predictive Control, two-wheeled vehicles, simulation, MuJoCo.

Agradecimentos

Gostaria de expressar minha profunda gratidão aos meus pais, pelo apoio incondicional e encorajamento durante toda a minha jornada acadêmica. Agradeço também ao professor Marcelo Caniato Renhe pela orientação e paciência durante o desenvolvimento deste trabalho, bem como aos professores do Departamento de Ciência da Computação, cujos ensinamentos foram fundamentais para minha formação. Agradeço ainda aos meus amigos e colegas, que compartilharam comigo momentos de desafio e aprendizado ao longo desses anos. Por fim, dedico este trabalho a todos que, de alguma forma, contribuíram para a minha trajetória.

Conteúdo

Lista de Figuras	6
Lista de Tabelas	8
Lista de Abreviações	9
1 Introdução	10
1.1 Justificativa	11
1.2 Questões de pesquisa	11
1.3 Objetivos	12
1.4 Materiais e Métodos	13
1.5 Organização do Trabalho	14
2 Fundamentação	15
2.1 Dinâmica dos veículos de duas rodas	15
2.1.1 Geometria	17
2.2 Motores de física	18
2.2.1 MuJoCo (<i>Multi-Joint dynamics with Contact</i>)	19
2.3 Teoria de controle	20
2.3.1 Otimização de trajetória	21
2.3.2 Controle Preditivo Baseado em Modelo (MPC)	21
2.4 Considerações Finais	23
3 Revisão bibliográfica	24
3.1 Trabalhos Relacionados	24
3.1.1 Dinâmica Inversa	25
3.1.2 Aprendizado Por Reforço	26
3.1.3 Redes Neurais	29
3.1.4 Controle Ótimo	30
3.2 Análise Comparativa	31
3.3 Considerações Finais	32
4 Implementação	34
4.1 Modelo do humanóide	34
4.2 Modelo da bicicleta	35
4.3 Modelo final	37
4.4 Controlador	39
4.4.1 Rota	39
4.4.2 Função de custo	40
4.4.3 Algoritmo de otimização	42
4.5 Cenários de teste	43
4.5.1 Linha reta em terreno plano	43
4.5.2 Zigue-zague em terreno plano	44
4.5.3 Escadas	45
4.5.4 Linha reta em terreno acidentado	45

4.5.5	Circuito de obstáculos	46
4.6	Métricas de avaliação	47
5	Resultados	48
5.1	Configuração da máquina	48
5.2	Cenário de linha reta	48
5.3	Cenário de zigue-zague	51
5.4	Cenário de terreno acidentado	54
5.5	Cenário de escadaria	57
5.6	Cenário de circuito de obstáculos	60
5.7	Discussão dos resultados	63
6	Conclusão	66
	Bibliografia	68

Lista de Figuras

2.2	Principais medidas da geometria de uma bicicleta. Fonte: (WILSON; SCHMIDT, 2020).	17
2.3	Diagrama do fluxo de controle fechado. Fonte: do autor.	20
3.1	Modelo de bicicleta Whipple-Carvalho: quatro corpos rígidos (roda traseira, quadro traseiro, conjunto do guidão, roda dianteira) conectados por três juntas rotativas (cubo traseiro, eixo de direção, cubo dianteiro). Fonte: (SCHWAB; MEIJAARD, 2013).	25
4.1	Modelo do humanoíde incluído na biblioteca MuJoCo. Fonte: do autor.	35
4.2	Modelo da bicicleta desenvolvido. O ponto em vermelho é a localização arbitrária na bicicleta escolhida para computar a distância da rota. Fonte: do autor.	36
4.3	Modelo final da bicicleta e humanoide. Fonte: do autor.	38
4.4	Posições das restrições de igualidade do modelo. Em vermelho, os pontos atribuídos à bicicleta, e em azul os pontos atribuídos ao humanoide. Fonte: do autor.	39
4.5	Exemplo de rota gerada para um dos cenários de teste. Os pontos em verde são os pontos de ancoragem da rota, os pontos em azul são os pontos de controle, e em roxo a rota final gerada pelo <i>spline</i> cúbico. Fonte: do autor.	40
4.6	Cenário de linha reta em terreno plano. Fonte: do autor.	44
4.7	Cenário de zigue-zague em terreno plano. Fonte: do autor.	44
4.8	Cenário de descida de escadaria. Fonte: do autor.	45
4.9	Cenário de linha reta em terreno acidentado. A diferença de altura entre o ponto mais baixo e mais alto do terreno é 40 cm. Fonte: do autor.	46
4.10	Cenário de circuito de obstáculos. Fonte: do autor.	46
5.1	Resultados para o cenário de linha reta. Fonte: do autor.	49
5.2	Trajetórias geradas para o cenário de linha reta, agrupadas por velocidade alvo. Fonte: do autor.	51
5.3	Mapa de calor para a taxa de sucesso média no cenário de linha reta. Fonte: do autor.	52
5.4	Resultados para o cenário de zigue-zague. Fonte: do autor.	52
5.5	Trajetórias geradas para o cenário de zigzag, agrupadas por velocidade alvo. Fonte: do autor.	54
5.6	Mapa de calor para a taxa de sucesso média no cenário de zigue-zague. Fonte: do autor.	54
5.7	Resultados para o cenário de terreno acidentado. Fonte: do autor.	55
5.8	Trajetórias geradas para o cenário de terreno acidentado, agrupadas por velocidade alvo. Fonte: do autor.	56
5.9	Mapa de calor para a taxa de sucesso média no cenário de terreno acidentado. Fonte: do autor.	57
5.10	Resultados para o cenário de escadaria. Fonte: do autor.	58
5.11	Trajetórias geradas para o cenário de escadaria, agrupadas por velocidade alvo. Fonte: do autor.	59

5.12	Mapa de calor para a taxa de sucesso média no cenário de escadaria. Fonte: do autor.	60
5.13	Visão lateral das trajetórias geradas para o cenário de escadaria. Fonte: do autor.	60
5.14	Resultados para o cenário de circuito de obstáculos. Fonte: do autor. . . .	61
5.15	Trajeto�rias geradas para o cen�rio de circuito de obst�culos, agrupadas por velocidade alvo. Fonte: do autor.	62
5.16	Vis�o lateral das trajet�rias geradas para o cen�rio de circuito de obst�culos. Fonte: do autor.	63
5.17	Mapa de calor para a taxa de sucesso m�dia no cen�rio de circuito de obst�culos. Fonte: do autor.	63
5.18	Taxa de sucesso para cada cen�rio de teste. Fonte: do autor.	64

Lista de Tabelas

3.1	Listagem dos trabalhos relacionados.	31
3.2	Tabela comparativa entre os trabalhos relacionados.	31
4.1	Tabela de Controle das Articulações	35
5.1	Valores testados para cada parâmetro.	48
5.2	Tabela de resultados agregados do cenário de linha reta - Simulação tempo real.	50
5.3	Tabela de resultados agregados do cenário de zigue-zague - Simulação tempo real.	53
5.4	Tabela de resultados agregados do cenário de terreno acidentado - Simulação tempo real.	56
5.5	Tabela de resultados agregados do cenário de escadaria - Simulação tempo real.	59
5.6	Tabela de resultados agregados do cenário de circuito - Simulação tempo real.	61

Lista de Abreviações

MPC	Model Predictive Control
MuJoCo	Multi-Joint dynamics with Contact
NEAT	Neuroevolution of Augmenting Topologies
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
DRL	Deep Reinforcement Learning
PID	Proportional-integral-derivative

1 Introdução

Veículos de duas rodas em linha, como bicicletas e motocicletas, apresentam desafios únicos em termos de estabilidade e controle devido à sua configuração. Quando em baixa velocidade, esses veículos não possuem estabilidade lateral, exigindo que o conjunto do guidão seja rotacionado corretamente para evitar quedas. Além disso, a navegação de tais veículos até um objetivo específico requer a resolução simultânea de dois problemas críticos: equilíbrio dinâmico e orientação precisa (WHIPPLE, 1899).

O desafio se intensifica ao considerar um piloto virtual, cujo corpo tem massa e movimentos que simulam um ser humano real. Esse piloto virtual deve não apenas manter o equilíbrio do veículo, mas também interagir de maneira realista com a sua dinâmica, influenciando sua movimentação e sua resposta a diferentes condições de condução (SCHWAB; MEIJAARD, 2013). O piloto nesse caso é um componente essencial do veículo, controlando-o por meio de aplicações precisas de torque no guidão e de ajustes no seu posicionamento, que, como será visto no decorrer desse trabalho, impacta consideravelmente a dinâmica do sistema.

A pesquisa sobre o controle de veículos de duas rodas tem uma vasta gama de aplicações práticas que a tornam relevante. Em particular, veículos autônomos de duas rodas representam uma área emergente, prometendo soluções eficientes e sustentáveis para mobilidade urbana (SANCHEZ; PASTOR; LARSON, 2020). Além disso, o uso de simulações avançadas permite um *design* orientado por simulação, otimizando o desenvolvimento de novos veículos e sistemas de controle antes de serem testados fisicamente, como em Picotti, Bianchin e Bruschetta (2024). No campo dos jogos eletrônicos, a criação de simulações realistas de pilotagem aprimora a experiência do usuário e a fidelidade do jogo (MIKKOLA, 2024).

A escolha do método Controle Preditivo Baseado em Modelo (MPC) (TODO-ROV; EREZ; TASSA, 2012) para este trabalho se deu principalmente pela vantagem do MPC em relação a métodos baseados em aprendizado, como o aprendizado por reforço, de não requerer um processo de treinamento extenso e bem formulado, que pode ser de-

morado (RANDLØV; ALSTRØM, 1998). O MPC utiliza um modelo explícito do sistema para prever e otimizar o comportamento futuro, permitindo ajustes em tempo real.

Neste trabalho serão explorados os passos necessários para se modelar o veículo utilizando um motor de física para corpos rígidos, incluindo um piloto humanoide. O MPC será então utilizado para controlar o agente em um ambiente de simulação, buscando uma solução integrada que permita um controle estável e eficaz. A complexidade deste problema reside na necessidade de coordenar movimentos sutis e ajustes constantes para garantir a estabilidade e a orientação correta, refletindo as nuances da pilotagem real.

1.1 Justificativa

O Controle Preditivo Baseado em Modelo (MPC) se destaca como uma metodologia promissora para enfrentar esses desafios, por sua capacidade de prever o comportamento futuro do sistema e ajustar as ações de controle em tempo real. A aplicação do MPC em veículos de duas rodas permite explorar a viabilidade de soluções de controle que sejam simultaneamente robustas e eficientes, adaptando-se às mudanças dinâmicas do ambiente e do sistema. Além disso, a recente incorporação do suporte ao MPC no motor de física MuJoCo (TODOROV; EREZ; TASSA, 2012; HOWELL et al., 2022), amplamente utilizado para simulações de robótica, abre novas possibilidades para a implementação e experimentação de controle preditivo em ambientes de simulação e oferece uma plataforma robusta para controle usando MPC.

A justificativa para este estudo reside, portanto, na necessidade de desenvolver soluções de controle avançadas que possam ser aplicadas a veículos de duas rodas, tanto no contexto de veículos autônomos quanto em simulações. O aprofundamento teórico e a validação prática do MPC neste contexto têm o potencial de contribuir significativamente para a área de controle de sistemas dinâmicos, oferecendo novas perspectivas e soluções para um problema ainda pouco explorado na literatura.

1.2 Questões de pesquisa

Dentre as questões que esse trabalho busca responder, se destacam:

- Quais os desafios na aplicação do método MPC para problemas de controle como este, que envolvem dinâmicas complexas e muitos graus de liberdade?
- É possível que uma movimentação natural do piloto se manifeste sem usar dados de captura de movimento?
- Quão robusto e eficiente o controlador é em diferentes cenários de teste?
- A eficiência do controlador sobre a simulação em tempo real será suficiente para possíveis aplicações práticas do método?

Essas questões ainda não foram respondidas de forma satisfatória na literatura, o que justifica a realização deste estudo e a busca por soluções eficazes para o controle de veículos de duas rodas.

1.3 Objetivos

O objetivo principal deste trabalho é desenvolver um controlador em tempo real, baseado no método MPC, para um veículo de duas rodas simulado fisicamente, incluindo a modelagem de um piloto humanoide que interage dinamicamente com o sistema. Esse objetivo é sustentado por uma série de objetivos secundários que visam garantir a eficácia e a robustez do controlador em diferentes cenários de simulação. São elas:

- Desenvolver modelos precisos tanto do veículo quanto do piloto humanoide, levando em consideração as características físicas, como massa, geometria, atrito e torque, que influenciam diretamente na estabilidade e na capacidade de manobra do sistema;
- Implementar o controlador MPC de forma que ele seja capaz de ajustar continuamente as variáveis de controle com base no estado atual do sistema, visando a manutenção do equilíbrio e o cumprimento de trajetórias específicas;
- Avaliar o desempenho do controlador em uma série de cenários simulados que reproduzam condições variadas de terreno, obstáculos, e diferentes velocidades de movimento, assegurando que o controlador seja capaz de lidar com situações reais;

1.4 Materiais e Métodos

Para o desenvolvimento deste trabalho, foi utilizada uma abordagem baseada em simulação para avaliar o desempenho de controladores aplicados a veículos de duas rodas. A metodologia foi dividida em três etapas principais: modelagem do veículo e do piloto, implementação do controlador e simulação dos cenários de teste.

Na etapa de **modelagem do veículo e do piloto**, o veículo de duas rodas foi modelado utilizando o motor de física MuJoCo. Esse modelo do veículo inclui todas as características físicas relevantes, como a geometria e os parâmetros de massa, atrito e torque, com base em dados disponíveis para esses veículos. Além disso, foi incluído um piloto humanoide virtual que interage de maneira realista com o veículo, influenciando sua dinâmica por meio da movimentação de seus membros, modificado a partir do modelo de humanoide já disponível na MuJoCo e utilizado na literatura.

Em seguida, foi feita a implementação do controlador, utilizando a estratégia de Controle Preditivo Baseado em Modelo (MPC). O MPC opera em tempo real, ajustando continuamente as variáveis de controle para manter a estabilidade e garantir a execução da trajetória desejada. A implementação do MPC foi facilitada pelo suporte nativo do MuJoCo a este tipo de controle, permitindo uma integração direta com o ambiente de simulação.

Finalmente, a etapa de simulação dos cenários de teste consistiu na construção e simulação de cenários projetados para avaliar a robustez e a eficiência do controlador. Esses cenários incluem diferentes condições de terreno, obstáculos comuns para veículos de duas rodas, e variações de velocidade. Os testes foram executados diversas vezes para avaliar o desempenho médio do controlador.

Essas etapas foram pensadas para permitir uma análise do comportamento do controlador em condições que aproximam situações reais, fornecendo dados valiosos sobre a aplicabilidade do MPC em veículos de duas rodas e suas possíveis melhorias.

1.5 Organização do Trabalho

Este trabalho está estruturado em capítulos que apresentam de forma lógica e sequencial os elementos necessários para a compreensão e desenvolvimento da pesquisa. No Capítulo 2, é desenvolvida a Fundamentação Teórica, que aborda os conceitos fundamentais relacionados à dinâmica de veículos de duas rodas, motores de física e teoria de controle. Este capítulo também explora em profundidade o método MPC, suas características, vantagens e limitações, preparando o terreno para a implementação prática no contexto deste trabalho.

O Capítulo 3 é dedicado à Revisão Bibliográfica, onde são discutidos os principais trabalhos relacionados à estabilização e controle de veículos de duas rodas. A análise comparativa desses estudos permite situar este trabalho no estado da arte, destacando suas contribuições e limitações.

No Capítulo 4, a Implementação é descrita em detalhes. Este capítulo apresenta os modelos desenvolvidos, incluindo o piloto humanoide e o veículo de duas rodas, além das estratégias de controle adotadas. A integração desses elementos no ambiente de simulação do MuJoCo também é discutida, destacando as decisões técnicas e adaptações realizadas.

O Capítulo 5 é dedicado aos Resultados, onde são apresentados os cenários de teste, as métricas de avaliação e as análises obtidas a partir das simulações. Este capítulo avalia o desempenho do controlador MPC em diferentes condições, discutindo os desafios enfrentados e os resultados alcançados.

Por fim, no Capítulo 6, são apresentadas as Considerações Finais, que revisitam os objetivos do trabalho, sintetizam as principais contribuições e destacam as possibilidades de trabalhos futuros. Este capítulo reflete sobre os avanços obtidos com o desenvolvimento e simulação do controlador MPC, bem como sobre as limitações observadas e as melhorias que podem ser implementadas em estudos subsequentes.

2 Fundamentação

Neste capítulo, são explorados alguns dos conceitos essenciais para compreender a aplicação do método MPC para veículos de duas rodas. Inicialmente, é discutida a dinâmica dos veículos de duas rodas na Seção 2.1, apresentando um panorama geral que auxilia a compreender o motivo por trás deste trabalho. Em seguida, aborda-se o papel dos motores de física ao longo da Seção 2.2, que são uma peça central neste trabalho. Na Seção 2.3, conceitos importantes do campo da teoria de controle serão apresentados, com foco na otimização de trajetória e no método MPC, que são objetos de estudo deste trabalho.

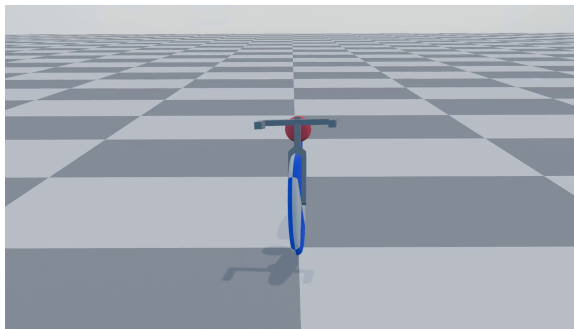
2.1 Dinâmica dos veículos de duas rodas

Semelhante a um carro, um veículo de duas rodas tende a seguir a direção da roda dianteira. Porém, em um veículo de duas rodas o piloto não consegue iniciar uma curva imediatamente. Quando o conjunto de direção do veículo é rotacionado, a trajetória do veículo é imediatamente alterada, mas a inércia e a força centrífuga fazem com que o piloto seja “deixado para trás” nesse movimento. Isso faz com que o veículo adquira uma inclinação no sentido oposto à curva, o que não é uma situação estável e conduziria a uma queda.

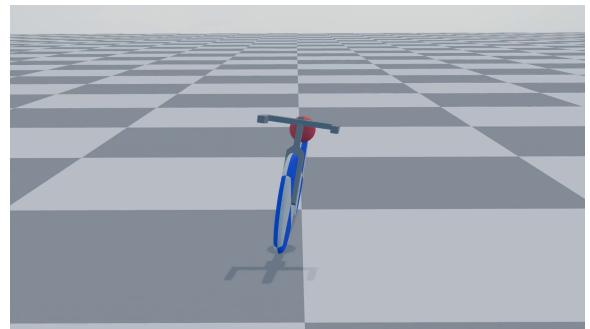
Portanto, fica claro que para realizar uma curva de forma estável é preciso balancear esse fenômeno de alguma forma. Uma boa maneira de entender o porquê disso ser necessário é analisar uma brincadeira comum onde tal fenômeno também ocorre: quando tentamos balancear uma vassoura na mão. O que se pode perceber nessa brincadeira é que, se tentamos mover a mão com a vassoura completamente na vertical, a parte superior fica para trás e ela tomba na direção oposta do movimento. Porém, enquanto crianças aprendemos rapidamente que se deixarmos a vassoura tombar levemente (ou forçar isso movendo a mão no sentido oposto), passamos a conseguir mover a mão nessa direção sem derrubar a vassoura (WILSON; SCHMIDT, 2020, p. 354-355).

A mesma coisa acontece nos veículos de duas rodas. Antes de iniciar uma curva,

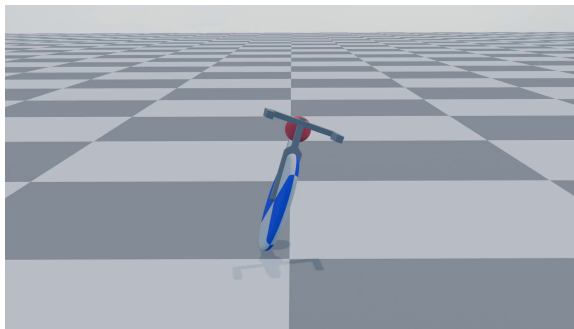
o piloto deve inclinar o veículo no sentido da curva. Dessa forma, é possível cancelar as forças laterais e permanecer estável na trajetória. Isso significa que, para virar para a esquerda, por exemplo, o piloto normalmente precisa girar o guidão para a direita, o que fará com que ele rotacione para a esquerda e possa continuar na curva. Esse fenômeno é chamado de contra-esterço e está ilustrado na Figura 2.1. O veículo também pode adquirir essa rotação quando o piloto posiciona seu corpo para o lado que quer virar, porém isso normalmente é feito em conjunto com a atuação do guidão, já que essa inclinação é lenta e não suficiente para trajetórias acentuadas (WILSON; SCHMIDT, 2020, p. 356-357).



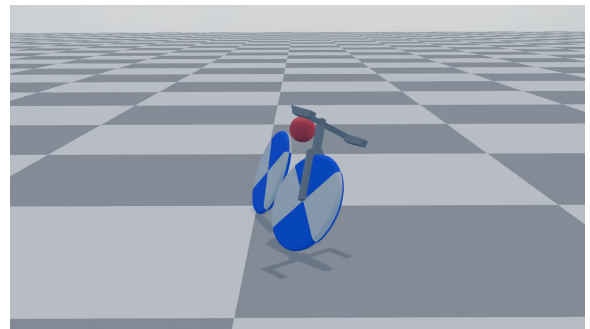
(a) Rotação do guidão para a direita.



(b) Inclinação da bicicleta para a esquerda.



(c) Geometria da bicicleta faz o guidão virar para o lado da inclinação.



(d) Veículo segue na trajetória de curva.

Figura 2.1: Simulação de condução de uma bicicleta por meio de contraesterço. Fonte: do autor.

Essa característica dos veículos de duas rodas é um dos fatores que mais dificultam o controle dessa classe de veículo. Ela implica que para realizar uma curva, o piloto deve considerar não somente a rotação do guidão, mas também como essa rotação afeta a inclinação do veículo e por sua vez o equilíbrio. Felizmente, a bicicleta moderna possui características em sua construção que a tornam mais fácil de ser pilotada, e em algumas situações ela é auto estável, o que significa que ela possui uma tendência de se equilibrar naturalmente quando em movimento. Esse fato, alinhado à capacidade do cérebro humano

de se adaptar rapidamente, tornam o ciclismo uma atividade acessível e possível de ser aprendida bem cedo, pouco tempo depois que aprendemos a caminhar, por exemplo, e mesmo sem conhecer as nuances de seu funcionamento.

Mesmo assim, o controle desse tipo de veículo é complexo, já que as dinâmicas são alteradas significativamente a depender de fatores como características físicas do veículo (massa, geometria, centro de gravidade, pneus, etc.), do piloto (dimensões, massa, posicionamento, etc.), do conjunto veículo-piloto (velocidade, arrasto, proporção da massa total, etc.) e do ambiente (relevo, atrito, obstáculos, etc.) (WILSON; SCHMIDT, 2020, p. 349-387).

2.1.1 Geometria

A geometria de um veículo de duas rodas é um conjunto de medidas que desempenham um papel fundamental na sua estabilidade e manobrabilidade e definem a configuração de um veículo. Dentre elas, o ângulo de inclinação da direção, a distância entre eixos, o deslocamento da suspensão dianteira e o *trail* são consideradas as mais importantes (WILSON; SCHMIDT, 2020, p. 363) e estão ilustradas na Figura 2.2.

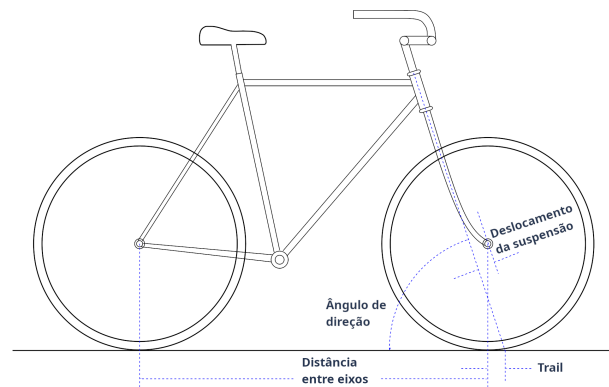


Figura 2.2: Principais medidas da geometria de uma bicicleta. Fonte: (WILSON; SCHMIDT, 2020).

A distância entre eixos, ou *wheelbase*, é a distância entre os centros das rodas dianteira e traseira. Uma maior distância entre eixos geralmente resulta em maior estabilidade, especialmente em alta velocidade, mas pode reduzir o raio de giro e a agilidade do veículo. Já o ângulo de direção, também conhecido como *rake*, é o ângulo entre a linha da coluna de direção e uma linha vertical perpendicular ao solo. Esse ângulo afeta a estabilidade direcional do veículo. Um maior ângulo de inclinação tende a proporcionar

maior estabilidade em linha reta, enquanto um menor ângulo pode tornar o veículo mais ágil em curvas.

O deslocamento da suspensão dianteira, ou *fork offset*, é a distância entre o eixo da direção e o eixo da roda dianteira. Ele impacta diretamente a estabilidade do veículo, já que um maior deslocamento da suspensão dianteira gera um *trail* maior. O *trail* é a distância entre o ponto de contato da roda dianteira com o solo e o ponto onde o eixo da direção intersecta o solo. Quanto mais *trail* uma bicicleta tradicional tem, mais estável ela parece ser, embora um excesso de *trail* possa fazer com que ela seja difícil de pilotar. Bicicletas com *trail* negativo (onde o ponto de contato está à frente de onde o eixo de direção se intersecta com o solo), embora ainda sejam pilotáveis, são instáveis.

A geometria do veículo de duas rodas influencia diretamente o comportamento do piloto e as estratégias de controle. Por exemplo, a necessidade de contraesterço em alta velocidade é um resultado direto da interação entre o ângulo de direção, o *trail* e as forças centrífugas. O ajuste fino desses parâmetros é essencial para otimizar o desempenho do veículo para diferentes condições de condução, como velocidade, terreno e estilo de pilotagem (WILSON; SCHMIDT, 2020, p. 364).

2.2 Motores de física

Motores de física são ferramentas essenciais na simulação de sistemas dinâmicos, permitindo a replicação de comportamentos físicos complexos de forma precisa e eficiente. Eles são amplamente utilizados em áreas como robótica, jogos, realidade virtual, computação gráfica, arquitetura e muitas outras. Esses motores proporcionam um ambiente virtual onde leis físicas, como a conservação de energia, dinâmica de corpos rígidos e interação entre diferentes materiais, são simuladas de forma realista (KAUP et al., 2024).

A escolha de um motor de física adequado é crucial para a fidelidade da simulação e para os resultados dos experimentos. Para veículos de duas rodas, onde a dinâmica é complexa e requer precisão nas interações entre os componentes e o ambiente, a seleção do motor influencia diretamente a qualidade dos resultados.

2.2.1 MuJoCo (*Multi-Joint dynamics with Contact*)

O MuJoCo é um motor de física altamente eficiente e preciso, desenvolvido especificamente para simulação de corpos rígidos com múltiplas articulações (TODOROV; EREZ; TASSA, 2012). Ele é amplamente utilizado em pesquisas de robótica, animação e visualização científica devido à sua capacidade de simular sistemas complexos com alta fidelidade e bom desempenho.

A escolha do MuJoCo para esse trabalho foi devido a diversas características de projeto que o tornam interessante para o problema tratado nesse trabalho. Primeiramente, ele é capaz de lidar com simulações de alta complexidade, envolvendo múltiplos corpos articulados e contatos dinâmicos, o que é essencial para representar a interação entre o veículo de duas rodas e o piloto humanoide, que consistirão de dezenas de corpos rígidos. Além disso, ele possui suporte nativo para otimização e controle usando MPC, o que facilita a implementação do controlador.

Outra vantagem é sua eficiência computacional. Ele é projetado para simulações realistas com alta performance, sendo capaz de simular certos modelos com muita rapidez. Como será visto nas seções seguintes, a eficácia do método MPC depende fortemente da capacidade de avaliar o máximo de passos possíveis de um modelo e, portanto, quantas simulações em paralelo são possíveis de serem executadas.

Uma das principais características do MuJoCo que o difere dos demais é a implementação do modelo de contato entre os corpos. O MuJoCo é um motor de física de corpos rígidos, o que significa que os objetos simulados são considerados como sendo indeformáveis. Portanto, uma de suas principais responsabilidades é tratar as colisões entre os corpos, garantindo que elas respeitam as leis físicas e geram resultados realistas. A maioria dos motores de física usados para aplicações de robótica gasta muito tempo de processamento nessa etapa, para garantir resultados precisos. O MuJoCo, porém, usa uma formulação de contatos diferente, que relaxa a restrição de interseção dos corpos e permite parametrizar forças de repulsão manualmente. Essa formulação é significativamente mais rápida de ser computada e acompanha uma característica que é desejada: como no mundo real todos os materiais têm algum nível de deformação, é possível parametrizar os corpos físicos de acordo e alcançar resultados fiéis. Do contrário, seria necessário usar

múltiplos corpos físicos conectados para representar um único corpo deformável, que apesar de funcional, é uma técnica que aumenta consideravelmente o custo computacional. Essa diferença foi um dos grandes motivos da escolha do MuJoCo em oposição a outros motores de física similares, como o Bullet Physics¹ e o NVIDIA PhysX², já que é esperado que a diferença de performance seja significativa.

2.3 Teoria de controle

O campo da teoria de controle trata do controle de sistemas dinâmicos com o objetivo de desenvolver um modelo ou algoritmo que utilize as entradas do sistema para levá-lo a um estado desejado. Normalmente, esse estado deve ser atingido enquanto certas métricas são minimizadas em algum grau de otimalidade (HOWELL et al., 2022).

O componente capaz de realizar essa tarefa é chamado **controlador**, que deve produzir **ações** a serem passadas ao sistema a fim de atingir o objetivo. Isso pode ocorrer de duas formas. Um controlador de fluxo aberto opera sem utilizar a retroalimentação (*feedback*) para ajustar suas ações. Nesse sistema, a saída não é medida ou comparada com o estado desejado, e as ações de controle são baseadas apenas em entradas determinadas ou em um modelo fixo do sistema. Já um controlador de fluxo fechado utiliza retroalimentação para ajustar continuamente suas ações com base na diferença entre o estado atual e o estado desejado, conforme ilustrado na Figura 2.3. Este tipo de sistema mede a saída e a compara com o estado desejado, ajustando as ações de controle conforme necessário para corrigir erros.

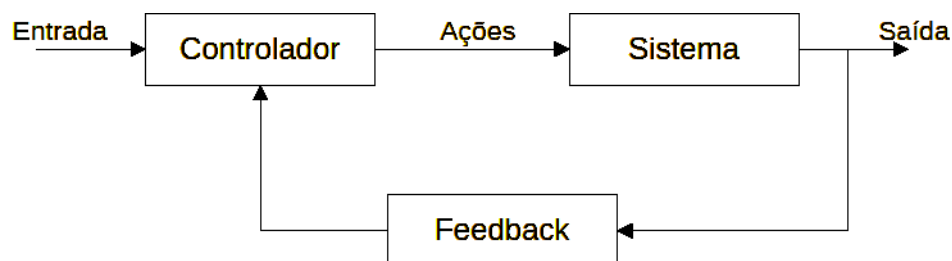


Figura 2.3: Diagrama do fluxo de controle fechado. Fonte: do autor.

¹<https://pybullet.org>

²<https://developer.nvidia.com/physx-sdk>

2.3.1 Otimização de trajetória

Nesse contexto, a otimização de trajetória consiste em solucionar o problema de controle ótimo para um sistema, dado um intervalo de tempo finito (HOWELL et al., 2022). O sistema é considerado governado pela dinâmica discreta

$$s_{t+1} = f(s_t, a_t), \quad (2.1)$$

onde o novo estado s_{t+1} é uma função do estado anterior s_t e do conjunto de ações tomadas a_t . O problema é então apresentado como:

$$\min \sum_{t=0}^T c(s_t, a_t), \quad (2.2)$$

onde T é o horizonte de otimização e $c(s_t, a_t)$ é uma função de custo definida para todo o intervalo $[0, T]$.

Uma separação comum entre os métodos de otimização de trajetória é entre os métodos diretos e os métodos *shooting* (HOWELL et al., 2022). Os métodos diretos possuem tanto os estados quanto os controles como variáveis de decisão, e as dinâmicas são impostas por meio de restrições. Isso faz com que seja possível representar trajetórias que não são realizáveis fisicamente, por exemplo, para fixar o estado final sem saber como atingi-lo. Já nos métodos *shooting-based*, somente os controles são variáveis de decisão, e as dinâmicas são impostas pela simulação. Portanto, somente trajetórias fisicamente realizáveis são consideradas, o que reduz o espaço de busca e alivia o otimizador de precisar garantir as restrições.

2.3.2 Controle Preditivo Baseado em Modelo (MPC)

O método MPC é um caso específico de otimização de trajetória, onde somente o primeiro passo da solução encontrada é aplicado e o processo é repetido indefinidamente para próximos horizontes. Isso permite ao método ser utilizado em aplicações de tempo real, reagindo às alterações das dinâmicas do sistema (HOWELL et al., 2022).

Essa característica é conhecida como horizonte recuante: o algoritmo é míope, no sentido que só otimiza para o horizonte atual e não considera os custos posteriores a esse

intervalo. Dessa forma, esse método não garante a otimalidade da solução encontrada, mas na prática possui bons resultados para problemas de controle que não envolvem tarefas muito longas.

O algoritmo básico do MPC consiste de três elementos básicos: um modelo (representação interna) do sistema, uma função objetivo e um algoritmo de otimização que manipule as variáveis de controle. No contexto desse trabalho, os modelos são instâncias do simulador que o planejador utiliza para avaliar as soluções, e uma instância é eleita como “verdadeira”, representando um sistema real. A função objetivo, normalmente de minimização, precisa ser definida para todo o intervalo do horizonte de otimização. Um exemplo de função objetivo quadrática é a dada por

$$J = \sum_{i=1}^N w_{x_i} (r_i - x_i)^2, \quad (2.3)$$

onde r_i é o valor de referência para a variável observada x_i , sendo w_{x_i} o coeficiente de importância relativa da variável. A função é portanto um somatório de N termos ponderados, um para cada variável de controle.

Por fim, um algoritmo de otimização deve ser empregado para encontrar uma trajetória que minimize a função objetivo. Um exemplo de tal algoritmo é a **amostragem preditiva**. A amostragem preditiva é um método simples e baseado em amostras para controle preditivo. Apesar de trivial, ele é eficaz na prática e se mostra competitivo em relação a métodos mais complexos, principalmente pelo fato de encontrar soluções razoáveis muito rapidamente. Um pseudocódigo para o método está descrito no Algoritmo 1.

O algoritmo consiste em perturbar uma trajetória nominal aleatoriamente em diferentes amostras, seguindo uma distribuição gaussiana. Em seguida, simular todas as amostras, e selecionar a trajetória que minimiza a função objetivo como a nova trajetória nominal. Esse processo é repetido para cada novo horizonte, até que a simulação termine.

2.4 Considerações Finais

Ao longo deste capítulo, foram explorados os fundamentos teóricos que embasam o desenvolvimento de um controlador para veículos de duas rodas utilizando o método MPC. Iniciou-se pela análise da dinâmica desses veículos, ressaltando a complexidade envolvida em manter a estabilidade e o controle durante a condução. Em seguida, foi discutida a importância dos motores de física na simulação precisa desses sistemas, com destaque para o MuJoCo, que se mostrou uma ferramenta eficaz devido à sua capacidade de lidar com simulações complexas em tempo real. Além disso, abordou-se os princípios da teoria de controle, enfatizando a relevância da otimização de trajetória e do MPC na busca por um controle eficiente e robusto. Esses conceitos fornecem a base necessária para a implementação de sistemas que não apenas estabilizam o veículo, mas também garantem sua orientação precisa em diferentes cenários de simulação.

3 Revisão bibliográfica

Desde a introdução da bicicleta no século XIX, a estabilidade e o controle desses veículos se tornaram objetos de estudo de vários trabalhos, principalmente pela natureza complexa do problema, que apresenta vários desafios na área da teoria de controle. A bicicleta moderna é fruto de melhorias feitas ao longo de inúmeras iterações ao longo das décadas, muitas delas realizadas de forma empírica. As primeiras “bicicletas” eram difíceis de controlar, instáveis, pesadas e perigosas. Foi por meio das observações daqueles que pilotavam e dos testes de novos modelos que a bicicleta adquiriu as características que conhecemos hoje, como rodas pneumáticas, transmissão por corrente, suspensões e a geometria do quadro (HERLIHY, 2004). Neste capítulo, são apresentados os principais trabalhos relacionados na Seção 3.1, seguido de uma análise comparativa na Seção 3.2 e considerações finais na Seção 3.3.

3.1 Trabalhos Relacionados

O desenvolvimento da bicicleta ter ocorrido dessa forma permitiu o surgimento de vários trabalhos interessados em entender e analisar o seu funcionamento, já que este não é imediatamente intuitivo, e os resultados poderiam auxiliar nas melhorias à bicicleta. Whipple (1899) e Carvallo (1901) são reconhecidos como os primeiros a encontrarem as equações que descrevem o movimento e estabilidade das bicicletas. O modelo Whipple-Carvallo considera a bicicleta como sendo formada por quatro corpos rígidos: as rodas traseira e dianteira, o conjunto do guidão e o quadro traseiro, sendo o piloto parte do quadro traseiro e estático, como mostra a Figura 3.1. Esse modelo serviu de base para dezenas de outros trabalhos ao longo do século seguinte, conforme revisado por Schwab e Meijaard (2013). Apesar de ter permitido a Whipple encontrar os intervalos de velocidade em que a bicicleta é auto estável, seu modelo considerava as rodas como sendo completamente rígidas, com bordas afiadas e sem nenhum deslizamento. Naturalmente, esse não é o caso para bicicletas reais e, portanto, esse resultado é pouco relevante para aplicações práticas.

A maior parte dos trabalhos subsequentes focaram em estender o modelo para incluir fatores não considerados anteriormente, como rodas com formato toroidal com deformação e atrito (KLEIN; SOMMERFELD, 1897) (DOEHRING, 1953). Porém, todos esses trabalhos tiveram como foco a análise qualitativa da estabilidade desses veículos, e não o controle da trajetória.

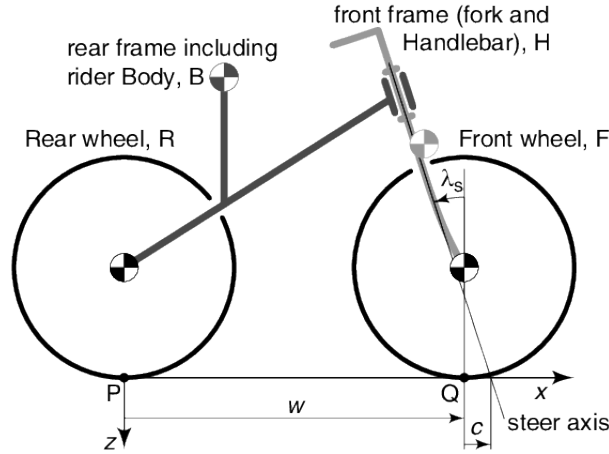


Figura 3.1: Modelo de bicicleta Whipple-Carvalho: quatro corpos rígidos (roda traseira, quadro traseiro, conjunto do guidão, roda dianteira) conectados por três juntas rotativas (cubo traseiro, eixo de direção, cubo dianteiro). Fonte: (SCHWAB; MEIJAARD, 2013).

3.1.1 Dinâmica Inversa

Enquanto os trabalhos anteriores focaram em responder questões relacionadas à auto estabilidade e equilíbrio, Getz e Marsden (1995) foram os primeiros a desenvolver um controlador capaz de não só manter o veículo equilibrado, mas seguir uma trajetória parametrizada. Esse trabalho pode ser considerado o primeiro a conseguir controlar a locomoção da bicicleta, o que além do equilíbrio lateral requer um posicionamento correto do veículo no espaço.

Esse trabalho utilizou a técnica de dinâmica inversa, que é amplamente utilizada em robótica e sistemas de controle, consistindo em calcular as forças e torques necessários para que um sistema siga uma trajetória desejada, a partir de seus estados atuais e de seu modelo dinâmico. Basicamente, o controlador utiliza equações da dinâmica inversa para determinar os comandos de controle que devem ser aplicados para atingir um estado específico. No caso da bicicleta autônoma, isso significa calcular a força no guidão e os ajustes de velocidade e inclinação necessários para manter a estabilidade e seguir

a trajetória definida. Os testes realizados demonstraram que o controlador baseado em dinâmica inversa foi eficaz em manter a bicicleta estável ao seguir uma trajetória predefinida. Isso incluiu manobras como curvas acentuadas e mudanças de direção rápidas. A precisão do controle foi avaliada com base na capacidade do sistema de seguir a trajetória desejada sem perder a estabilidade, mesmo em situações complexas.

Além dos pontos positivos, o trabalho apresenta algumas limitações importantes. O modelo da bicicleta utilizado para o controle era bastante simplificado, não levando em consideração fatores como o formato e o atrito dos pneus, que poderiam influenciar significativamente a dinâmica do sistema em situações reais. Outro ponto negativo é que os testes realizados foram bastante básicos, focando apenas em trajetórias em um plano simples e sem considerar terrenos irregulares, obstáculos ou cenários mais desafiadores, que são comuns em aplicações práticas. Além disso, o modelo proposto não incluía a presença de um piloto, o que limita ainda mais a representatividade dos resultados, já que o comportamento da bicicleta autônoma poderia ser afetado pela interação com um condutor humano em condições reais de uso.

3.1.2 Aprendizado Por Reforço

Randløv e Alstrøm (1998) foram os primeiros a utilizar aprendizado por reforço para tentar resolver esse problema. Para isso, o problema foi dividido em duas partes: aprender a manter o equilíbrio da bicicleta e depois conduzi-la até um objetivo predefinido. O método utilizado foi o algoritmo de aprendizado por reforço Sarsa, que foi modificado para reduzir o tempo de cálculo. A técnica de “*shaping*”, inspirada pela psicologia comportamental, foi usada para simplificar o processo de aprendizado, dividindo o problema em subtarefas mais fáceis, permitindo que o agente resolvesse essas subtarefas gradualmente até atingir o objetivo principal.

O sistema de controle foi projetado com base no estado do sistema em cada instante, onde o agente recebe informações sobre o ângulo e a velocidade angular do guidão, bem como o ângulo e a aceleração angular da inclinação da bicicleta em relação à vertical. Com essas informações, o agente escolhe as ações de torque a serem aplicadas no guidão e o deslocamento do centro de massa da bicicleta. Inicialmente, o agente aprendeu

a equilibrar a bicicleta em um ambiente simulado, e depois passou a dirigir a bicicleta em direção a um ponto alvo. O processo de aprendizado mostrou que, após cerca de 1700 tentativas, o agente era capaz de manter o equilíbrio por longos períodos, e após cerca de 4200 tentativas, conseguia atingir o objetivo de direção.

O método de aprendizado por reforço se provou eficaz, com o uso do *shaping* sendo fundamental para acelerar o processo de aprendizado. Por outro lado, o modelo da bicicleta utilizado foi uma simplificação do problema real. Não foram incluídas interações mais complexas no modelo, como o efeito do atrito dos pneus ou o comportamento dinâmico completo da bicicleta. Isso significa que o modelo não é completamente representativo do cenário real de uma bicicleta, o que poderia ser um ponto negativo em termos de generalização para ambientes mais complexos ou em um sistema físico real. Além disso, o agente precisou de milhares de tentativas para aprender a tarefa, o que demonstra que a técnica ainda requer uma grande quantidade de iterações para encontrar uma boa solução.

Em Tan et al. (2014), o autor apresenta uma abordagem para a simulação e controle de um personagem humano montado em uma bicicleta, utilizando técnicas de aprendizado por reforço. O objetivo do estudo foi permitir que o personagem aprenda a realizar manobras complexas e acrobacias com diferentes tipos de bicicletas, como BMX e unicycle. O sistema desenvolvido consiste em dois componentes principais: aprendizado offline e simulação online. No aprendizado offline, os controladores para as acrobacias são otimizados através de busca de políticas, com a ajuda do algoritmo NEAT (Neuroevolution of Augmenting Topologies), que otimiza tanto a parametrização quanto os parâmetros dos controladores. O sistema utiliza redes neurais para o controle de *feedback*, e *splines* para o controle *feed-forward*, permitindo que o personagem reaja em tempo real aos comandos do usuário e mantenha o equilíbrio durante as manobras. Os resultados mostraram que os controladores aprendidos foram robustos o suficiente para resistir a grandes perturbações e permitiram a realização de uma ampla variedade de acrobacias. O estudo também avaliou o impacto da otimização da parametrização nas políticas de controle, mostrando que uma abordagem combinada, envolvendo aprendizado de redes neurais e a evolução de topologias, pode produzir controladores altamente eficazes para tarefas complexas de controle de veículos de duas rodas.

Apesar de ter conseguido desenvolver com êxito um controlador robusto, resistente a perturbações e capaz de completar as manobras, algumas das manobras simuladas no trabalho de Tan et al. (2014) não pareciam tão naturais quanto as realizadas por ciclistas humanos, devido a decisões de simplificar o modelo e forçar certos comportamentos para facilitar o problema. Além disso, o estudo não minimizou o esforço do ciclista durante o aprendizado offline, o que poderia ter produzido movimentos mais suaves e realistas.

Já em Baltes, Christmann e Saeedvand (2023), os autores propõem e avaliam um controlador de aprendizado por reforço profundo (DRL) para estabilizar e controlar um robô humanoide montado em um scooter de duas rodas. O estudo compara o desempenho desse controlador com um controlador clássico *Proportional-Integral-Derivative* (PID). O objetivo é permitir que o robô controle o scooter em cenários desafiadores, como diferentes velocidades e rápidas mudanças de direção, utilizando o algoritmo de *Proximal Policy Optimization* (PPO) para o aprendizado do DRL. O projeto utiliza um modelo 3D do sistema robô-scooter, desenvolvido e simulado no ambiente NVIDIA Isaac Gym, e testa a eficácia dos controladores em condições como manter o equilíbrio sem perturbações, equilibrar-se com distúrbios externos e seguir trajetórias em linha reta e senoidais. Os resultados indicaram que o controlador DRL obteve um desempenho superior ao PID, com uma melhoria média de 52% na precisão do controle de trajetória em diferentes velocidades. Embora o PID tenha conseguido equilibrar o scooter, ele foi mais sensível a oscilações e erros de rastreamento de trajetória, especialmente em velocidades mais altas. O controlador DRL, por outro lado, mostrou maior estabilidade e robustez, adaptando-se melhor a variações de velocidade e comandos de direção, além de exigir menos ajustes de parâmetros em comparação com o PID. Apesar disso, o controlador DRL exige um treinamento intensivo em simulação, o que é um processo demorado e custoso.

Por fim, em Zhu et al. (2023) é proposto um sistema de controle baseado em aprendizado por reforço para um robô bicicleta com roda de reação (RWBR), focando na estabilidade em pavimentos curvos. O estudo aborda os desafios da instabilidade lateral e características subatuadas desse tipo de robô em terrenos irregulares. A estratégia proposta combina o controle convencional com aprendizado por reforço (RL). O sistema foi testado em um ambiente simulado no Matlab Simscape, comparando o desempenho do

controlador proposto com controladores tradicionais, como o PID e o MPC. Os resultados demonstraram que o controlador desenvolvido obteve uma robustez superior e melhor desempenho, especialmente em cenários de pavimentos curvos e com incertezas no modelo.

O sistema de Zhu et al. (2023) mostrou-se capaz de lidar com distúrbios externos e incertezas no modelo de maneira eficiente, proporcionando maior estabilidade ao robô bicicleta em terrenos curvos do que as outras técnicas avaliadas. No entanto, o uso de uma roda de reação, capaz de transferir velocidade angular para o quadro do robô, altera profundamente as dinâmicas de controle e permite a resolução separada dos problemas de equilíbrio e orientação sem precisar lidar com os fenômenos observados em uma bicicleta real.

3.1.3 Redes Neurais

Uma solução simples e eficaz para o controle de uma bicicleta utilizando uma rede neural mínima de apenas dois neurônios é apresentada em Cook (2004). A principal motivação do trabalho é entender como humanos conseguem controlar bicicletas com relativa facilidade, sem a necessidade de realizar complexas análises matemáticas ou entender completamente as dinâmicas do veículo. Trabalhos anteriores que usaram técnicas como aprendizado por reforço exigiram milhares de tentativas e, apesar de apresentar certo grau de efetividade, não garantiram um controle suave e preciso, apresentando dificuldades em manter trajetórias em linha reta, por exemplo.

No trabalho citado, foi desenvolvida uma simulação de bicicleta virtual, e uma rede de dois neurônios foi desenhada com base em observações de como humanos controlam o ângulo de inclinação da bicicleta durante a condução. Um neurônio controla o ângulo desejado de inclinação, enquanto o segundo neurônio calcula o torque necessário a ser aplicado no guidão para manter a estabilidade ou direcionar a bicicleta.

O controlador, apesar de simples, foi capaz de conduzir a bicicleta na direção desejada, ajustando continuamente a inclinação da bicicleta para garantir a estabilidade. As entradas para a rede incluem o ângulo atual de inclinação, a orientação desejada, e suas respectivas taxas de variação. Surpreendentemente, a simplicidade do modelo provou ser suficiente para controlar a bicicleta de maneira eficaz em uma ampla gama

de velocidades. No entanto, o método não foi capaz de lidar com todas as situações de instabilidade, especialmente em velocidades muito baixas ou em curvas acentuadas, onde a bicicleta tendia a perder o equilíbrio.

3.1.4 Controle Ótimo

Mikkola (2024) abordou o desenvolvimento de controlador baseado em planejamento online para animação física de skateboarding. Apesar de não se tratar de bicicletas, o problema compartilha similaridades com o tratado nesse trabalho. O objetivo foi replicar movimentos realistas de skate em um ambiente simulado, utilizando uma aproximação de um skate real e de um personagem humanoide. Baseado no framework MuJoCo Model Predictive Control (MJPC), o problema é formulado como uma série de subtarefas como *pushing* (impulsionar o skate) e *steering* (virar o skate), que usam poses heurísticas e funções de recompensa, sem a necessidade de dados de captura de movimento e utilizando os controles para realizar manobras de skate em terreno plano.

Os resultados indicaram que o sistema atingiu certo sucesso em replicar os movimentos básicos de empurrar e direcionar o skate. No entanto, manobras mais complexas, como truques, não foram possíveis de serem executadas. O estudo também comparou o desempenho de diferentes algoritmos de planejamento, destacando as diferenças entre métodos baseados em amostras e em gradiente. O uso do MPC permitiu ao sistema se adaptar a variações no ambiente e nas metas de movimento, sem a necessidade de ajustes manuais de parâmetros.

Picotti, Bianchin e Bruschetta (2024) propõem o uso de um controlador de MPC em tempo real auxiliado por aprendizado baseado em processos gaussianos para controlar uma motocicleta virtual. O objetivo do trabalho é melhorar a precisão do controle de trajetórias ao incorporar um modelo de previsão que combina uma descrição física simplificada da motocicleta com um componente de aprendizado para compensar erros. O modelo desenvolvido utiliza dados adquiridos durante execuções de treinamento para ajustar a previsão de acelerações e aplicar correções durante a execução em tempo real.

De acordo com o autor, o controle com base no aprendizado superou modelos puramente físicos, apresentando melhor desempenho de rastreamento de trajetória e robustez

em diferentes condições. A aplicação do controle ótimo, ajustado com processos gaussianos, conseguiu corrigir erros que não haviam sido previstos pelo modelo físico inicial, o que se mostrou especialmente útil em cenários com mudanças abruptas de aceleração e curvas acentuadas. O controlador foi validado em simulações de alta fidelidade, que emularam o comportamento da motocicleta em várias condições de pista e velocidades.

3.2 Análise Comparativa

Na Tabela 3.1 temos a listagem dos trabalhos revisados e de suas respectivas estratégias de controle aplicadas. Na Tabela 3.2 estão apresentadas a modelagem do problema utilizada em cada trabalho, a principal métrica de avaliação e se a modelagem inclui o piloto.

	Trabalho	Estratégia de controle
1	Getz e Marsden (1995)	Dinâmica Inversa
2	Randløv e Alstrøm (1998)	Aprendizado por Reforço
3	Tan et al. (2014)	
4	Baltes, Christmann e Saeedvand (2023)	
5	Zhu et al. (2023)	
6	Cook (2004)	Redes Neurais
7	Mikkola (2024)	Controle Ótimo (MPC)
8	Picotti, Bianchin e Bruschetta (2024)	

Tabela 3.1: Listagem dos trabalhos relacionados.

	Modelagem do problema	Métrica de avaliação	Com piloto
1	Trajetória sobre plano	Precisão da trajetória	Não
2			
3	Direção sobre terrenos irregulares	Precisão da direção	Sim
4	Trajetória sobre plano	Velocidade máxima	
5	Trajetória sobre plano curvado	Precisão da trajetória	Não
6	Trajetória e direção sobre plano		
7	Trajetória sobre terrenos irregulares		Sim
8	Trajetória sobre plano		

Tabela 3.2: Tabela comparativa entre os trabalhos relacionados.

Os primeiros estudos, como os de Getz e Marsden (1995), focaram em métodos clássicos de dinâmica inversa. Embora esses métodos tenham alcançado resultados interessantes para controle de trajetória, eles não são robustos para situações com variabilidade de terreno ou interações complexas entre o veículo e o piloto.

Trabalhos mais recentes, como os de Picotti, Bianchin e Bruschetta (2024), que utilizam uma combinação de MPC com aprendizado, demonstraram resultados superiores em cenários com mudanças abruptas de condições e maior robustez ao lidar com trajetórias complexas. Isso se deve à capacidade de prever e ajustar as variáveis de controle em tempo real, uma característica intrínseca ao MPC.

Adicionalmente, o trabalho de Randløv e Alstrøm (1998), que explorou o aprendizado por reforço, foi pioneiro ao demonstrar a viabilidade de agentes aprenderem a controlar uma bicicleta em ambiente simulado. No entanto, a alta quantidade de interações necessárias para o aprendizado torna esse método custoso e eficaz somente em alguns casos específicos incluídos no treinamento.

Dessa forma, observamos que o MPC se apresenta como uma solução promissora para controle de bicicletas, e apesar dos recentes avanços em pesquisa do método, não foi encontrado nenhum trabalho que tenha tratado especificamente desses veículos. Espera-se, com esse trabalho, fornecer um método de controle robusto e eficiente, em tempo real, em cenários dinâmicos e sem processo de treinamento.

3.3 Considerações Finais

Embora existam diversas abordagens para o controle de veículos de duas rodas, o tema ainda é relativamente pouco explorado. Diferentes métodos, como controle baseado em dinâmica inversa, aprendizado por reforço, controle ótimo e redes neurais, foram aplicados com graus variados de sucesso, mas nenhum deles emergiu como uma solução definitiva para o problema.

Ademais, não há consenso sobre qual método é o mais eficaz para o controle de veículos de duas rodas, com diferentes abordagens se mostrando eficazes em cenários específicos. Por exemplo, o controle preditivo baseado em modelo (MPC) tem mostrado bons resultados em tempo real, mas depende fortemente da precisão do modelo utilizado, enquanto o aprendizado por reforço pode oferecer uma solução mais flexível, embora o processo de treinamento seja intensivo e demorado.

Em síntese, o campo apresenta um grande potencial para avanços, especialmente com a integração de novas tecnologias e métodos híbridos que combinem diferentes

técnicas de controle. Este trabalho, ao explorar o uso de MPC em um ambiente de simulação com o motor MuJoCo, visa contribuir para esse campo ainda em crescimento, oferecendo uma abordagem que busca o equilíbrio entre robustez, eficiência e viabilidade prática.

4 Implementação

Neste capítulo, são detalhados os procedimentos e as escolhas técnicas que nortearam a implementação do modelo de controle baseado em MPC aplicado ao sistema composto pelo veículo de duas rodas e seu piloto humanoide. A descrição aborda desde a modelagem matemática do sistema dinâmico até a integração dos componentes no ambiente de simulação do MuJoCo, destacando os aspectos essenciais para garantir a precisão e a robustez do controlador. Além disso, são apresentados os ajustes realizados para compatibilizar as características do controlador às exigências do sistema simulado, bem como as estratégias adotadas para lidar com os desafios computacionais inerentes à aplicação do MPC em cenários de alta complexidade.

4.1 Modelo do humanóide

O modelo humanóide utilizado é o mesmo apresentado em Tassa, Erez e Todorov (2012), que é distribuído junto com a biblioteca MuJoCo. É também o mesmo modelo utilizado em Mikkola (2024). A Figura 4.1 mostra o modelo em sua posição inicial. O humanoide possui 1,60m de altura e 55kg de massa total. No total, são 22 graus de liberdade divididos entre quadril, ombro, abdômen (2 graus de liberdade cada) e joelho, cotovelos e calcanhar (1 grau de liberdade). A Tabela 4.1 mostra os graus de liberdade de cada articulação, que compõe o vetor de controle, e intervalos de valores de cada um.

Os modelos na MuJoCo são especificados por meio de arquivos XML, que contêm definições físicas do modelo, configurações visuais (iluminação, câmera, etc.) e configurações da simulação. Por meio desse mecanismo, foi possível importar o modelo no humanóide em um arquivo que unifica e inclui definições para o modelo final. A única alteração feita no modelo distribuído junto com a biblioteca foi reduzir a amplitude de movimento do joelho em 20° , para evitar que as pernas se estiquem demais e acabem travando o movimento de pedalar.

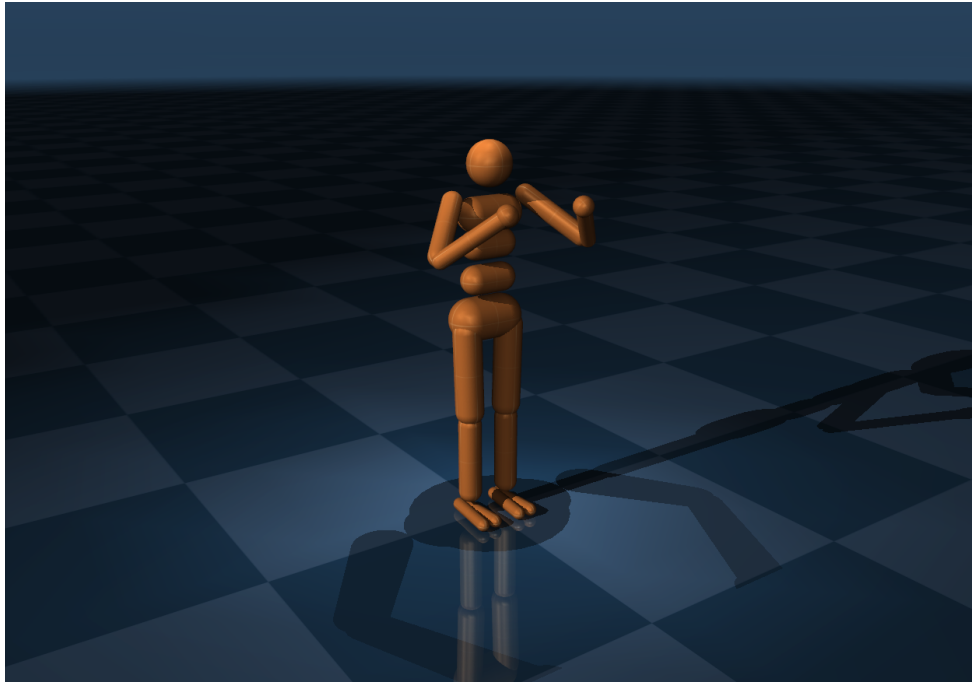


Figura 4.1: Modelo do humanóide incluído na biblioteca MuJoCo. Fonte: do autor.

Articulação	Graus de Liberdade	Intervalo de Valores
Abdômen	x, y, z	$[-1, -1]$
Quadril Direito	x, y, z	$[-1, -1]$
Joelho Direito	x	$[-1, -1]$
Tornozelo Direito	x, y	$[-1, -1]$
Quadril Esquerdo	x, y, z	$[-1, -1]$
Joelho Esquerdo	x	$[-1, -1]$
Tornozelo Esquerdo	x, y	$[-1, -1]$
Ombro Direito	x, y	$[-1, -1]$
Cotovelo Direito	x	$[-1, -1]$
Ombro Esquerdo	x, y	$[-1, -1]$
Cotovelo Esquerdo	x	$[-1, -1]$

Tabela 4.1: Tabela de Controle das Articulações

4.2 Modelo da bicicleta

O modelo da bicicleta foi desenvolvido para este trabalho. Ele foi baseado em um modelo de bicicleta muito comum nos Países Baixos, onde a bicicleta é muito utilizada para locomoção nos centros urbanos (TAI; VERAART, 2019). Esse tipo de bicicleta foi escolhido por ser muito comum e também por sua simplicidade, já que não apresenta suspensões traseira nem dianteira, o que simplifica a modelagem. Além disso, por ser usada principalmente em relevos planos, seu sistema de transmissão também é simplificado. A Figura 4.2 mostra o modelo desenvolvido. A bicicleta possui uma massa total de 19,3kg, com

aproximadamente 1,83m de comprimento e 1,14m de altura. O modelo possui 12 graus de liberdade, sendo 2 nos eixos de cada roda, 1 no eixo de direção do guidão, 1 para o pedivela, 2 para os pedais e 6 para o corpo da bicicleta.

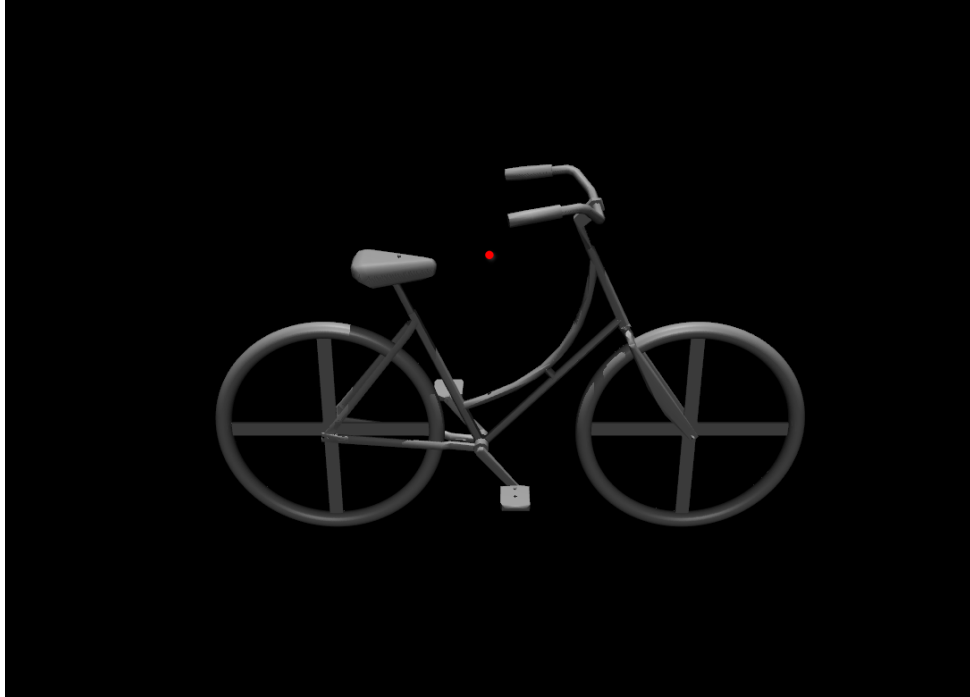


Figura 4.2: Modelo da bicicleta desenvolvido. O ponto em vermelho é a localização arbitrária na bicicleta escolhida para computar a distância da rota. Fonte: do autor.

A malha 3D da bicicleta foi desenvolvida no software Blender e cada parte exportada separadamente para o formato STL. Esses arquivos são referenciados no arquivo XML que define o modelo da bicicleta, que também contém as definições físicas e visuais do modelo. A MuJoCo gera automaticamente a malha de colisão convexa a partir da malha visual, portanto as partes devem ser separadas levando em conta o resultado desse processo. Nas rodas, foram usadas geometrias toroidais para representar os pneus, com uma densidade de vértices relativamente alta, já que é esperado que a interação com o terreno seja um dos pontos críticos do controle. Cada roda é composta de 8192 vértices, sendo 16384 para o par. O restante do modelo é composto por somente 832 vértices, totalizando 17216 vértices no modelo completo. As malhas foram simplificadas ao máximo, já que sua função principal é puramente visual.

O modelo usa uma restrição de igualdade para simular o mecanismo de transmissão da bicicleta. Restrições de igualdade para articulações na MuJoCo são formuladas como uma equação polinomial de grau 4, que é resolvida numericamente a cada passo de

simulação. A restrição de igualdade para a roda traseira é dada por:

$$y - y_0 = a_0 + a_1(x - x_0) + a_2(x - x_0)^2 + a_3(x - x_0)^3 + a_4(x - x_0)^4 \quad (4.1)$$

onde y_0 e x_0 são os ângulos iniciais da roda traseira e do pedivela, respectivamente, y e x são os ângulos atuais, e a_0 , a_1 , a_2 , a_3 e a_4 são coeficientes parametrizáveis. Nesse caso, todos os coeficientes foram fixados em 0, exceto a_1 , que foi fixado em 2,75. Além disso, os ângulos iniciais das articulações é 0, portanto o resultado é

$$y = 2.75x \quad (4.2)$$

o que significa simplesmente que a roda traseira deve girar 2,75 voltas para cada volta do pedivela. Portanto, estamos aproximando o comportamento de uma bicicleta de marcha única, com razão de transmissão de 1:2,75, que é um valor comum em bicicletas urbanas e de passeio.

4.3 Modelo final

Os modelos do humanoide e da bicicleta foram importados em um terceiro arquivo, onde as demais configurações são realizadas. Primeiro, o humanoide foi posicionado na bicicleta, de forma empírica, por meio de alterações manuais nos valores das articulações que posteriormente foram salvas em um *keyframe*. O mecanismo de *keyframe* no MuJoCo permite definir um vetor de números reais que pode ser carregado no início da simulação, sobrescrevendo as posições iniciais das articulações. O resultado desse posicionamento pode ser visto na Figura 4.3.

Foram desabilitadas as colisões entre certos corpos do modelo, por motivos variados. Primeiro, foram usadas restrições de igualdade para manter as partes do corpo posicionadas, que serão detalhadas mais a seguir. Para garantir a estabilidade da simulação e garantir o posicionamento correto, as colisões entre os corpos envolvidos nessas restrições foram desativadas. Além disso, foram observadas algumas colisões problemáticas entre os pés e a bicicleta que estavam atrapalhando a movimentação das pernas e que também fo-



Figura 4.3: Modelo final da bicicleta e humanoide. Fonte: do autor.

ram desativadas. Por fim, foram desativadas as colisões entre corpos que nunca deveriam colidir ao longo da simulação, já que isso contribui para um melhor desempenho.

Os dois modelos são conectados por meio de cinco restrições de igualidade entre pontos arbitrários: dois pontos nas mãos conectados ao guidão (rotação permitida), um ponto no quadril conectado ao assento (rotação permitida) e dois pontos conectando os pés aos pedais (rotação não permitida). Portanto, as mãos e o quadril permanecem nos locais corretos ao longo da simulação, mas com liberdade de rotação para permitir a movimentação necessária dos membros. Os pés permanecem estacionários em relação aos pedais tanto em posição quanto em rotação, dessa forma simulando um mecanismo de trava que existe em bicicletas reais. A Figura 4.4 mostra a posição de cada ponto no modelo final.

Essa configuração é mais flexível do que a encontrada nos trabalhos da literatura, que restringem a movimentação do humanoide mais rigorosamente. Essa abordagem apresenta a vantagem de simplificar o modelo e reduzir as possibilidades de controle, e consequentemente, o espaço de busca. Porém, esse trabalho optou por uma abordagem mais flexível, que permite analisar a capacidade do controlador de lidar com um modelo mais complexo e realista.

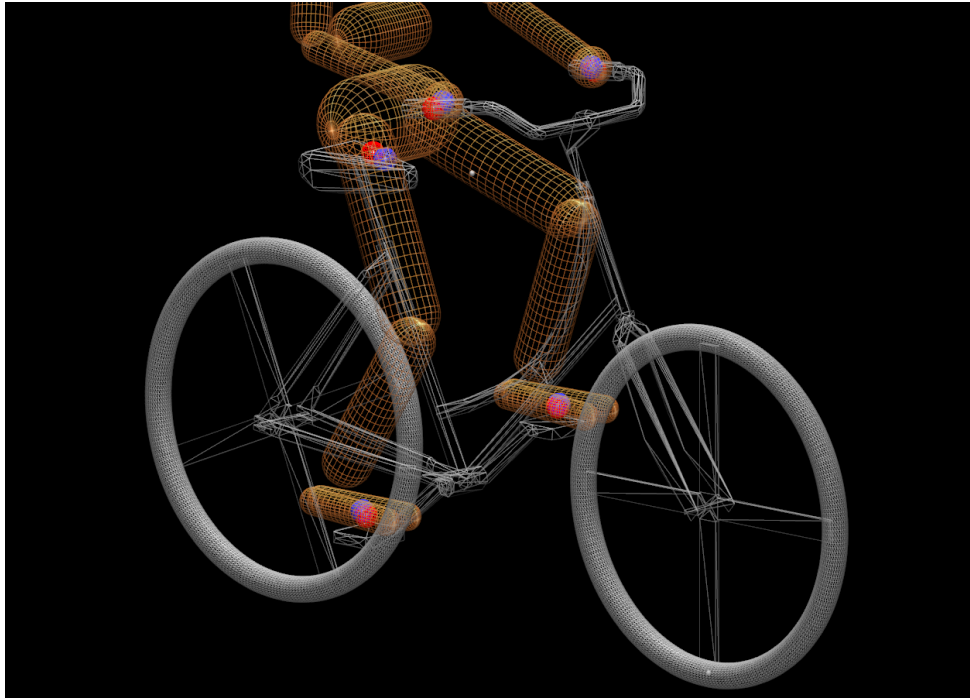


Figura 4.4: Posições das restrições de igualdade do modelo. Em vermelho, os pontos atribuídos à bicicleta, e em azul os pontos atribuídos ao humanoide. Fonte: do autor.

4.4 Controlador

O problema foi definido e modelado como um problema de rastreamento de trajetória, onde o objetivo é fazer com que o humanoide conduza a bicicleta por uma trajetória pré-definida. Essa trajetória é definida por uma sequência de pontos no espaço, que são interpolados por um *spline* cúbico. O controlador é responsável por gerar as ações necessárias para que o humanoide siga essa trajetória, mantendo a bicicleta equilibrada e estável.

4.4.1 Rota

As rotas são previamente definidas manualmente, geradas no software Blender para cada um dos cenários de teste e exportadas para um arquivo CSV. Cada rota é composta por uma sequência de pontos no espaço, que são interpolados por um *spline* cúbico. A Figura 4.5 mostra um exemplo de rota gerada para um dos cenários de teste.

No início da simulação, as rotas são carregadas e a sequência final de pontos da rota é gerada, inserindo 20 pontos intermediários entre cada par de âncoras do *spline*. Isso significa que a densidade de pontos ao longo da rota não é constante, mas sim proporcional

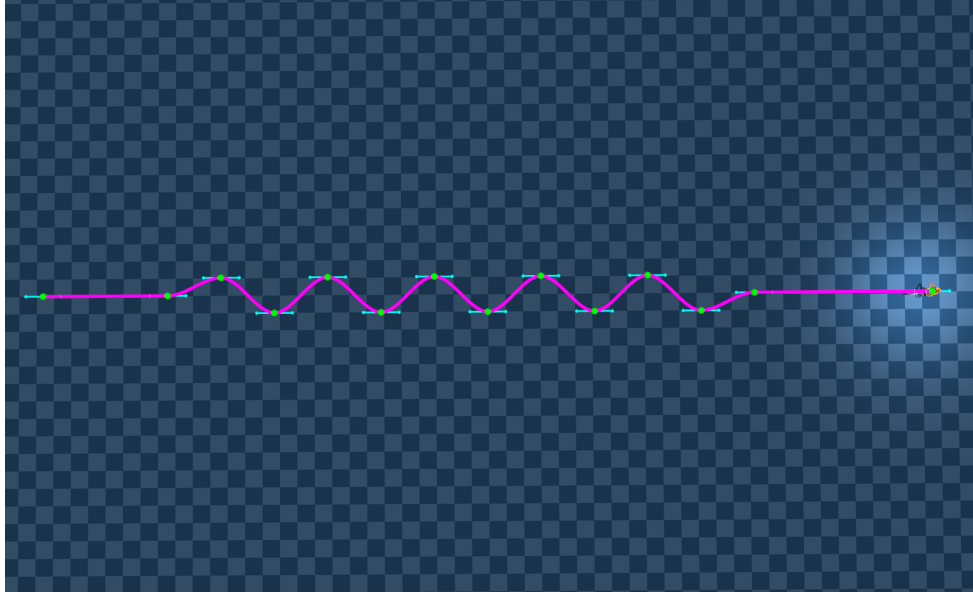


Figura 4.5: Exemplo de rota gerada para um dos cenários de teste. Os pontos em verde são os pontos de ancoragem da rota, os pontos em azul são os pontos de controle, e em roxo a rota final gerada pelo *spline* cúbico. Fonte: do autor.

à distância entre os pontos de ancoragem. Essa característica permite maior flexibilidade ao definir a rota, já que é possível ter mais pontos em trechos mais complexos e menos pontos em trechos mais simples. Além disso, 20 pontos intermediários se mostraram suficientes para garantir suavidade às rotas definidas e ainda manter boa performance.

4.4.2 Função de custo

A função de custo é um somatório de termos independentes que apresenta o seguinte formato:

$$c(s, u) = \sum_{i=1}^N w_i \times n_i(r_i(s, u)) \quad (4.3)$$

onde o custo c de aplicar o vetor de controle u ao estado s é dado pelo somatório de N termos onde w_i é o seu peso, n_i é uma função norma positiva com mínimo em 0, e $r_i(s, a)$ é um vetor de residuais cujos elementos são minimizados quando a tarefa é resolvida. Essa função de custo é apresentada em Todorov, Erez e Tassa (2012), e é adotada no framework MJPC utilizado nesse trabalho.

Para este trabalho, foram escolhidos três termos:

- Distância entre a bicicleta e a curva
- Módulo da diferença entre velocidade linear atual e velocidade alvo

- Somatório de esforço de controle em todos os atuadores

Distância entre a bicicleta e a curva

O primeiro termo serve para manter o veículo na trajetória desejada, assim como priorizar a inclinação vertical da bicicleta. O ponto escolhido para ser usado pode ser visto no centro da Figura 4.2, que foi escolhido empiricamente. As rotas foram então definidas para que, na posição vertical e com as duas rodas em contato com o terreno, a distância entre o ponto e a rota seja mínima. O vetor residual é então dado por:

$$r_1(s, u) = \vec{P}_{\text{bicicleta}} - \vec{P}_{\text{rota}} \quad (4.4)$$

onde $\vec{P}_{\text{bicicleta}}$ é a posição global do ponto na bicicleta e $\vec{P}_{\text{rota}}$ é o ponto mais próximo da rota. Esse último é encontrado por meio de uma busca linear, já que podemos descartar os pontos já visitados e encerrar a busca quando a distância entre o ponto e a rota começa a aumentar. Por fim, como se trata de uma distância euclidiana, a função norma é simplesmente a norma euclidiana:

$$n_1(r_1) = \|r_1\|_2 \quad (4.5)$$

Módulo da diferença entre velocidade linear atual e velocidade alvo

O segundo termo controla a velocidade alvo do veículo, para que ele se mantenha percorrendo a rota. A velocidade alvo foi mantida com magnitude constante em todos os cenários de teste ao longo de toda a trajetória, sendo definida como um vetor tangente à rota em cada ponto. O vetor residual é então dado por:

$$r_2(s, u) = \vec{v}_{\text{alvo}} - \vec{v}_{\text{atual}} \quad (4.6)$$

onde $\vec{v}_{\text{alvo}}$ é a velocidade linear alvo e $\vec{v}_{\text{atual}}$ é a velocidade linear atual. A função norma é a norma euclidiana:

$$n_2(r_2) = \|r_2\|_2 \quad (4.7)$$

Somatório de esforço de controle em todas os atuadores

O terceiro termo controla o esforço de controle, para que o controlador não faça movimentos bruscos e desnecessários. O vetor de controle é composto por 21 elementos, que são as forças e torques aplicados em cada uma das articulações do humanoide. Os elementos desse vetor estão apresentados em ordem na tabela 4.1. A norma nesse caso é o cosseno hiperbólico:

$$r_3(s, u) = u \quad (4.8)$$

$$n_3(r_3) = \sum_{i=1}^{21} \cosh(r_{3i}) \quad (4.9)$$

A função de custo final é então dada por:

$$c(s, u) = w_1 \times \|r_1\|_2 + w_2 \times \|r_2\|_2 + w_3 \times \sum_{i=1}^{21} \cosh(r_{3i}) \quad (4.10)$$

Neste trabalho, w_1 foi fixado em 1 e w_3 em 0,05. A ideia é que o terceiro termo seja pequeno, somente o suficiente para diferenciar soluções onde são feitos movimentos desnecessários e contribuir para um movimento mais suave e estável. Já o segundo termo foi variado para avaliar seu impacto no controle, conforme será discutido na seção de resultados. Dessa forma, podemos manter o parâmetro w_1 fixo, e avaliar a proporção entre w_1 e w_2 .

4.4.3 Algoritmo de otimização

O algoritmo de otimização escolhido para avaliar neste trabalho foi o Algoritmo 1, baseado em amostragem, pela sua simplicidade, rapidez e eficácia. O algoritmo funciona da seguinte forma: a cada passo, amostras são geradas aleatoriamente no espaço de controle. Para cada amostra, a função de custo é avaliada ao longo do horizonte escolhido e a amostra com menor custo é escolhida como a melhor solução. O processo é repetido até o fim da simulação.

O horizonte de controle foi fixado em 1,2s, determinado empiricamente. O intervalo de tempo foi fixado em 0,02s, que é o mesmo intervalo da simulação. Portanto, cada horizonte corresponde a 60 passos. O framework MJPC se encarrega de fazer a

Algoritmo 1: Amostragem preditiva

Entrada: Número de amostras N , desvio padrão σ , estado inicial x_0 e trajetória nominal $\Pi(\theta)$

Saída: Nova trajetória nominal

1 início

2 Gera $N - 1$ amostras θ_i de $\mathcal{N}(\theta, \sigma^2)$

3 Para o nominal θ e cada amostra θ_i , executa o plano a partir de x_0

4 Nova trajetória nominal é a que minimiza $J(\theta_i)$

5 fim

amostragem, distribuir as simulações em paralelo e escolher a melhor solução, avaliando um número variável de amostras até que o tempo limite seja atingido.

O horizonte de controle é um parâmetro muito importante nesse caso. Quanto mais longo, mais o controlador consegue enxergar o futuro e planejar suas ações, mas também mais custoso computacionalmente. O intervalo de tempo foi escolhido para ser o mesmo que o da simulação, para que o controlador possa agir em tempo real. O número de amostras foi escolhido para ser suficiente para garantir a convergência do algoritmo, mas não tão grande a ponto de tornar o algoritmo lento.

4.5 Cenários de teste

Para testar o controlador, foram desenvolvidos alguns cenários com diferentes tipos de terreno e obstáculos, como curvas, rampas e escadas. Os cenários foram baseados nos apresentados em (TAN et al., 2014), onde manobras complexas de bicicleta são analisadas. Dessa forma, os resultados obtidos constroem uma visão geral mais completa do controlador, e não somente de casos simples como rotas em linha reta.

4.5.1 Linha reta em terreno plano

Este é o cenário mais simples, onde o humanoide deve conduzir a bicicleta por uma rota reta em terreno plano. O objetivo é avaliar a capacidade do controlador de manter a bicicleta equilibrada e estável em condições ideais. A rota possui 50 m de comprimento e é composta por 254 pontos equidistantes, e está representada na Figura 4.6.

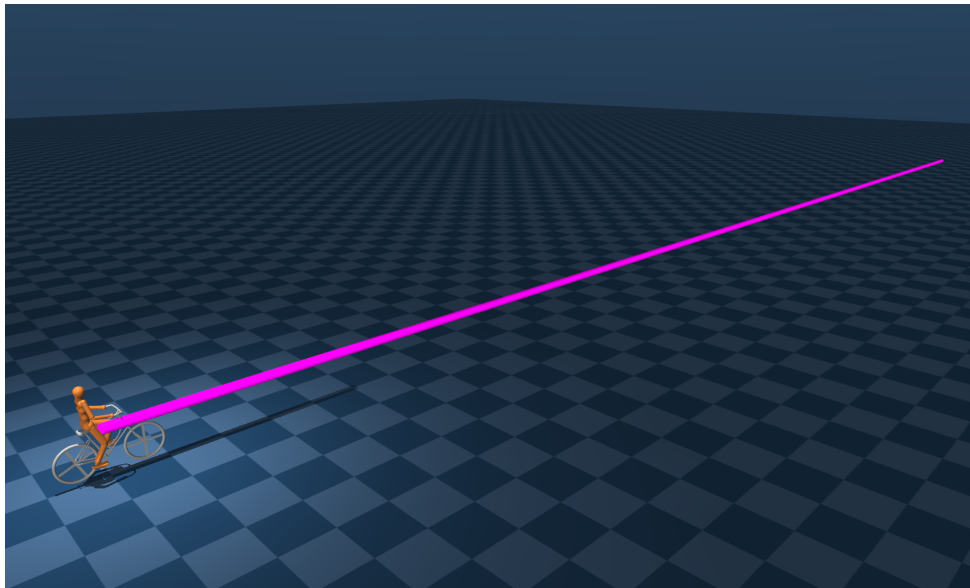


Figura 4.6: Cenário de linha reta em terreno plano. Fonte: do autor.

4.5.2 Zigue-zague em terreno plano

Neste cenário, o humanoide deve conduzir a bicicleta por uma rota em zigue-zague em terreno plano. O objetivo é avaliar a capacidade do controlador de lidar com mudanças rápidas de direção. A distância entre o ponto inicial e final é de 50 m, com 12 curvas. Esse cenário está representado na Figura 4.7.

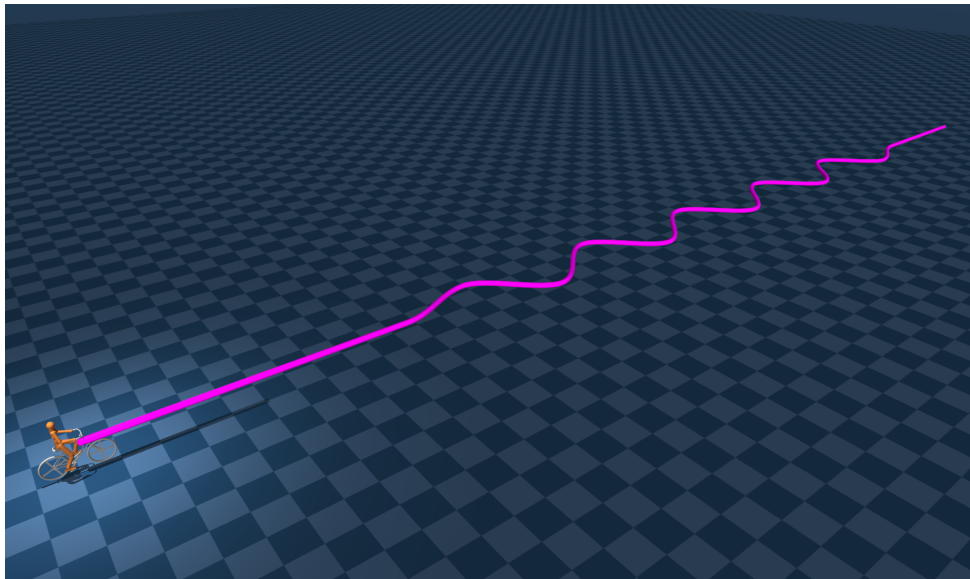


Figura 4.7: Cenário de zigue-zague em terreno plano. Fonte: do autor.

4.5.3 Escadas

Neste cenário, o humanoide deve conduzir a bicicleta por uma rota que inclui descer uma escadaria com 3 lances de escada, com 20 degraus cada. Cada degrau possui 15 cm de altura e 35 cm de largura, e a rota se estende por aproximadamente 70 metros no eixo x e 12 metros no eixo z. O objetivo é avaliar a capacidade do controlador de lidar com esse tipo de obstáculo, que é comum em ambientes urbanos. Esse cenário está representado na Figura 4.8.

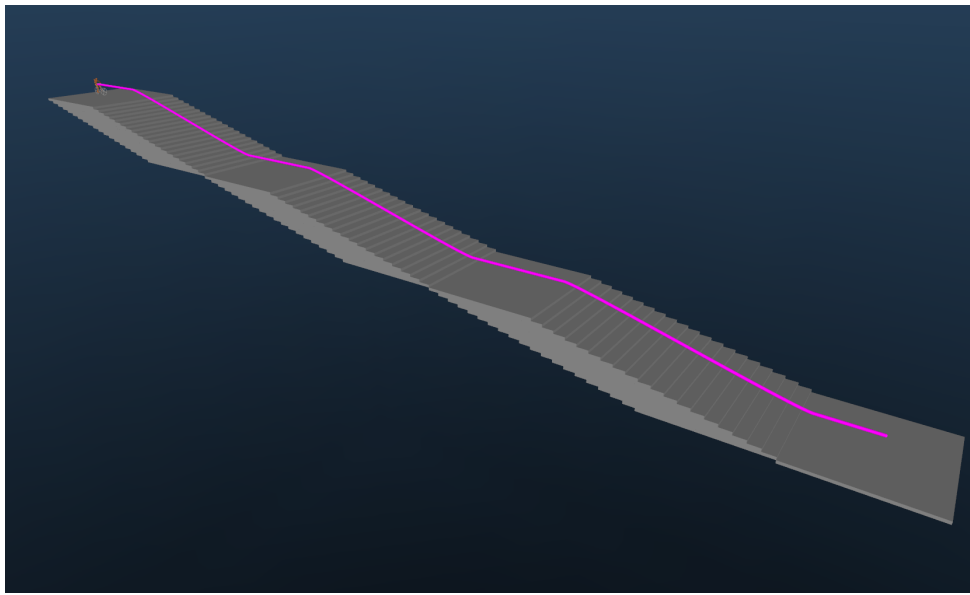


Figura 4.8: Cenário de descida de escadaria. Fonte: do autor.

4.5.4 Linha reta em terreno acidentado

O terreno acidentado ou irregular é uma maneira de testar a robustez do controlador em um cenário mais desafiador. Neste cenário, o humanoide deve conduzir a bicicleta por uma rota reta em terreno acidentado, que é a mesma do cenário de linha reta. A malha do terreno é gerada por meio de um mapa de altura, que é uma imagem em escala de cinza onde cada pixel representa a altura do terreno. A imagem foi gerada por meio da aplicação do ruído de Perlin, implementado em Python. Esse cenário é mais desafiador, pois é preciso ajustar a direção da bicicleta a todo momento para impedir a queda. A Figura 4.9 ilustra esse cenário.

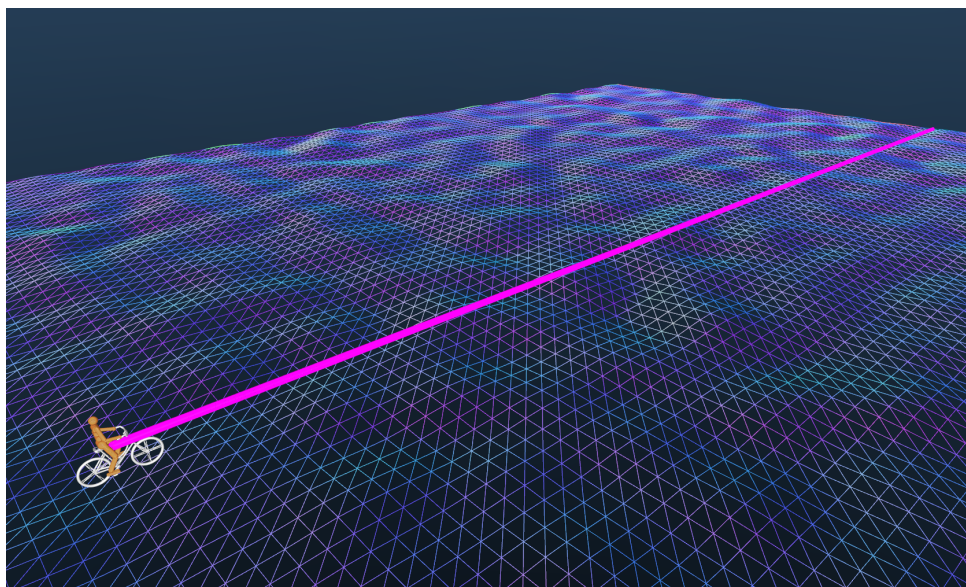


Figura 4.9: Cenário de linha reta em terreno acidentado. A diferença de altura entre o ponto mais baixo e mais alto do terreno é 40 cm. Fonte: do autor.

4.5.5 Circuito de obstáculos

Por fim, o circuito de obstáculos é um cenário que combina diversos obstáculos em um único ambiente. O humanoide deve conduzir a bicicleta por um circuito que inclui curvas, subida de rampa, descida de escada e zigue-zague, distribuídos em uma área de aproximadamente 80 por 20 metros. A Figura 4.10 mostra esse cenário.

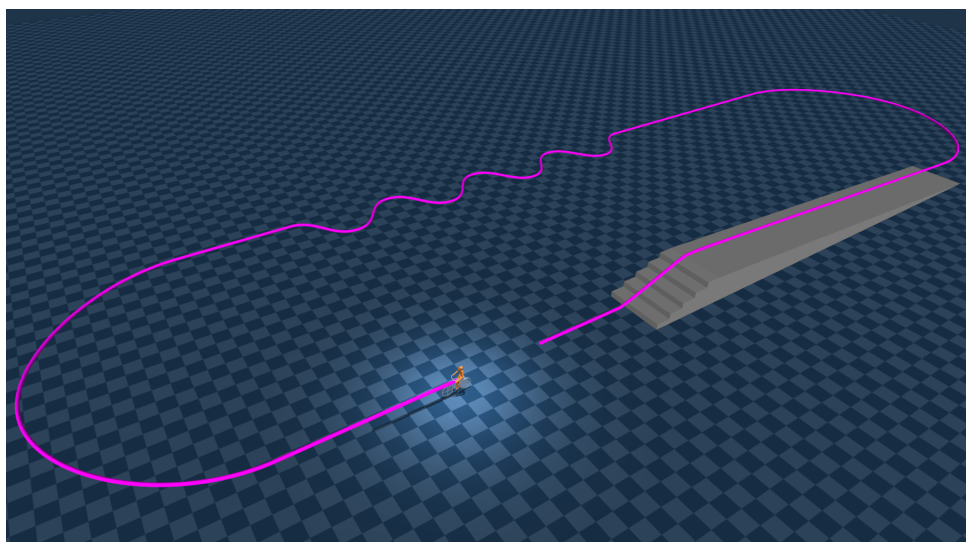


Figura 4.10: Cenário de circuito de obstáculos. Fonte: do autor.

Esse cenário tenta construir um ambiente mais dinâmico, mais próximo de um ambiente real, onde o piloto deve ser capaz de alternar entre diferentes tipos de estratégias para superar os obstáculos. Por exemplo, ao iniciar uma subida, pode ser interessante

aumentar a velocidade para ganhar impulso, enquanto ao descer uma escada, é necessário diminuir a velocidade para evitar uma queda.

A rota nesse caso é circular, iniciando e terminando no mesmo ponto. Ela inicia com uma curva aberta para a direita, com raio de 10m, seguida de uma sequência de zigue-zague (mesma configuração do cenário de zigue-zague) e outra curva aberta, dessa vez para a esquerda. Em seguida, uma subida de rampa de aproximadamente 7° de inclinação e 30m de comprimento, finalizando com um lance de escada com 5 degraus (mesmo formato do cenário de escada).

4.6 Métricas de avaliação

Para avaliar a eficácia do controlador, foram definidas algumas métricas de avaliação que permitem quantificar o desempenho do controlador em cada cenário. São elas:

- **Erro máximo da trajetória:** a distância máxima entre a rota definida e a rota percorrida pela bicicleta em qualquer passo da simulação.
- **Erro médio da trajetória:** a distância média entre a rota definida e a rota percorrida pela bicicleta ao longo de todos os passos da simulação.
- **Tempo total:** o tempo total necessário para percorrer a rota.
- **Taxa de sucesso:** corresponde à fração de pontos percorridos da rota antes de uma queda.

Essas métricas são calculadas para cada simulação e são usadas para avaliar o desempenho do controlador em diferentes cenários e condições. Elas foram escolhidas para fornecer uma visão abrangente do desempenho do controlador, levando em consideração aspectos como precisão da trajetória, eficiência do controle e robustez em diferentes situações.

5 Resultados

Foram coletados os resultados das simulações variando os parâmetros velocidade alvo, peso da velocidade e velocidade da simulação, com 10 iterações para cada combinação de parâmetros. A Tabela 5.1 mostra todos os valores testados para cada parâmetro. Portanto, foram realizadas 320 simulações para cada cenário, totalizando 1600 simulações. Os vídeos das execuções além do código fonte e dados coletados são disponibilizados³.

Parâmetro	Valores
Velocidade objetivo	1,5m/s, 2,5m/s, 3,5m/s, 4,5m/s
Peso da velocidade	0,05, 0,1, 0,15, 0,2
Velocidade da simulação	100%, 50%

Tabela 5.1: Valores testados para cada parâmetro.

5.1 Configuração da máquina

As simulações foram realizadas em uma máquina rodando Arch Linux 64 bits, com kernel 6.12.8, KDE Plasma 6 - Wayland, processador Intel Core i9-9900KF, 32 GB de memória RAM DDR4 2666 MHz e placa de vídeo NVIDIA GeForce RTX 2060. O tempo total de simulação dos 1600 experimentos foi de aproximadamente 24 horas.

Nessa máquina, os testes de performance indicam que é possível simular por volta de 137752 passos por segundo, ou seja, aproximadamente 2755 vezes a velocidade real. Considerando o horizonte de controle de 1,2s e o *time step* de 0,02s, no intervalo de tempo entre cada passo da simulação principal, é possível avaliar em média 45 trajetórias completas.

5.2 Cenário de linha reta

O cenário de linha reta é o mais simples dentre os cenários de teste desenvolvidos, e ele foi, sem dúvida, o que apresentou os melhores resultados. A Figura 5.1 mostra o resultado

³<https://drive.google.com/drive/folders/1bAJsmAtNVMZlcuePe6j1h3hyQRdly0FK?usp=sharing>

para esse cenário. A escala do eixo y foi aumentada para facilitar a visualização.

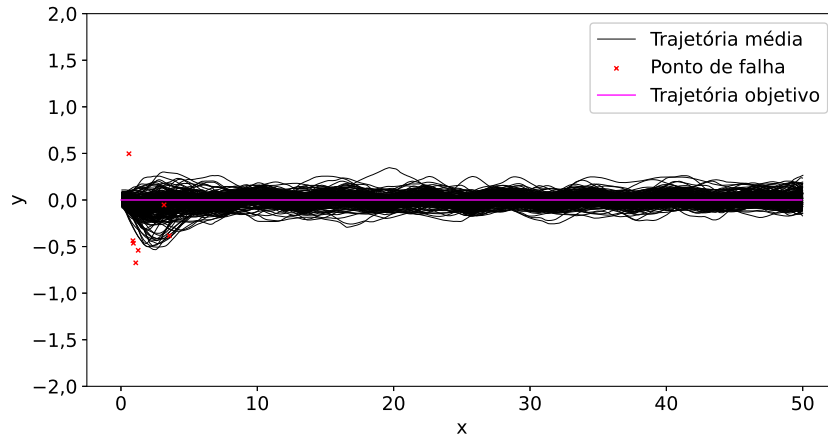


Figura 5.1: Resultados para o cenário de linha reta. Fonte: do autor.

Das 320 simulações realizadas, 6 falharam, todas logo no início da trajetória e em simulações em tempo real. Esse comportamento é esperado, já que o início do movimento é o momento mais difícil: o controlador inicia com uma trajetória de controle nula, e precisa rapidamente encontrar uma sequência de ações que permita manter a bicicleta equilibrada. Fazer isso a partir de um estado de repouso é um desafio. No entanto, após esse período inicial, o controlador rapidamente se ajusta e consegue manter a bicicleta estável e equilibrada ao longo de toda a trajetória, já que ele utiliza a solução da iteração anterior como ponto de partida.

A Tabela 5.2 mostra os resultados para todas as simulações em tempo real, agrupados por velocidade alvo e peso da velocidade. Os resultados mostram que o controlador foi capaz de alcançar uma taxa de sucesso de 100% para quase todas as combinações de parâmetros.

O erro médio de trajetória para todos os casos permaneceu relativamente pequeno, variando entre 6 cm e 10 cm, aproximadamente. Porém, pode-se observar uma tendência de erros maiores para velocidades alvo maiores e pesos da velocidade maiores. Esse comportamento pode ser interpretado como uma consequência do *trade-off* da função objetivo do controlador: ao priorizar a velocidade, o controlador pode acabar sacrificando a precisão da trajetória. No entanto, os resultados mostram que o controlador foi capaz de manter a bicicleta equilibrada e estável em todas as simulações de maneira eficiente em quase todos os casos, com a menor taxa de sucesso sendo de 81%.

Velocidade objetivo	Peso da velocidade	Erro médio (cm)	Taxa de sucesso média	Tempo médio (s)	Erro máximo (cm)
1,5m/s	0,05	6,21	1,00	37,50	17,20
	0,10	6,08	1,00	35,45	16,72
	0,15	6,25	1,00	35,12	18,34
	0,20	6,19	1,00	34,50	18,31
2,5m/s	0,05	6,48	1,00	24,55	33,52
	0,10	6,54	1,00	22,52	22,76
	0,15	6,89	1,00	21,95	21,85
	0,20	7,52	1,00	21,68	39,54
3,5m/s	0,05	7,20	1,00	19,39	27,32
	0,10	7,48	1,00	17,63	28,01
	0,15	7,87	0,95	16,29	57,85
	0,20	8,12	0,95	15,93	46,68
4,5m/s	0,05	7,61	1,00	16,81	27,52
	0,10	8,96	1,00	15,57	51,04
	0,15	9,67	0,95	14,21	54,28
	0,20	9,85	0,81	11,69	67,68

Tabela 5.2: Tabela de resultados agregados do cenário de linha reta - Simulação tempo real.

As 6 falhas observadas ocorreram nos casos de velocidade alvo de 3,5m/s e 4,5m/s, com pesos da velocidade de 0,15 e 0,20. Uma possível causa para isso é que o termo de velocidade na função de custo se torna grande demais, resultando em soluções que priorizam alcançar a velocidade alvo mais rapidamente, mas levam à queda. No entanto, a taxa de sucesso média para esses casos foi de 95% nos casos 3,5m/s e 0,15, 3,5m/s e 0,2 e 4,5m/s e 0,15, e 81% para o caso 4,5m/s e 0,2, e portanto ainda apresenta uma eficiência razoável mesmo nesses casos mais desafiadores.

Em relação ao erro máximo, percebe-se a mesma tendência do erro médio de trajetória, com valores maiores para velocidades alvo maiores e pesos da velocidade maiores, mas parece estar mais correlacionado com o peso da velocidade do que com a velocidade alvo. Os valores ficaram entre 16 cm e 68 cm, aproximadamente, com os maiores valores associados aos pesos 0,15 e 0,2. A Figura 5.2 mostra todas as trajetórias geradas para o cenário de linha reta, agrupadas por velocidade alvo, e ilustra a tendência de erros maiores e mais falhas para velocidades alvo maiores.

Na Figura 5.3 é apresentado um mapa de calor para a taxa de sucesso média no cenário de linha reta, agrupado por velocidade alvo e peso da velocidade. Conforme

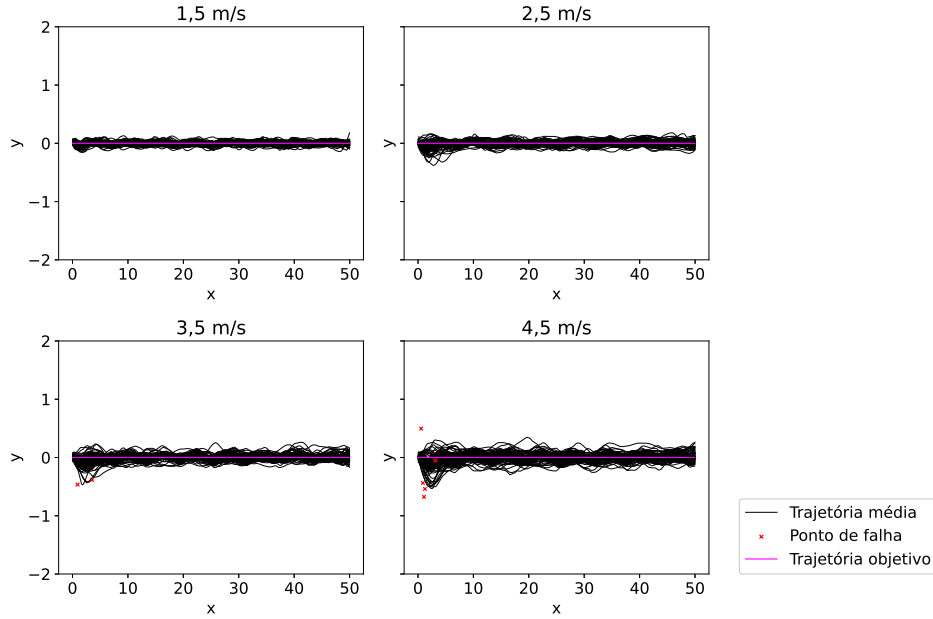


Figura 5.2: Trajetórias geradas para o cenário de linha reta, agrupadas por velocidade alvo. Fonte: do autor.

discutido anteriormente, o resultado mostra uma queda de eficiência para os casos onde a velocidade alvo é 3,5 m/s ou 4,5 m/s e o peso é 0,15 ou 0,2.

Em resumo, a principal dificuldade nesse cenário foi o início do movimento. Quando a bicicleta atinge a velocidade alvo, bastam pequenos ajustes para mantê-la na trajetória desejada.

5.3 Cenário de zigue-zague

O cenário de zigue-zague é um pouco mais desafiador que o cenário de linha reta, já que envolve mudanças rápidas de direção. A Figura 5.4 mostra o resultado para esse cenário. A escala do eixo y foi aumentada para facilitar a visualização.

É possível identificar que, assim como no cenário de linha reta, algumas falhas ocorrem logo no início da trajetória. Porém, a grande maioria ocorreu durante as curvas, especialmente no ápice, onde a inclinação da bicicleta é máxima. A análise das execuções mostra que a maioria das quedas é do tipo *low-side*, isto é, a bicicleta tomba para o lado de dentro da curva, por conta de velocidade insuficiente para manter o equilíbrio.

A Tabela 5.3 mostra os resultados para todas as execuções em tempo real desse cenário, agrupados por velocidade alvo e peso da velocidade. Em relação ao cenário de

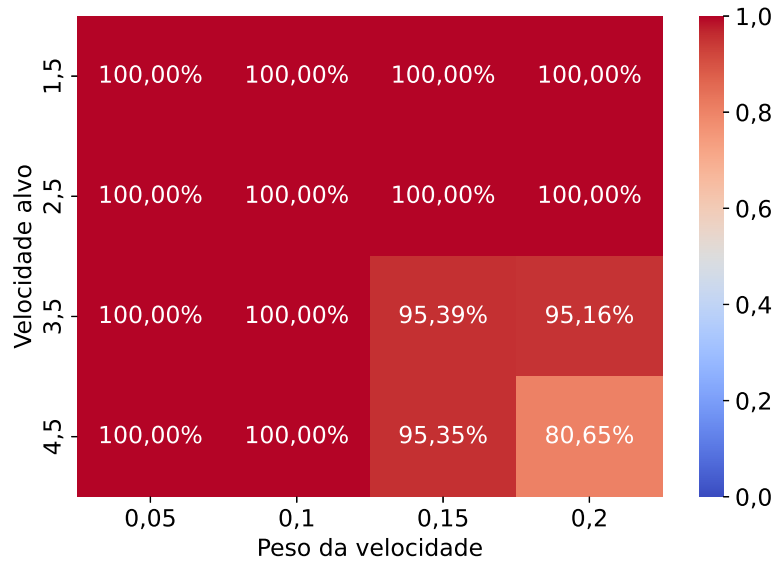


Figura 5.3: Mapa de calor para a taxa de sucesso média no cenário de linha reta. Fonte: do autor.

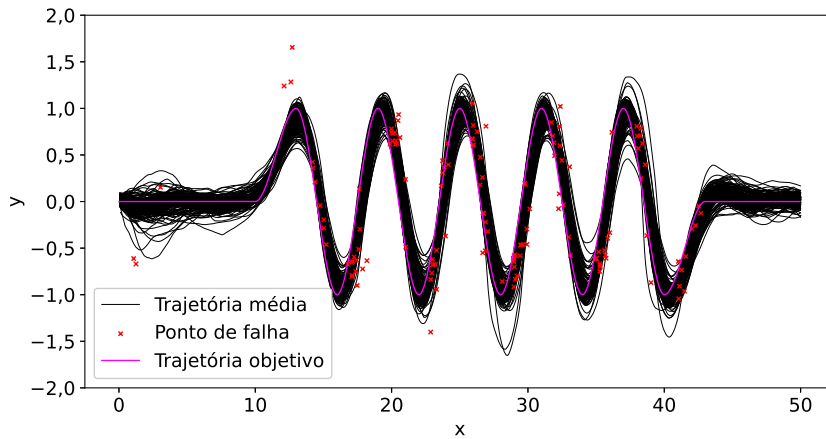


Figura 5.4: Resultados para o cenário de zigue-zague. Fonte: do autor.

linha reta, o cenário de zigue-zague apresentou taxas de sucesso menores em quase todos os casos, com a menor taxa sendo de 63% para o caso de velocidade alvo de 4,5m/s e peso da velocidade de 0,15. O erro médio de trajetória também foi maior, variando entre 6cm e 23cm, aproximadamente. Assim como no cenário anterior, o erro máximo tende a ser maior para velocidades alvo maiores e pesos da velocidade maiores, com valores entre 22cm e 94cm.

Apesar do resultado geral pior, a combinação de velocidade alvo de 4,5 m/s e peso da velocidade de 0,05 apresentou uma taxa de sucesso de 99%, com erro médio de apenas 14,75 cm e erro máximo de 58,07 cm. Esse resultado mostra que o controlador

Velocidade objetivo	Peso da velocidade	Erro médio (cm)	Taxa de sucesso média	Tempo médio (s)	Erro máximo (cm)
1,5m/s	0,05	6,24	0,74	35,45	22,48
	0,10	7,76	0,66	30,59	67,40
	0,15	8,55	0,79	34,90	32,43
	0,20	9,39	0,76	32,82	30,17
2,5m/s	0,05	8,83	0,81	26,42	44,54
	0,10	11,16	0,84	25,25	37,79
	0,15	12,81	0,82	23,54	45,49
	0,20	14,38	0,74	20,87	48,00
3,5m/s	0,05	11,14	0,83	22,29	41,63
	0,10	15,17	0,90	21,97	60,21
	0,15	16,97	0,69	16,52	67,85
	0,20	18,09	0,82	18,99	76,61
4,5m/s	0,05	14,75	0,99	23,81	58,07
	0,10	19,08	0,79	18,02	87,05
	0,15	20,43	0,63	13,90	93,68
	0,20	22,53	0,72	15,50	94,15

Tabela 5.3: Tabela de resultados agregados do cenário de zigue-zague - Simulação tempo real.

desenvolvido é muito sensível aos parâmetros, e que, realizando um processo de ajuste fino para encontrar os melhores parâmetros, é possível obter bons resultados.

A Figura 5.5 mostra todas as trajetórias agrupadas por velocidade alvo, e ilustra a tendência de erros maiores para velocidades alvo maiores. Podemos perceber que, para velocidades altas, existe uma tendência de a bicicleta escapar da curva e corrigir posteriormente, causando os altos erros máximos observados.

Por fim, a Figura 5.6 mostra um mapa de calor para a taxa de sucesso média no cenário de zigue-zague, agrupado por velocidade alvo e peso da velocidade. Assim como no cenário de linha reta, é possível observar uma região de máximo local, onde o controlador atinge a taxa de sucesso máxima, que se degrada conforme nos afastamos dessa região. No caso desse cenário, essa região não é tão clara, e parece conter mais de um máximo local.

Nesse cenário, os resultados indicam que existe uma velocidade alvo ideal para a trajetória especificada. Isso faz sentido, já que existe uma correlação entre o ângulo de inclinação e a velocidade durante uma curva e, portanto, a velocidade deve ser ajustada considerando o raio da curva. Os vídeos das simulações mostram que a principal causa

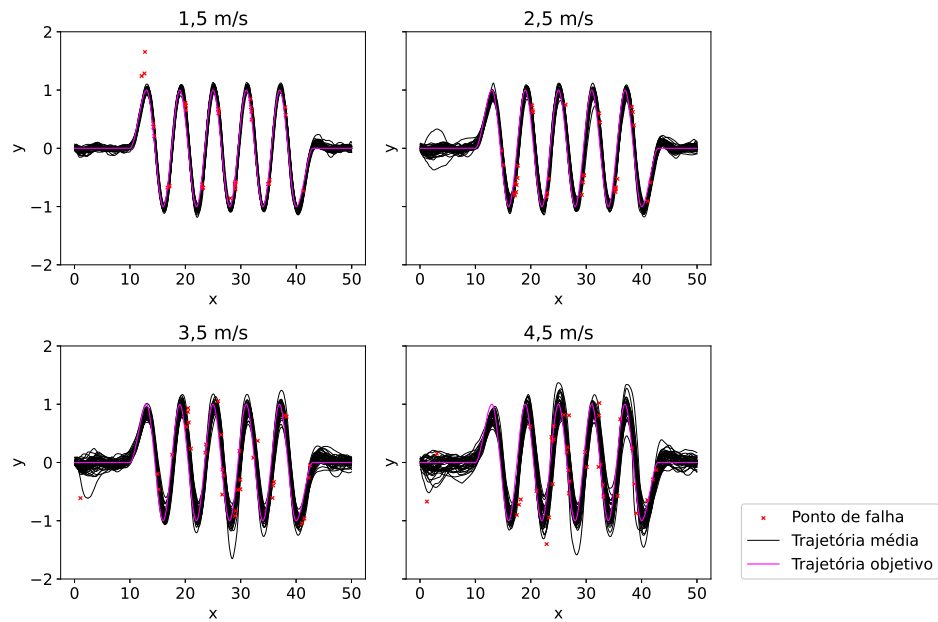


Figura 5.5: Trajetórias geradas para o cenário de zigzag, agrupadas por velocidade alvo. Fonte: do autor.

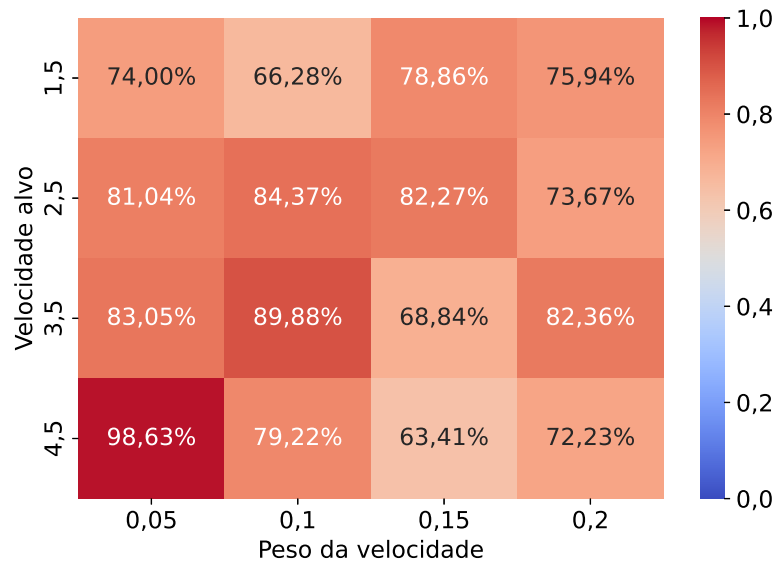


Figura 5.6: Mapa de calor para a taxa de sucesso média no cenário de zigue-zague. Fonte: do autor.

das falhas parece ser, de fato, uma velocidade incompatível com a trajetória.

5.4 Cenário de terreno acidentado

O cenário de terreno acidentado é um incremento em dificuldade em relação aos cenários anteriores. A instabilidade da bicicleta é maior, já que o terreno gera perturbações cons-

tantes, e o controlador precisa ajustar a direção da bicicleta a todo momento para evitar a queda. A Figura 5.7 mostra o resultado para esse cenário.

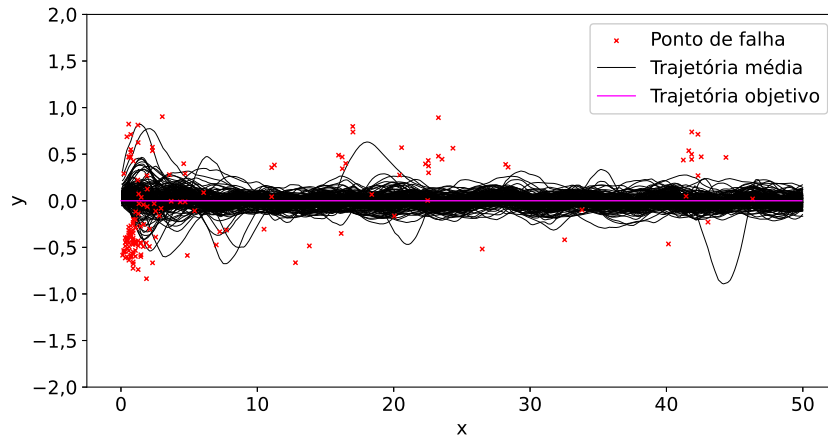


Figura 5.7: Resultados para o cenário de terreno acidentado. Fonte: do autor.

Neste cenário, existe uma concentração grande de falhas logo no início da trajetória. Começar a pedalar em um terreno acidentado é um desafio ainda maior do que começar em um terreno plano, já que é necessário mais esforço nos pedais para superar a inclinação na direção do movimento, além de ajustes constantes na direção do guidão e posicionamento do corpo para manter o equilíbrio. Essa dificuldade, alinhada aos fatores já mencionados, contribui para o resultado observado.

A Tabela 5.4 mostra os resultados para todas as simulações em tempo real desse cenário, agrupados por velocidade alvo e peso da velocidade. Apesar da rota definida ser a mesma que no cenário em linha reta, os erros médio e máximo são similares ao do cenário de zigue-zague, estando no intervalo de 6cm a 14cm para o erro médio e 50cm a 115cm para o erro máximo, aproximadamente. As taxas de sucesso foram menores, variando entre 27% no pior caso e 95% no melhor.

A Figura 5.8 mostra todas as trajetórias geradas para o cenário de terreno acidentado, agrupadas por velocidade alvo. Nesse caso, a variação das trajetórias não aumenta tanto com a velocidade alvo, se mantendo relativamente constante para os casos de 2,5m/s, 3,5m/s e 4,5m/s. Uma possível explicação para isso é que o terreno já causa perturbações significativas em baixa velocidade, e o aumento da velocidade na verdade diminui a influência do terreno na trajetória, tornando o erro mais constante.

A Figura 5.9 mostra um mapa de calor para a taxa de sucesso média no cenário de

Velocidade objetivo	Peso da velocidade	Erro médio (cm)	Taxa de sucesso média	Tempo médio (s)	Erro máximo (cm)
1,5m/s	0,05	7,21	0,27	14,29	60,15
	0,10	6,87	0,71	29,44	74,45
	0,15	6,98	0,76	29,99	72,17
	0,20	7,47	0,72	27,27	89,50
2,5m/s	0,05	7,45	0,60	18,57	59,00
	0,10	8,07	0,95	25,66	47,50
	0,15	8,22	0,85	21,30	68,08
	0,20	8,42	0,61	15,09	74,18
3,5m/s	0,05	8,38	0,95	23,32	77,49
	0,10	10,01	0,57	12,45	68,08
	0,15	10,35	0,52	10,94	84,06
	0,20	12,54	0,31	6,64	116,37
4,5m/s	0,05	8,93	0,56	12,46	61,53
	0,10	11,21	0,38	7,91	91,05
	0,15	13,66	0,37	7,56	113,25
	0,20	13,23	0,43	8,55	74,38

Tabela 5.4: Tabela de resultados agregados do cenário de terreno acidentado - Simulação tempo real.

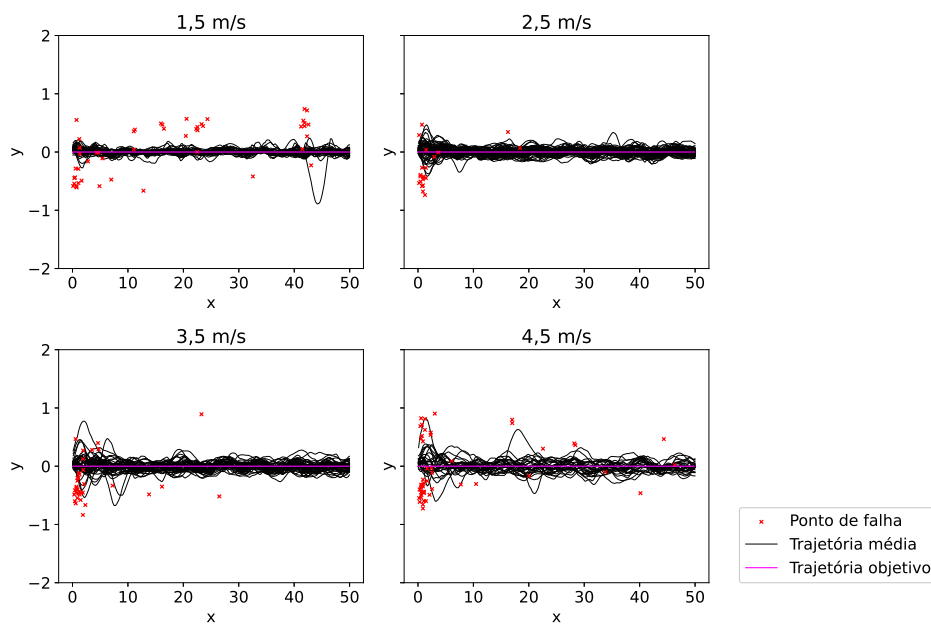


Figura 5.8: Trajetórias geradas para o cenário de terreno acidentado, agrupadas por velocidade alvo. Fonte: do autor.

terreno acidentado, agrupado por velocidade alvo e peso da velocidade. Assim como nos cenários anteriores, é possível observar uma região de máximo local, onde o controlador atinge a taxa de sucesso máxima. Nesse caso, o melhor resultado foi obtido para a

combinação de velocidade alvo de 3,5 m/s e peso da velocidade de 0,05, com uma taxa de sucesso de 95,26%, e para a velocidade alvo de 2,5 m/s e peso da velocidade de 0,1, com uma taxa de sucesso de 95,16%.

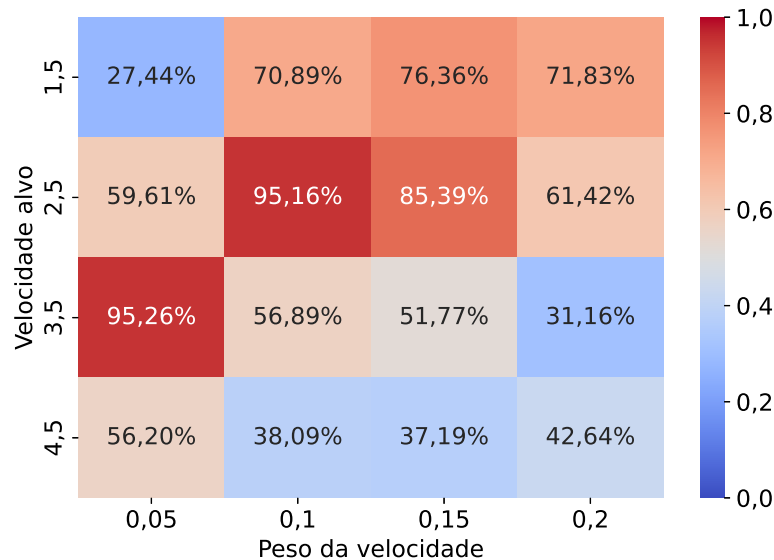


Figura 5.9: Mapa de calor para a taxa de sucesso média no cenário de terreno acidentado. Fonte: do autor.

Portanto, os resultados mostram que esse cenário é significativamente mais sensível à escolha dos parâmetros do controlador, e que a taxa de sucesso pode variar bastante dependendo dos parâmetros escolhidos. Ainda assim, foi possível encontrar combinações de parâmetros que resultaram em taxas de sucesso acima de 95%.

Os resultados no cenário de terreno acidentado evidenciam que o controlador é capaz de lidar com perturbações contínuas causadas pelo terreno, mas com um impacto perceptível na taxa de sucesso e na precisão. A análise das trajetórias revelou que as perturbações geradas pelo terreno exigem ajustes frequentes no controle do guidão, e o controlador frequentemente reage de forma tardia em velocidades mais altas, resultando em quedas. Além disso, em velocidades mais baixas a irregularidade do terreno é mais perceptível, e o controlador tem dificuldade em manter a trajetória.

5.5 Cenário de escadaria

O cenário de escadaria é especialmente desafiador, já que a manobra de descer uma série de degraus com uma bicicleta é uma tarefa complexa mesmo para um humano. A Figura

5.10 mostra o resultado para este cenário. Podemos observar aglomerações de falhas nas regiões dos degraus, com poucas falhas ocorrendo na região plana de transição entre os andares. Além disso, uma análise inicial indica que as trajetórias foram muito mais instáveis.

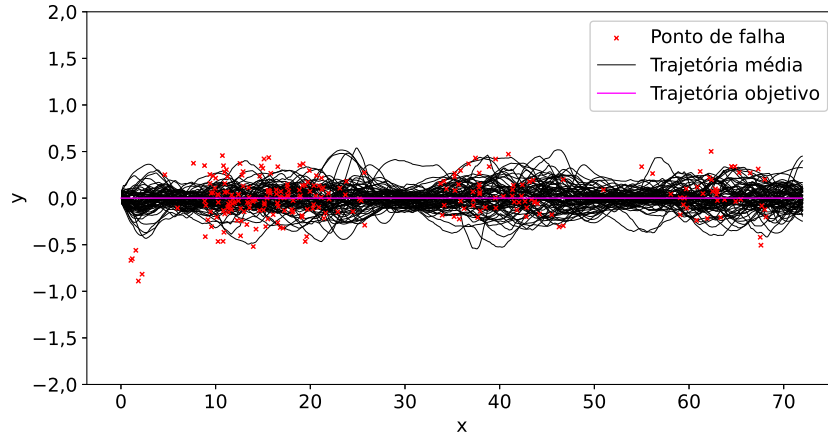


Figura 5.10: Resultados para o cenário de escadaria. Fonte: do autor.

A Tabela 5.5 mostra os resultados para todas as simulações em tempo real desse cenário, agrupados por velocidade alvo e peso da velocidade. Como imaginado, os erros médios para esse cenário foram maiores que para os cenários anteriores, com o menor erro médio sendo de 10,68 cm e o maior sendo de 14,28 cm. Já os erros máximos se mantiveram similares aos do cenário de zigzag, variando entre 46 cm e 89 cm. As taxas de sucesso foram as menores de todos os cenários, variando entre 23% e 76%.

A Figura 5.11 mostra todas as trajetórias geradas para o cenário de escadaria, agrupadas por velocidade alvo. Nenhuma execução para 1,5m/s foi completada, com a grande maioria tendo falhado no primeiro lance de degraus. Assim como no cenário de terreno acidentado, a variação das trajetórias não aumenta tanto com a velocidade alvo, se mantendo relativamente constante para os casos de 2,5m/s, 3,5m/s e 4,5m/s. A Figura 5.12 mostra um mapa de calor para a taxa de sucesso média no cenário de escadaria, agrupado por velocidade alvo e peso da velocidade. Assim como nos cenários anteriores, é possível observar uma região de máximo local, onde o controlador atinge a taxa de sucesso máxima.

A Figura 5.13 mostra as trajetórias geradas para o cenário de escadaria em uma visão lateral do cenário. A linha correspondente à velocidade alvo de 3,5 m/s apresenta os

Velocidade objetivo	Peso da velocidade	Erro médio (cm)	Taxa de sucesso média	Tempo médio (s)	Erro máximo (cm)
1,5m/s	0,05	11,17	0,28	12,90	68,04
	0,10	11,08	0,24	9,77	67,68
	0,15	12,23	0,24	9,86	72,92
	0,20	12,61	0,23	9,74	70,66
2,5m/s	0,05	11,77	0,41	13,51	58,75
	0,10	12,86	0,52	16,63	66,22
	0,15	12,52	0,57	17,88	49,54
	0,20	12,93	0,54	17,17	65,23
3,5m/s	0,05	10,68	0,72	17,57	51,75
	0,10	11,72	0,74	17,65	49,51
	0,15	13,27	0,76	18,06	52,81
	0,20	12,47	0,74	16,77	46,64
4,5m/s	0,05	12,46	0,67	13,86	51,89
	0,10	12,82	0,61	12,13	56,43
	0,15	13,65	0,57	11,13	65,17
	0,20	14,28	0,44	8,50	89,10

Tabela 5.5: Tabela de resultados agregados do cenário de escadaria - Simulação tempo real.

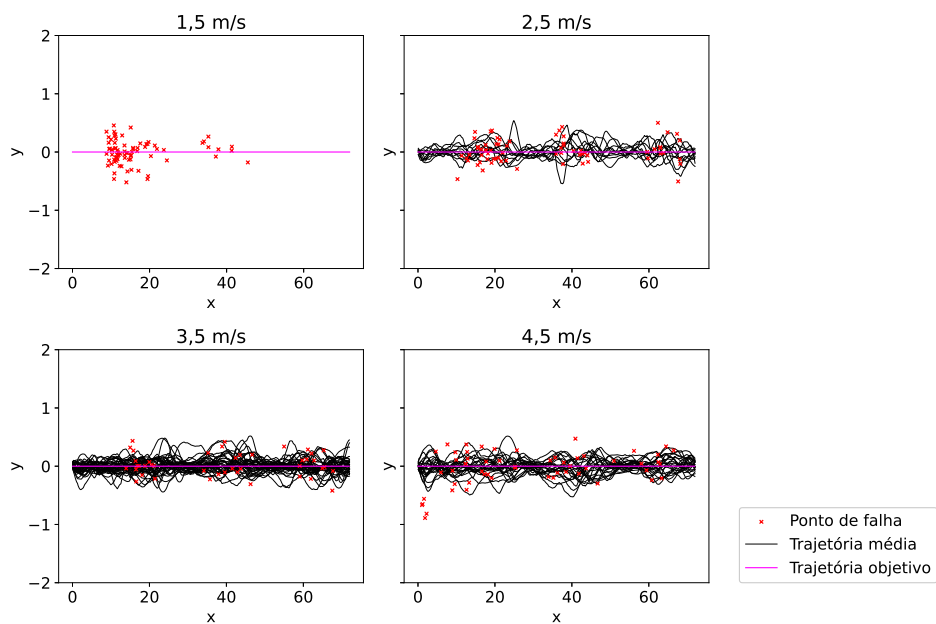


Figura 5.11: Trajetórias geradas para o cenário de escadaria, agrupadas por velocidade alvo. Fonte: do autor.

melhores resultados, com taxas de sucesso acima de 70% para todos os pesos da velocidade. A combinação de velocidade alvo de 3,5 m/s e peso da velocidade de 0,15 apresentou a melhor taxa de sucesso, de 76%. Os resultados pioram significativamente para velocidades

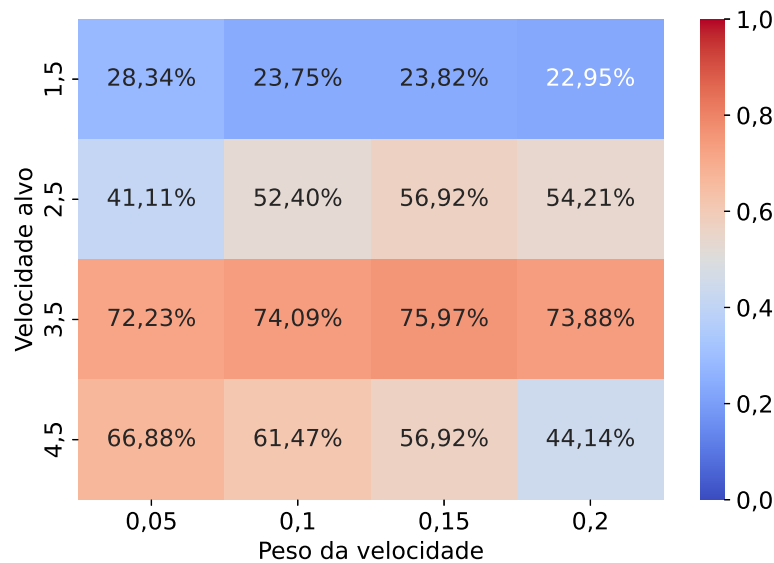


Figura 5.12: Mapa de calor para a taxa de sucesso média no cenário de escadaria. Fonte: do autor.

alvo maiores ou menores, e, portanto, a escolha da velocidade alvo é crucial para o sucesso desse cenário.

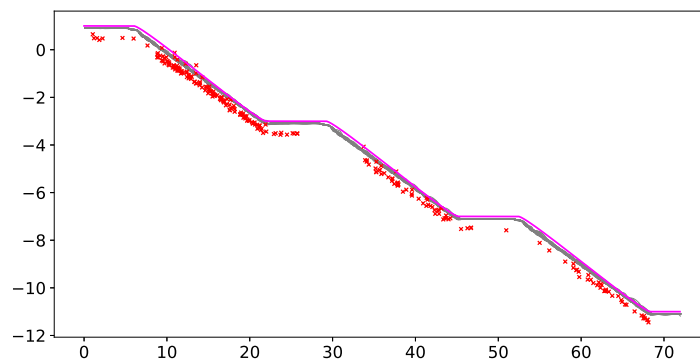


Figura 5.13: Visão lateral das trajetórias geradas para o cenário de escadaria. Fonte: do autor.

5.6 Cenário de circuito de obstáculos

O cenário de circuito de obstáculos é uma combinação de diversos desafios, sendo uma tentativa de avaliar o desempenho em um cenário mais complexo. A Figura 5.14 mostra o resultado para este cenário, onde é possível identificar claramente as áreas com diferentes características geométricas. É possível observar 4 pontos principais de falha neste cenário:

a sequência de zigue-zague, a rampa, o lance de degraus e, assim como nos outros cenários, o início. A Tabela 5.6 mostra os resultados para todas as simulações em tempo real deste cenário, agrupados por velocidade alvo e peso da velocidade.

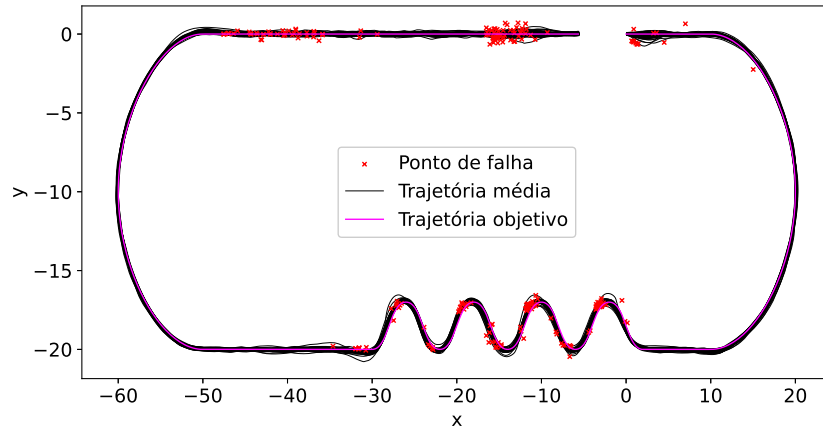


Figura 5.14: Resultados para o cenário de circuito de obstáculos. Fonte: do autor.

Velocidade objetivo	Peso da velocidade	Erro médio (cm)	Taxa de sucesso média	Tempo médio (s)	Erro máximo (cm)
1,5m/s	0,05	7,83	0,57	78,11	66,16
	0,10	8,96	0,64	85,25	47,36
	0,15	10,20	0,65	89,78	64,49
	0,20	11,14	0,58	81,33	60,17
2,5m/s	0,05	9,84	0,75	74,21	53,62
	0,10	11,52	0,66	57,61	47,97
	0,15	12,86	0,71	61,83	76,59
	0,20	13,39	0,67	57,76	64,85
3,5m/s	0,05	11,84	0,74	60,39	68,68
	0,10	13,75	0,81	58,58	53,66
	0,15	14,85	0,71	49,30	56,33
	0,20	15,84	0,56	38,09	62,35
4,5m/s	0,05	13,34	0,84	58,74	57,14
	0,10	16,06	0,70	44,52	83,26
	0,15	18,05	0,67	41,11	96,95
	0,20	19,34	0,64	38,24	94,55

Tabela 5.6: Tabela de resultados agregados do cenário de circuito - Simulação tempo real.

O comportamento e intervalo dos erros médio e máximo se mantiveram similares aos cenários anteriores, com valores entre 8 e 20 cm e 47 e 97 cm, respectivamente. As taxas de sucesso, no entanto, foram relativamente baixas, variando entre 57% e 84%. A Figura 5.15 mostra todas as trajetórias geradas para o cenário de circuito de obstáculos,

agrupadas por velocidade alvo. Para a velocidade alvo de 1,5 m/s e 2,5 m/s, existe uma grande concentração de falhas na rampa. Esse resultado pode ser um indicativo de que o momento adquirido pela bicicleta em velocidades mais altas auxilia na subida, e a natureza míope do método não consegue prever a necessidade de uma velocidade maior para superar esse obstáculo.

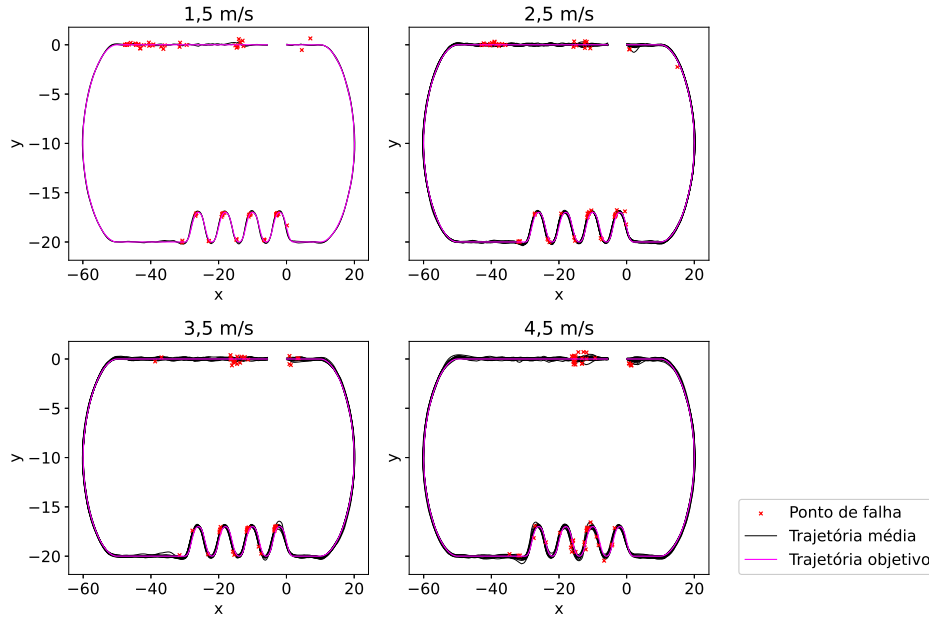


Figura 5.15: Trajetórias geradas para o cenário de circuito de obstáculos, agrupadas por velocidade alvo. Fonte: do autor.

Já a Figura 5.16 mostra as trajetórias geradas para o cenário de circuito de obstáculos em uma visão lateral do cenário. O eixo y está fora da escala para melhor visualização da elevação do terreno. É possível observar que a maioria das quedas que ocorrem na rampa se encontram na primeira metade da subida, o que é mais um indicativo da hipótese anterior.

A Figura 5.17 mostra um mapa de calor para a taxa de sucesso média no cenário de circuito de obstáculos, agrupado por velocidade alvo e peso da velocidade. Assim como nos cenários anteriores, é possível observar uma região de máximo local, que nesse cenário parece ser para velocidades alvo maiores e pesos da velocidade menores. A combinação de velocidade alvo de 4,5 m/s e peso da velocidade de 0,05 apresentou a melhor taxa de sucesso, de 84,16%.

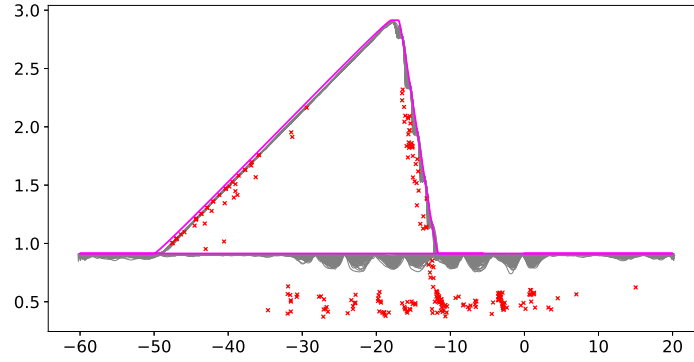


Figura 5.16: Visão lateral das trajetórias geradas para o cenário de circuito de obstáculos. Fonte: do autor.

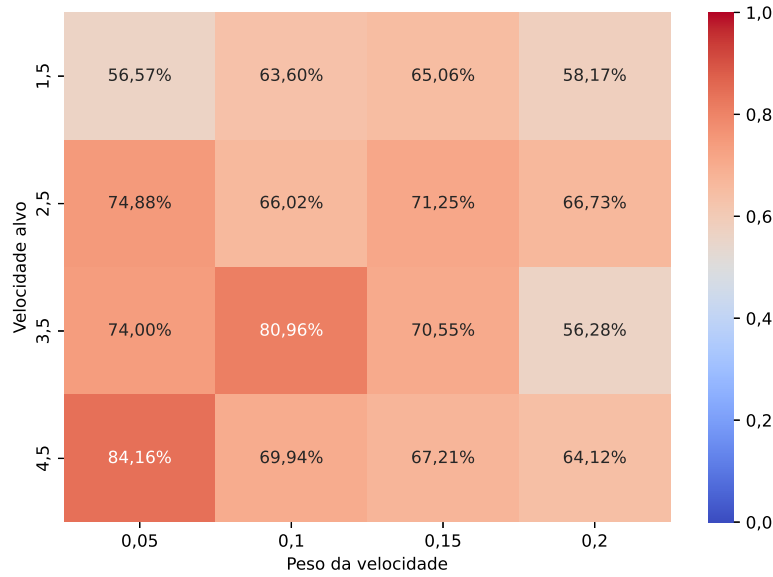


Figura 5.17: Mapa de calor para a taxa de sucesso média no cenário de circuito de obstáculos. Fonte: do autor.

5.7 Discussão dos resultados

A Figura 5.18 mostra o número de trajetórias finalizadas para cada um dos cenários. O controlador desenvolvido apresentou bom desempenho em cenários mais simples e com rotas regulares, como o de linha reta, zigue-zague e terreno acidentado, mas teve dificuldades em cenários mais dinâmicos, como o de escadaria e circuito de obstáculos. Uma possível explicação para isso é a função de custo utilizada, que fixa a velocidade alvo em um valor constante, o que acaba não funcionando bem para esses casos, onde a velocidade ideal varia.

Além disso, a natureza míope do MPC pode ser um fator limitante, já que o

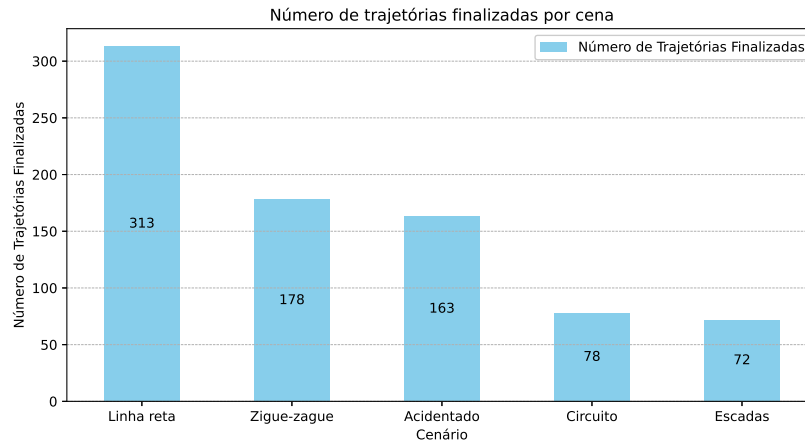


Figura 5.18: Taxa de sucesso para cada cenário de teste. Fonte: do autor.

controlador não consegue prever alguns obstáculos. Por exemplo, ao subir a rampa no cenário de circuito, a velocidade da bicicleta fica reduzida, e o horizonte de 1,2 segundos passa a incluir somente um comprimento pequeno da rota no futuro, o que pode ser uma causa de falha na escadaria que se segue. O peso da velocidade menor parece dar mais liberdade para o controlador ajustar a velocidade dependendo do obstáculo, o que pode ser uma explicação para um melhor desempenho.

Em termos de qualidade visual do movimento do humanoide, os resultados obtidos aproximam um comportamento realista, permitindo a simulação de um ciclista operando o modelo de bicicleta. No entanto, em alguns momentos, principalmente em mudanças bruscas de direção ou ao lidar com perturbações, o humanoide apresentou poses que não são totalmente naturais. Isso sugere que a abordagem utilizada poderia ser melhorada para capturar de forma mais realista os movimentos esperados de um ciclista real e assim permitir o uso do método em aplicações que requerem um alto nível de fidelidade visual.

Em relação aos resultados de Tan et al. (2014), podemos afirmar que o realismo de movimento alcançado foi superior, já que nesse trabalho o movimento do ciclista foi extensivamente restringido, o que ocasionou uma movimentação rígida e artificial. Além disso, o método MPC não requer aprendizado, e todos os resultados foram coletados de simulações em tempo real sem nenhum conhecimento prévio pelo controlador. Já o método de aprendizado por reforço profundo, apesar de ter alcançado resultados bons para o cenários mostrados, demanda um treinamento extensivo e específico para cada cenário.

Já em relação aos resultados de Cook (2004), o método proposto nesse trabalho é significativamente mais flexível e adaptável, já que o MPC não requer um modelo de bicicleta específico e pode ser aplicado a qualquer tipo de veículo de duas rodas. Além disso, o método de Cook (2004) não considera a dinâmica do ciclista, o que pode limitar a aplicabilidade do método em cenário mais complexos.

Para aprimorar a naturalidade do movimento, algumas melhorias podem ser exploradas em trabalhos futuros. Uma possibilidade é a introdução de um termo adicional na função de custo do MPC, que penalize posturas excessivamente artificiais ou que favoreça transições mais suaves entre movimentos. Além disso, a incorporação de dados de captura de movimento reais poderia fornecer referências mais precisas para os padrões esperados de um ciclista humano, melhorando ainda mais a qualidade da simulação.

6 Conclusão

Neste trabalho, foi desenvolvido e analisado um controlador baseado no método de Controle Preditivo Baseado em Modelo (MPC) para a simulação de um veículo de duas rodas com um piloto humanoide no motor de física MuJoCo. O objetivo foi avaliar a capacidade do controlador em manter a estabilidade e seguir trajetórias pré-definidas em diferentes cenários, variando parâmetros de velocidade e peso na função de custo.

Os resultados obtidos demonstraram que o MPC foi capaz de conduzir a bicicleta em diversos cenários, especialmente em terrenos planos e trajetórias retilíneas. A taxa de sucesso foi alta nesses casos, e o erro médio de trajetória permaneceu em valores proporcionalmente baixos para a escala das rotas consideradas. Entretanto, desafios adicionais surgiram em terrenos acidentados, trajetórias em zigue-zague e na descida de escadarias, onde a complexidade do problema aumentou significativamente.

A análise dos diferentes cenários evidenciou que a escolha dos parâmetros do controlador influencia diretamente no desempenho. Para velocidades alvo mais altas, houve um aumento no erro de trajetória e uma tendência maior a quedas, especialmente em curvas acentuadas. Em contrapartida, velocidades mais baixas comprometeram a estabilidade em terrenos irregulares e na descida de escadas, demonstrando que há uma faixa ideal de velocidade para cada tipo de ambiente. O peso atribuído à velocidade na função de custo também teve impacto significativo, influenciando a estabilidade e o comportamento do piloto virtual durante a condução.

Outro ponto relevante identificado foi a dificuldade inicial do controlador em encontrar uma solução eficiente para iniciar o movimento. Muitas das falhas ocorreram nos primeiros instantes da simulação, quando o veículo partia do repouso. Esse aspecto sugere que estratégias de inicialização podem ser incorporadas ao método para melhorar a transição do estado de repouso para o deslocamento.

Além disso, os testes demonstraram que, apesar do sucesso na maioria dos casos, o controlador enfrentou dificuldades na adaptação a obstáculos complexos e mudanças bruscas no terreno. Isso indica que métodos híbridos, combinando MPC com técnicas

de aprendizado por reforço ou ajustes automáticos de parâmetros, poderiam melhorar a robustez e a eficiência do sistema.

Como sugestões para trabalhos futuros, algumas direções podem ser exploradas:

- investigar o uso de MPC adaptativo, ajustando dinamicamente os parâmetros do controlador com base no contexto da simulação;
- incorporar aprendizado por reforço para melhorar a capacidade de adaptação do piloto a diferentes cenários e otimizar estratégias de controle;
- explorar estratégias de inicialização para reduzir o tempo necessário para encontrar trajetórias estáveis desde o repouso;
- avaliar a implementação do controlador em um ambiente físico real, utilizando sensores e atuadores para validar os resultados da simulação;
- melhorar a modelagem do piloto humanoide, permitindo maior flexibilidade e comportamento mais natural durante a condução.

Dessa forma, este trabalho contribui para o estudo de controle de veículos de duas rodas em ambientes simulados, evidenciando a viabilidade do MPC para a tarefa proposta e identificando desafios e oportunidades para futuras melhorias.

Bibliografia

- BALTES, J.; CHRISTMANN, G.; SAEEDVAND, S. A deep reinforcement learning algorithm to control a two-wheeled scooter with a humanoid robot. *Engineering Applications of Artificial Intelligence*, Elsevier, v. 126, p. 106941, 2023.
- CARVALLO, E. *Théorie du mouvement du monocycle et de la bicyclette...* [S.l.]: Gauthier-Villars, 1901.
- COOK, M. It takes two neurons to ride a bicycle. *Demonstration at NIPS*, Citeseer, v. 4, 2004.
- DOEHRING, E. *Über die Stabilität und die Lenkkräfte von Einspurfahrzeugen*. Tese (Doutorado) — Technische Hochschule Braunschweig, 1953.
- GETZ, N.; MARSDEN, J. Control for an autonomous bicycle. In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 1995. v. 2, p. 1397–1402 vol.2.
- HERLIHY, D. V. *Bicycle: the history*. [S.l.]: Yale University Press, 2004.
- HOWELL, T. et al. Predictive Sampling: Real-time Behaviour Synthesis with MuJoCo. dec 2022. Disponível em: <https://arxiv.org/abs/2212.00541>.
- KAUP, M. et al. A review of nine physics engines for reinforcement learning research. *arXiv preprint arXiv:2407.08590*, 2024.
- KLEIN, F.; SOMMERFELD, A. *Über die theorie des kreisels*. [S.l.]: BG Teubner, 1897. v. 1.
- MIKKOLA, E. Online planning and control of physics-based skateboarding animation. 2024.
- PICOTTI, E.; BIANCHIN, F.; BRUSCHETTA, M. Real-time learning-based nonlinear model predictive control of a virtual motorcycle employing grey-box modeling through gaussian processes. *Control Engineering Practice*, Elsevier, v. 144, p. 105837, 2024.
- RANDLØV, J.; ALSTRØM, P. Learning to drive a bicycle using reinforcement learning and shaping. In: *ICML*. [S.l.: s.n.], 1998. v. 98, p. 463–471.
- SANCHEZ, N. C.; PASTOR, L. A.; LARSON, K. Autonomous bicycles: A new approach to bicycle-sharing systems. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.: s.n.], 2020. p. 1–6.
- SCHWAB, A.; MEIJAARD, J. A review on bicycle dynamics and rider control. *Vehicle System Dynamics*, v. 51, 07 2013.
- TAI, S.-Y. T. T.; VERAART, F. Making the bicycle dutch: The development of the bicycle industry in the netherlands, 1860–1940. In: *Invisible bicycle*. [S.l.]: Brill, 2019. p. 126–155.

TAN, J. et al. Learning bicycle stunts. *ACM Transactions on Graphics (TOG)*, ACM New York, NY, USA, v. 33, n. 4, p. 1–12, 2014.

TASSA, Y.; EREZ, T.; TODOROV, E. Synthesis and stabilization of complex behaviors through online trajectory optimization. In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.: s.n.], 2012. p. 4906–4913.

TODOROV, E.; EREZ, T.; TASSA, Y. Mujoco: A physics engine for model-based control. In: IEEE. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.], 2012. p. 5026–5033.

WHIPPLE, F. J. The stability of the motion of a bicycle. *Quarterly Journal of Pure and Applied Mathematics*, v. 30, n. 120, p. 312–348, 1899.

WILSON, D. G.; SCHMIDT, T. *Bicycling science*. [S.l.]: MIT press, 2020.

ZHU, X. et al. Online series-parallel reinforcement-learning-based balancing control for reaction wheel bicycle robots on a curved pavement. *IEEE Access*, IEEE, 2023.