

UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO DE JOINVILLE
CURSO DE ENGENHARIA MECATRÔNICA

PAULO HENRIQUE KUBIACK GORLA

SISTEMA AVANÇADO DE ASSISTÊNCIA AO MOTORISTA PARA EFICIÊNCIA
ENERGÉTICA DE VEÍCULOS EM RUAS INCLINADAS UTILIZANDO ALGORITMO
ATOR-CRÍTICO

Joinville
2023

PAULO HENRIQUE KUBIACK GORLA

SISTEMA AVANÇADO DE ASSISTÊNCIA AO MOTORISTA PARA EFICIÊNCIA
ENERGÉTICA DE VEÍCULOS EM RUAS INCLINADAS UTILIZANDO ALGORITMO
ATOR-CRÍTICO

Trabalho de Conclusão de Curso apresentado
como requisito parcial para obtenção do título
de bacharel em Engenharia Mecatrônica
no curso de Engenharia Mecatrônica,
da Universidade Federal de Santa
Catarina, Centro Tecnológico de Joinville.

Orientador: Dr. Ricardo José Pfitscher

Joinville
2023

AGRADECIMENTOS

Agradeço ao professor doutor Ricardo José Pfitscher pela orientação na execução deste trabalho. Agradeço sinceramente aos meus familiares, amigos e colegas pela constante colaboração e suporte ao longo desse processo.

RESUMO

A economia de combustível é um tema de amplo interesse e discussão por consumidores, indústria e governos. Enquanto proprietários de veículos buscam equilibrar economia e desempenho, a indústria automotiva busca atender metas restritas de emissão de poluentes definidas por governos. Como resultado, boa parte dos veículos modernos incluem indicadores de eficiência na condução, tais como, consumo instantâneo, eficiência na condução em percurso e econômetros. Contudo, estes indicadores normalmente baseiam-se em métricas instantâneas, como por exemplo, a pressão no acelerador, sem considerar dados do trajeto, como aclives, declives e percursos anteriores. Considerando isso, este trabalho apresenta um sistema avançado de assistência ao motorista (ADAS), que recomenda ações para redução do consumo de combustível, em um percurso predefinido. Para tanto, a metodologia utilizada é a de aprendizado por reforço, mais especificamente algoritmo ator-crítico, que possibilita estimar o consumo apenas com informações já presentes no barramento Controller Area Network (CAN) e do Global Positioning System (GPS). Como resultado, o sistema foi capaz de, a partir dos dados de uma pista, indicar ações tomadas em cada trecho e, ao fim do percurso, obteve um aumento de 9.3% de eficiência quando comparado aos dados reais.

Palavras-chave: Consumo de Combustível. ADAS. A3C.

ABSTRACT

Fuel economy is a topic of broad interest and discussion by consumers, industry and governments. While vehicle owners pursue the equilibrium between economy and performance, the automotive industry seeks to meet restricted emission goals defined by governments. As a result, modern vehicles include efficiency indicators like instant consumption, conduction efficiency and econometers. On the other hand, these indicators are based on instant metrics, as the accelerator pressure, without taking account of the route data, as slopes and previous routes. In light of this, this paper introduces an Advanced Driver Assistance System (ADAS) that offers recommendations for reducing fuel consumption on a predefined route. To achieve this, the methodology employed is the reinforcement learning, more specifically the actor-critic algorithm, which allows to estimate the fuel consumption using information readily available on the Controller Area Network (CAN) and the Global Positioning System (GPS). The outcome was the ability to, based on data from a given track, suggest actions taken at each segment and, at the end of the journey, got a 9.3% improvement on the efficiency when compared to the real data.

Keywords: Fuel Consumption. ADAS. A3C.

LISTA DE FIGURAS

Figura 1 – Funções de Segurança e Conveniência Baseadas nos Sensores do Veículo	12
Figura 2 – Proposta de sistema de <i>eco-driving</i> usando predição baseada em modelo e informações do formato da pista	15
Figura 3 – Projeto do Sistema	21
Figura 4 – Coleta e Tratamento dos Dados	24
Figura 5 – Perfil de Altitude - Trajeto 287	25
Figura 6 – Fluxograma do Ambiente	26
Figura 7 – Recompensa por episódio <i>Worker 0</i> Ambiente 1	39
Figura 8 – Recompensa do melhor episódio Ambiente 1	39
Figura 9 – Recomendações Primeira Metade do Trajeto - Ambiente 1	40
Figura 10 – Recompensa por episódio <i>Worker 0</i> Ambiente 2	41
Figura 11 – Comparativo da Soma da Recompensa no Melhor Episódio	42
Figura 12 – Recomendações Primeira Metade do Trajeto - Ambiente 2	43

LISTA DE TABELAS

Tabela 1 – Parâmetros do Veículo	30
Tabela 2 – Ambientes de Treinamento	37
Tabela 3 – Comparativo	37
Tabela 4 – Ações - Ambiente 1	40

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	10
1.1.1	Objetivo Geral	10
1.1.2	Objetivos Específicos	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	SISTEMA AVANÇADO DE ASSISTÊNCIA AO MOTORISTA (ADAS)	12
2.2	CONSUMO ENERGÉTICO EM VEÍCULOS	14
2.2.1	Motores a Combustão	14
2.2.2	Motores Elétricos	14
2.2.3	Consumo Energético em uma Pista com Alteração de Nível	15
2.3	APRENDIZADO DE MÁQUINA	16
2.3.1	Aprendizado Supervisionado	16
2.3.2	Aprendizado Não-Supervisionado	16
2.3.3	Aprendizado Semi-Supervisionado	17
2.3.4	Aprendizado por Reforço	18
2.3.4.1	<u>Algoritmos Ator-crítico</u>	18
2.3.4.2	<u>Algoritmo <i>Asynchronous Advantage Actor-Critic</i> (A3C)</u>	19
2.3.4.3	<u>Ambiente de treinamento</u>	19
3	MATERIAIS E MÉTODOS	21
3.1	PROJETO DO SISTEMA	21
3.2	MODELOS DE ESTIMATIVA DE CONSUMO	22
3.2.1	Métodos Analíticos	22
3.2.2	Método de Aprendizado por Reforço	23
3.3	BASE DE DADOS E TRATAMENTO	24
3.4	HEURÍSTICA DO AMBIENTE	26
3.5	ALGORITMO A3C DESENVOLVIDO	30
4	RESULTADOS	36
4.1	Treinamento e Coleta de Resultados	36
4.2	Resultados	37
5	CONCLUSÕES	44
	REFERÊNCIAS	46
	ANEXO A	49

1 INTRODUÇÃO

A eficiência energética desempenha um papel crucial para os consumidores, tanto na escolha de um veículo quanto durante seu uso. Essa preocupação se intensifica diante do aumento nos preços dos combustíveis (HELFAND; WOLVERTON, 2009). Além disso, a relevância desse tema estende-se às fabricantes de automóveis, as quais precisam aderir a um conjunto de normas devido ao impacto que seus produtos causam na sociedade. Portanto, é imperativo que essas empresas considerem o consumo de combustível no processo de desenvolvimento de seus produtos. Emissões elevadas contribuem para o aumento do efeito estufa, concentração de gases, ameaças à saúde humana e danos aos sistemas infraestruturais nas esferas agrícola, ecológica e urbana (MICHALEK; PAPALAMBROS; SKERLOS, 2004).

Uma solução emergente para o desafio do consumo de combustíveis é representada pelos veículos elétricos (VEs). Apesar de enfrentarem obstáculos como custo e a dependência da sustentabilidade em relação à fonte de energia para recarga, eles são percebidos de maneira positiva em termos de custo-benefício (ALMANSOUR, 2022). No entanto, as fabricantes de VEs precisam aprimorar tanto os métodos de produção quanto a tecnologia empregada, visando estabelecer um equilíbrio positivo ao longo do ciclo de vida do veículo, desde a produção até o descarte (CUSENZA et al., 2019). A melhoria da eficiência dos veículos pode atenuar os desafios enfrentados pelas fabricantes e, para alcançar esse objetivo, é viável desenvolver sistemas que avaliem e aprimorem a eficiência energética dos veículos.

As abordagens propostas por Kamal et al. (2011, 2013) para estimar o consumo energético de um veículo indicam que é factível avaliar a eficiência com base em características intrínsecas do automóvel, como velocidade e aceleração instantâneas, além das condições da via, como sua inclinação. Estes parâmetros são perceptíveis do ponto de vista do condutor. No entanto, as ações específicas a serem tomadas para otimizar a eficiência do veículo não são explicitamente delineadas (KAMAL et al., 2011).

Frequentemente, as fabricantes de veículos incorporam em seus automóveis sistemas avançados de assistência ao motorista (ADAS - Advanced Driver Assistance System). Esses sistemas contribuem para aprimorar a compreensão do consumidor em relação ao veículo que está dirigindo, reduzindo ou prevenindo mal-entendidos das funções do veículo, aliviando o estresse associado à condução e simplificando a interação entre o veículo e o motorista por meio de uma interface facilitadora (LI, 2011). Além disso, os ADAS podem ser empregados para aumentar a eficiência energética dos veículos (TUNNELL et al., 2018).

Estudos de pesquisa indicam que a aplicação de aprendizado por reforço pode

ser benéfica na busca por uma direção mais eficiente (PERERA; KAMALARUBAN, 2021). O sistema proposto por Liu et al. (2020) alcançou uma redução significativa, entre 17% e 24%, no consumo em comparação com a base utilizada. Em Wu et al. (2021), a implementação de um algoritmo ator-crítico resultou em um aumento de 23,3% de eficiência em comparação com outra técnica aplicada. Tanto Zanette, Wainwright e Brunskill (2021) quanto Levine et al. (2020) destacam as vantagens técnicas associadas ao uso desse algoritmo em comparação com outras abordagens.

Considerando os avanços alcançados por abordagens de Inteligência Artificial (IA), este estudo explora as potencialidades de aplicar uma IA treinada com algoritmo ator-crítico em um veículo operacional. O treinamento desse modelo envolve a utilização de dados provenientes de uma Unidade de Controle Eletrônico (ECU) e do Sistema de Posicionamento Global (GPS). Com o objetivo de aprimorar a interação entre o motorista e o veículo, especialmente no que diz respeito à eficiência do motor e à sua relação com a velocidade e aceleração do automóvel, o sistema deve fornecer sugestões de aceleração de maneira intuitiva para o motorista.

O objetivo central é formular um algoritmo ator-crítico que avalie o consumo energético em veículos à combustão com base nos dados de um trajeto predefinido, coletados de uma ECU e do GPS de um veículo em funcionamento. A proposta é que o sistema, a partir dessa análise, forneça recomendações específicas relacionadas à aceleração do veículo, através de sinais simples, visando otimizar a eficiência energética. Este algoritmo tem o potencial de ser integrado posteriormente em um ADAS.

A abordagem metodológica adotada segue uma trajetória semelhante àquela empregada por Liu et al. (2020). Utilizar-se-á um algoritmo com capacidade de antecipar o consumo energético, contudo, com atuação passiva, concentrando-se na ampliação da compreensão do motorista em vez de agir diretamente sobre o veículo. Para alcançar esse propósito, serão utilizadas bases de dados provenientes de veículos que percorreram uma variedade de trajetos distintos. Essa abordagem será instrumental no desenvolvimento, testes e na obtenção dos resultados desejados para a pesquisa.

1.1 OBJETIVOS

Para abordar a questão da falta de compreensão por parte dos condutores em relação às ações que podem ser adotadas para aprimorar a eficiência energética na condução de veículos, este trabalho propõe os seguintes objetivos.

1.1.1 Objetivo Geral

O objetivo principal é desenvolver um ADAS fundamentado em aprendizado por reforço com algoritmo ator-crítico. Esse sistema terá a capacidade de fornecer

sugestões de ações relacionadas à velocidade do veículo, visando aumentar a eficiência energética.

1.1.2 Objetivos Específicos

- Descrever um sistema embarcado capaz de receber informações sobre o consumo energético de um veículo;
- Indicar ações a serem tomadas pelo motorista para melhorar a eficiência energética durante um percurso pré-definido, levando em conta o perfil de altitude;
- Avaliar a redução de consumo de combustível em simulações.

2 FUNDAMENTAÇÃO TEÓRICA

Para avaliar a viabilidade da implementação de um algoritmo ator-crítico visando a redução do consumo energético, utilizando dados de uma ECU já existente e de GPS, é necessário um entendimento dos conceitos, aspectos físicos e técnicas disponíveis para a correta modelagem e análise dos dados.

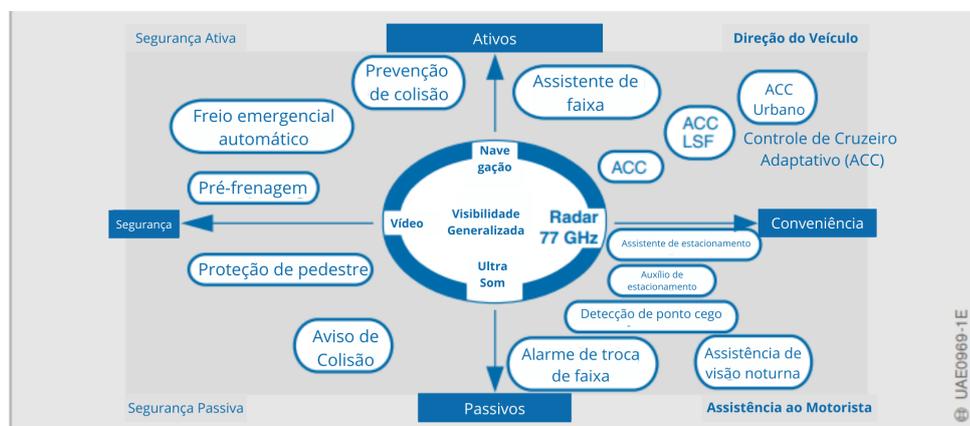
Por isso, este capítulo visa abordar teorias e técnicas utilizadas na pesquisa apresentada neste trabalho. Faz-se necessário compreender o ADAS, esclarecer a abordagem do consumo energético do veículo, assim como, dissertar sobre os conceitos de aprendizado por reforço e fundamentos dos algoritmos aplicados.

2.1 SISTEMA AVANÇADO DE ASSISTÊNCIA AO MOTORISTA (ADAS)

Os ADAS são sistemas que auxiliam o condutor de maneira passiva ou ativa, buscando estabelecer uma compreensão mútua entre o motorista e o veículo. O principal objetivo desses sistemas é mitigar ou eliminar riscos, proporcionando uma experiência de condução mais segura e confortável (LI et al., 2011).

Os ADAS podem ser categorizados em sistemas de segurança, voltados para a prevenção de acidentes, e sistemas de conforto e conveniência, destinados a facilitar a direção semiautomática. Essa classificação pode ser refinada ainda mais, distinguindo entre sistemas ativos e passivos, com base no nível de intervenção do motorista no sistema (REIF, 2014). A Figura 1 ilustra as diferentes categorias dos ADAS em quatro quadrantes.

Figura 1 – Funções de Segurança e Conveniência Baseadas nos Sensores do Veículo



Fonte: Traduzido de Reif (2014, p. 181.)

No quadrante inferior direito da Figura 1, encontram-se os sistemas passivos

de conveniência. Esses sistemas oferecem recomendações de manobras que podem ser executadas para aumentar o conforto do motorista, como é o caso do assistente de estacionamento. Eles indicam a dificuldade de estacionar em um determinado local ou projetam a trajetória na câmera de ré, facilitando a manobra.

No quadrante inferior esquerdo da Figura 1, estão os sistemas passivos de segurança. Esses sistemas apresentam indicadores que alertam o motorista sobre potenciais perigos, visando aumentar a segurança, como o aviso de colisão. Esse aviso, por meio de sinais visuais ou sonoros, informa sobre a proximidade do veículo em relação a um obstáculo iminente.

Os sistemas ativos de segurança, situados no quadrante superior esquerdo da Figura 1, atuam diretamente no veículo, tomando ações no lugar do motorista para aumentar a segurança. Um exemplo é o freio emergencial automático, que aciona os freios automaticamente ao detectar uma situação de colisão, evitando ou reduzindo os danos causados às pessoas e veículos envolvidos.

Por fim, os sistemas ativos de conveniência estão representados no quadrante superior direito da Figura 1. Esses sistemas também tomam ações no lugar do motorista, visando aumentar o conforto, como os Controles de Cruzeiro Adaptativos (ACC). Os ACCs controlam diretamente a velocidade do motor de acordo com o veículo à frente e a velocidade determinada pelo motorista.

Os sistemas ADAS podem ser classificados em cinco níveis de automação, conforme definido por Winner et al. (2014):

- Nível 1: Ausência de automação, em que o condutor é totalmente responsável pelo controle do veículo.
- Nível 2: Automação parcial do controle longitudinal ou lateral do veículo, com o condutor mantendo responsabilidade por parte do controle.
- Nível 3: Automação de grau intermediário, em que o sistema assume o controle tanto longitudinal quanto lateral do veículo, mas o condutor deve permanecer atento.
- Nível 4: Elevado grau de automação, permitindo que o condutor não precise ficar atento ao tráfego ou à direção do veículo, mas ainda pode escolher retomar o controle.
- Nível 5: Ausência total de controle manual, com o veículo sendo conduzido autonomamente, dispensando a necessidade de um condutor.

Os sistemas ADAS também podem ser categorizados de acordo com os modos de operação. Os modos de operação do ADAS incluem o Modo A, no qual o sistema fornece informações e alertas ao condutor sem interferir diretamente no veículo, como indicadores para a redução do consumo energético. No Modo B, o ADAS assume imediatamente o controle parcial do veículo, mas o condutor mantém o controle total,

como no caso do controle de velocidade de cruzeiro. Já no Modo C, o ADAS assume o controle total do veículo em situações de emergência, como frenagem de emergência e sistemas de anti-colisão (WINNER et al., 2014).

Para atingir o objetivo proposto, os ADAS utilizam ECUs para gerenciar sensores e atuadores. Cada ECU no veículo é responsável por uma função ou conjunto específico de funções. A ECU dedicada ao motor, por exemplo, comunica-se com diversas outras partes do veículo. Assim, é possível desenvolver ADAS que utilizem a ECU do motor para obter informações e interagir com outras ECUs do veículo (FAIZAN; PILLAI, 2019).

No escopo deste trabalho, foi decidido utilizar ECUs já existentes em um veículo para coletar dados e fornecer a um sistema ADAS de Nível 1, operando no Modo A. Esse sistema é capaz de indicar ações para reduzir o consumo energético do veículo.

2.2 CONSUMO ENERGÉTICO EM VEÍCULOS

O consumo energético em veículos é um elemento crucial para avaliar sua eficiência e impacto ambiental. Atualmente, existem duas tecnologias predominantes em veículos: motores à combustão e motores elétricos. Além disso, é possível encontrar a combinação de ambos em veículos conhecidos como carros híbridos (HEYWOOD, 2018).

2.2.1 Motores a Combustão

Motores a combustão, como os motores a gasolina e diesel, realizam a conversão de energia química proveniente do combustível em energia mecânica. Esse processo químico ocorre pela queima ou oxidação do combustível dentro do motor (HEYWOOD, 2018).

A eficiência do consumo de energia é expressa em termos de taxa de consumo ou eficiência energética, geralmente medida em litros por quilômetro ou quilômetros por litro, respectivamente. O consumo específico (*sfc* - specific fuel consumption) é uma medida da eficiência do motor e é calculado pela Equação 1.

$$sfc = \frac{\dot{m}_f}{P} \quad (1)$$

Onde \dot{m}_f é o fluxo de massa pelo tempo (g/s) e P é a potência (kW) (HEYWOOD, 2018).

2.2.2 Motores Elétricos

Os VEs são alimentados por baterias ou outros dispositivos de armazenamento de energia. A relevância dos motores elétricos cresce não apenas devido à redução

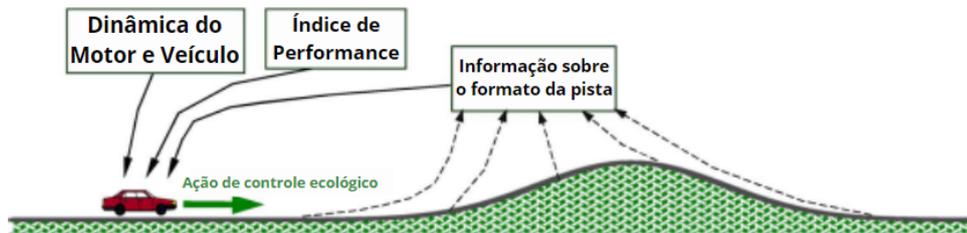
da poluição ambiental e sonora, mas também por diminuir a dependência do petróleo (LARMINIE; LOWRY, 2012).

O consumo de energia elétrica é geralmente expresso em termos de taxa de consumo ou eficiência, medida em quilowatt-hora por quilômetro ou quilômetro por quilowatt-hora, respectivamente. Essas métricas podem ser determinadas por meio de testes (LARMINIE; LOWRY, 2012).

2.2.3 Consumo Energético em uma Pista com Alteração de Nível

O desenvolvimento do ADAS requer um modelo de consumo ou eficiência energética do veículo, que é influenciado por diversos fatores, tais como a velocidade do motor, torque, relação de transmissão, temperatura, valor calórico do combustível, entre outros (KAMAL et al., 2011). Criar um modelo preciso do consumo energético é uma tarefa complexa e não é o foco específico deste trabalho. No entanto, é possível realizar uma aproximação utilizando uma função diferenciável da velocidade, considerando o sistema proposto na Figura 2, o que é suficiente para desenvolver o algoritmo proposto para o ADAS (KAMAL et al., 2011).

Figura 2 – Proposta de sistema de *eco-driving* usando previsão baseada em modelo e informações do formato da pista



Fonte: Traduzido de Kamal et al. (2011)

A Figura 2 representa um sistema no qual a ação para o controle ecológico é determinada com base em informações sobre o formato da pista, índice de desempenho e dinâmica do motor e do veículo. Para veículos convencionais, o consumo nessa situação pode ser estimado pela Equação 2.

$$f_V = (b_0 + b_1x_2 + b_2x_2^2 + b_3x_2^3 + \hat{a}(c_0 + c_1x_2 + c_2x_2^2)) \quad (2)$$

Em que, \hat{a} é a soma da aceleração aparente a_V e a aceleração necessária para superar a força desaceleradora do desnível $a_\theta = g \sin \theta(x_1)$. E os parâmetros de b_0 à b_3 e c_0 à c_2 são coeficientes constantes específicos do motor do veículo, obtidos através de valores coletados de velocidade e aceleração, a obtenção destes pôde ser encontrada no trabalho desenvolvido por Kamal et al. (2011).

2.3 APRENDIZADO DE MÁQUINA

A técnica de Aprendizado de Máquina (AM) é um subcampo da Inteligência Artificial (IA) utilizado para realizar tarefas, substituindo a escrita manual de código pela derivação de um programa a partir de dados coletados sobre o problema. O programa aprende com experiência E sobre um conjunto de tarefas T e obtém desempenho P se o seu desempenho P em tarefas do conjunto T melhora com a experiência E (MITCHELL et al., 2007).

Os sistemas de AM podem ser classificados em quatro categorias dependendo do nível de supervisão. Essas categorias são (GÉRON, 2022):

- Supervisionado: Os dados de treinamento incluem a solução desejada pelo sistema, chamados de *labels*.
- Não-Supervisionado: Os dados de treinamento não incluem a solução, e o sistema não possui *labels*.
- Semi-Supervisionado: Uma pequena parte dos dados possui *labels*, enquanto a maior parte não possui.
- Por Reforço: O sistema, nesse contexto chamado de agente, observa o ambiente, executa ações e recebe recompensas ou penalidades.

2.3.1 Aprendizado Supervisionado

O aprendizado supervisionado é um dos pilares fundamentais do AM, amplamente aplicado em diversos campos. Nesse paradigma, algoritmos são treinados utilizando um conjunto de dados composto por pares de entrada e saída, possibilitando ao modelo aprender a mapear entradas para saídas desejadas. Uma das características distintivas do aprendizado supervisionado é a habilidade do modelo de generalizar para novos dados com base nos padrões extraídos do conjunto de treinamento (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Uma variedade de contextos permite a aplicação de um aprendizado supervisionado, desde reconhecimento de voz e visão computacional até previsão financeira e diagnóstico médico. Essa técnica desempenha um papel crucial no desenvolvimento de sistemas de inteligência artificial capazes de tomar decisões, classificar dados e enfrentar tarefas complexas. Diferentes algoritmos supervisionados, como árvores de decisão, Máquinas de Vetores de Suporte (SVM), redes neurais e regressão linear, estão disponíveis, cada um com suas características e aplicações específicas (BISHOP; NASRABADI, 2006).

2.3.2 Aprendizado Não-Supervisionado

O aprendizado não supervisionado, um dos alicerces da aprendizagem de máquina, destaca-se por sua capacidade de explorar dados sem depender de rótulos

ou saídas predefinidas. Em contraste com o aprendizado supervisionado, no qual os modelos se baseiam em exemplos rotulados, o aprendizado não supervisionado visa identificar estruturas e padrões subjacentes em conjuntos de dados. Essa abordagem revela-se particularmente valiosa quando se lida com dados não rotulados e na tarefa de redução de dimensionalidade (BISHOP; NASRABADI, 2006).

O aprendizado não-supervisionado encontra aplicações diversas, desde a segmentação de clientes com base em padrões de compra até a análise de tópicos em grandes conjuntos de documentos de texto e a redução da dimensionalidade para fins de visualização. Algoritmos não supervisionados, como K-Means, Análise de Componentes Principais (PCA) e Autoencoders, desempenham um papel fundamental na extração de informações valiosas de dados não rotulados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

O aprendizado não supervisionado integra-se como um componente essencial na caixa de ferramentas da aprendizagem de máquina, permitindo a descoberta de padrões e informações valiosas em dados que carecem de rótulos. Essa técnica é particularmente relevante em contextos nos quais a disponibilidade de rótulos é limitada ou impraticável (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.3.3 Aprendizado Semi-Supervisionado

O aprendizado semi-supervisionado representa uma abordagem altamente promissora no âmbito da aprendizagem de máquina, capitalizando em dados que incluem tanto amostras rotuladas quanto não rotuladas. Ao contrário do aprendizado supervisionado tradicional, que se baseia exclusivamente em dados rotulados, o aprendizado semi-supervisionado permite que os modelos utilizem informações adicionais não rotuladas, tornando-o uma estratégia eficaz em situações em que os rótulos são escassos ou dispendiosos de se obter (ZHU; GOLDBERG, 2022).

O aprendizado semi-supervisionado encontra aplicações diversas, abrangendo desde a classificação de documentos e o reconhecimento de fala até o processamento de imagens. Essa técnica ganha destaque quando se lida com grandes volumes de dados nos quais a rotulagem de todos os exemplos é inviável. Algoritmos de aprendizado semi-supervisionado, como a propagação de rótulos e a mistura de modelos generativos, desempenham um papel fundamental na exploração do potencial não explorado dos dados não rotulados (CHAPELLE; SCHOLKOPF; ZIEN, 2006).

O aprendizado semi-supervisionado constrói uma ponte valiosa entre o aprendizado supervisionado convencional e o não supervisionado, permitindo que os algoritmos aprendam com dados rotulados e não rotulados, otimizando a utilização das informações disponíveis, especialmente em cenários do mundo real (ZHU; GOLDBERG, 2022).

2.3.4 Aprendizado por Reforço

O Aprendizado por Reforço é uma técnica de AM que se baseia na noção de recompensas. Um agente interage com um ambiente, recebe informações e as processa para gerar uma resposta possível. Essa é expressa por meio de uma ação. Após essa etapa, o agente avalia o impacto da ação no ambiente, analisando a função de recompensa e ajustando os pesos dos fatores de probabilidade, promovendo um progresso positivo durante o treinamento (SUTTON; BARTO, 2018).

Nos Algoritmos de Aprendizado por Reforço, os agentes se ajustam ao ambiente com base em recompensas, semelhante ao comportamento fisiológico dos animais. Dessa forma, em um determinado momento t , o agente avalia sua situação anterior à execução da ação a , bem como sua situação no ambiente, e busca a solução ótima ou semi-ótima (POWELL, 2022).

Os algoritmos clássicos de Aprendizado por Reforço, como Q-Learning e SARSA, utilizam tabelas de probabilidade como base (SUTTON; BARTO, 2018). Essas tabelas são criadas com os possíveis estados do ambiente e as ações correspondentes. Os valores na tabela representam as probabilidades, ponderadas de ocorrência de uma determinada ação, conforme definido pelas equações relevantes (SUTTON; BARTO, 2018).

Durante o treinamento, ao realizar um passo para um estado atual s_t , a recompensa r_t resultante da ação a_t é analisada. Os valores do estado e ação são atualizados, considerando uma taxa de aprendizado α e uma taxa de desconto γ , a qual determina a importância das ações futuras no próximo estado $s_t + 1$. No Q-Learning, a melhor ação futura possível é sempre selecionada, no SARSA, a ação é escolhida de acordo com uma política específica (POWELL, 2022).

Nota-se que um algoritmo de aprendizado por reforço seria ideal para este trabalho. Uma vez que o motorista pode realizar ações a em um dado momento t e receber uma recompensa baseada no impacto que esta ação terá no consumo do veículo considerando o trecho a frente. Ademais, uma abordagem supervisionada pode ser inviável, visto que necessitaria a coleta de dados de diferentes rodovias e ações tomadas por motoristas em cada trecho das mesmas, inviabilizando um uso geral.

Existem diversos algoritmos capazes de realizar o aprendizado por reforço, no Capítulo 3 será discutido as formas de implementação. Para este trabalho será utilizado o Algoritmo *Asynchronous Advantage Actor-Critic* (A3C), por isso faz-se necessário o entendimento deste.

2.3.4.1 Algoritmos Ator-crítico

O ator-crítico é uma variação de aprendizado por reforço que possui dois principais componentes, o ator representa a política, e aprende a selecionar ações

baseadas em observações, e o crítico avalia a qualidade ou valor da ação escolhida. Eles combinam forças de métodos baseados em modelo e métodos baseados em políticas (POWELL, 2022).

Para realizar a combinação e extrair o melhor de ambos, o ator aplica um método de gradiente de política, otimizando uma medida de desempenho. Um algoritmo de gradiente de política amplamente utilizado é o algoritmo REINFORCE, introduzido por Ronald J. Williams em 1992. Ele utiliza o teorema do gradiente de política para atualizar os parâmetros do ator na direção de recompensas esperadas maiores (SUTTON; BARTO, 2018).

Com o mesmo objetivo, o componente crítico estima a função de valor, que mede as recompensas esperadas ao seguir uma determinada política. A estimativa da função é comumente feita utilizando métodos de diferença temporal, como TD(0) ou TD(λ). Os algoritmos ator-crítico, frequentemente, utilizam uma função de vantagem para melhorar o processo de aprendizado. A função de vantagem quantifica a vantagem de tomar uma ação específica, comparada com o valor médio de todas as ações em um determinado estado. Isso ajuda o ator a focar nas ações que levam a recompensas mais altas (WIERING; OTTERLO, 2012).

2.3.4.2 Algoritmo Asynchronous Advantage Actor-Critic (A3C)

O algoritmo A3C representa uma inovação de vanguarda no campo do Aprendizado por Reforço. Introduzido por Mnih et al. (2016), este algoritmo destaca-se por sua habilidade de enfrentar desafios em ambientes complexos e de alta dimensionalidade. O A3C é uma evolução em relação a outros algoritmos de aprendizado por reforço e combina aprendizado supervisionado com políticas estocásticas.

O A3C é uma síntese de duas abordagens cruciais: o método Ator-Crítico e a execução assíncrona por múltiplos agentes. O ator busca otimizar a política para maximizar recompensas, enquanto o crítico estima o valor de uma política. A execução assíncrona permite que múltiplos agentes treinem simultaneamente, acelerando consideravelmente o processo de aprendizado.

Este algoritmo demonstrou sucesso em diversas aplicações, desde jogos Atari até controle de robôs e treinamento de veículos autônomos. Sua capacidade de lidar com ambientes desafiadores e de alta complexidade o consagra como uma ferramenta de destaque no universo do Aprendizado por Reforço Profundo (MNIH et al., 2016).

2.3.4.3 Ambiente de treinamento

No contexto do Aprendizado por Reforço, o ambiente de treinamento é o elemento fundamental no qual algoritmos se desenvolvem e aprimoram suas

habilidades. Compreender profundamente as características que moldam o processo de aprendizado é essencial para projetar um ambiente eficaz. Isso envolve elementos críticos, como observações, ações, recompensas e as dinâmicas do ambiente.

As observações, ou estados, são as informações que o agente percebe a partir do ambiente, frequentemente representadas como vetores de alta dimensão. As ações, por sua vez, definem as escolhas disponíveis ao agente, variando de opções discretas a ações contínuas, dependendo do contexto. As recompensas desempenham um papel central, fornecendo feedback sobre o sucesso ou fracasso das ações do agente em relação aos objetivos estabelecidos. Além disso, as dinâmicas do ambiente descrevem como o estado do ambiente evolui em resposta às ações do agente, estabelecendo a conexão entre as ações tomadas e as mudanças nas observações (KAELBLING; LITTMAN; MOORE, 1996).

Projetar um ambiente de treinamento bem definido é uma etapa crucial na pesquisa de aprendizado por reforço, pois afeta diretamente a capacidade do agente de aprender comportamentos ótimos. A escolha criteriosa de observações, ações, recompensas e a modelagem das dinâmicas do ambiente exige uma compreensão profunda do problema em questão. Assim, um ambiente habilmente concebido constitui a base do sucesso para qualquer algoritmo de Aprendizado por Reforço (SUTTON; BARTO, 2018).

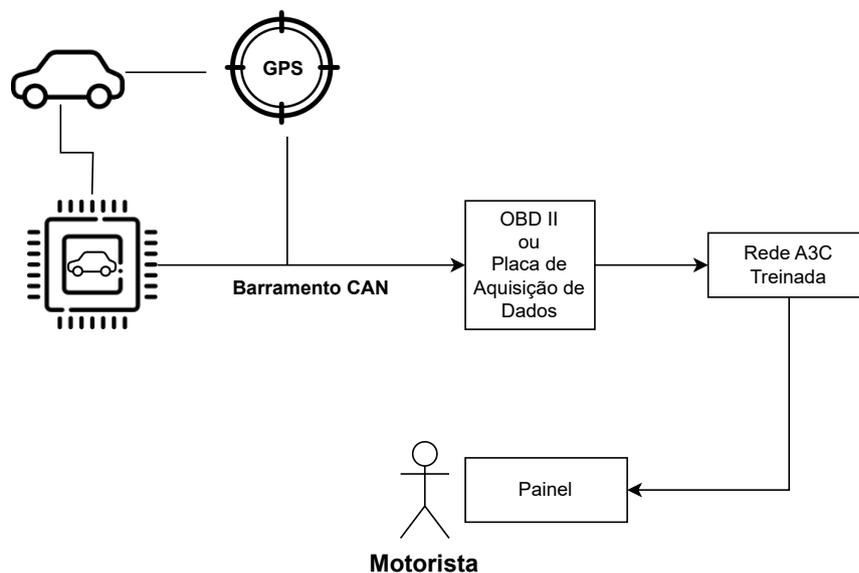
3 MATERIAIS E MÉTODOS

O estudo do consumo energético em veículos tem um impacto direto em todos os setores da sociedade, incluindo consumidores, fabricantes e governos. Para compreender a aplicabilidade deste trabalho, é necessário conhecer os modelos que são utilizados para orientar os estudos nesse tema. Neste capítulo, serão discutidos os métodos empregados com o objetivo de alcançar os objetivos propostos.

3.1 PROJETO DO SISTEMA

Espera-se que o modelo desenvolvido durante este trabalho de conclusão de curso seja aplicado em um sistema ADAS completo. Assim, apesar do trabalho conduzir seu foco para modelagem e avaliação do sistema de IA, um esboço do sistema que envolve desde a coleta até a recomendação ao motorista é apresentado. A Figura 3 ilustra o sistema proposto.

Figura 3 – Projeto do Sistema



Fonte: Produzido pelo Autor

O sistema proposto utiliza dados da ECU do motor e do GPS do veículo, os quais são coletados por meio da porta OBD II ou de uma placa desenvolvida para interpretar dados do barramento Controller Area Network (CAN). A própria placa do sistema embarcado armazena a rede A3C previamente treinada. Após receber os dados pré-processados, a rede é capaz de fornecer uma recomendação de ação visando a eficiência energética. A ação indicada para o motorista é exibida no painel em formato de seta para que seja intuitivo interpretar.

Em casos em que os veículos não possuem módulos GPS embutidos, é possível utilizar os dados de GPS de um aparelho celular conectado ao sistema embarcado. Se isso não for viável, ainda é possível gerar um plano de viagem para que o motorista o siga, embora sem recomendações em tempo real.

Este trabalho aborda apenas a etapa de treinamento da rede A3C e a recomendação das ações para o motorista. Os dados utilizados, que possibilitaram o desenvolvimento deste sistema, são apresentados na Seção 3.3.

Por se tratar de um sistema de recomendação de ações que não atua diretamente no veículo, e o foco ser a eficiência do veículo, que não está diretamente relacionada à segurança, o ADAS proposto se enquadra como um sistema Passivo de Conveniência, de Nível 1. O ADAS atua no Modo A, fornecendo apenas informações sobre ações para o motorista. Contudo, trabalhos futuros podem evoluir esse sistema para contemplar níveis e modos de condução mais avançados.

3.2 MODELOS DE ESTIMATIVA DE CONSUMO

Para o desenvolvimento deste trabalho, é essencial compreender os modelos existentes para a estimativa de consumo de um veículo. Os Métodos Analíticos compreendem modelos matemáticos capazes de medir o impacto das variáveis do ambiente no consumo e, a partir do cálculo, definir as melhores ações a serem tomadas. Por outro lado, os Métodos de Aprendizado por Reforço não exigem um conhecimento prévio do impacto de todas as variáveis.

3.2.1 Métodos Analíticos

Os métodos apresentados aqui são aqueles que demandam um conhecimento aprofundado sobre a física do consumo energético em um veículo. Os modelos de consumo podem ser categorizados em Caixa-Branca, Caixa-Preta e Caixa-Cinza (ZHOU; JIN; WANG, 2016):

- Modelos Caixa-Branca:
 - Método de Balanço de Carbono.
 - Modelo de Valor Médio Fenomenológico.
- Modelos Caixa-Preta:
 - Modelos Caixa-Preta de Consumo de Combustível Baseado no Motor.
 - Modelos Caixa-Preta de Consumo de Combustível Baseado no Veículo.
 - Modelos Caixa-Preta de Consumo de Combustível Baseado no Modal.
- Modelos Caixa-Cinza:
 - Softwares de Simulação de Emissão Transiente do Veículo (VeTESS).
 - Modelo de Consumo de Combustível Transiente para Veículos Não-Rodoviários.

Os modelos de Caixa-Branca são caracterizados por sua alta precisão, mas demandam uma complexidade significativa. É necessário instalar sensores caros dentro dos motores, além de possuir um conhecimento aprofundado dos processos de ingestão, compressão, combustão e exaustão (ZHOU; JIN; WANG, 2016).

Em contraste, os modelos Caixa-Preta são mais simples, porém apresentam uma precisão reduzida. Geralmente, são utilizadas variáveis dependendo da base do modelo. Nos modelos baseados no motor, empregam-se variáveis como torque, velocidade e potência do motor. Já nos modelos baseados no veículo, utilizam-se as variáveis de velocidade e aceleração instantâneas. Os modelos baseados nos modais consideram o modo básico de operação, como em cruzeiro, marcha lenta, acelerando e desacelerando (ZHOU; JIN; WANG, 2016).

Para equilibrar simplicidade e precisão, os modelos de Caixa-Cinza não demandam tantos dados adicionais em relação aos modelos Caixa-Preta para prever de forma mais precisa, e apresentam uma estrutura menos complexa do que os modelos Caixa-Branca. Esses modelos são frequentemente utilizados para realizar roteamento e direção ecológica, visando a redução da emissão de gases poluentes (ZHOU; JIN; WANG, 2016).

O objetivo deste trabalho é desenvolver um ADAS de alta precisão. Embora o modelo de Caixa-Branca seja considerado ideal devido à sua alta precisão, a implementação desse modelo demandaria sensores caros e geralmente não disponíveis nas ECUs dos veículos. No entanto, as informações já presentes no barramento CAN possibilitam a implementação de um modelo Caixa-Preta. Para alcançar uma precisão próxima ao modelo Caixa-Branca com os dados disponíveis para um modelo Caixa-Preta, uma abordagem como AM ou Aprendizado por Reforço se faz necessária. Essa abordagem permite que o sistema aprenda com os dados disponíveis, ajustando-se dinamicamente para melhorar sua precisão ao longo do tempo.

3.2.2 Método de Aprendizado por Reforço

Os métodos de aprendizado por reforço oferecem a vantagem de não exigir um conhecimento aprofundado dos processos internos do motor ou do modelo de consumo para alcançar uma alta precisão na previsão.

Para este trabalho, escolheu-se o aprendizado por reforço, pois se destaca em tarefas que envolvem tomadas de decisão sequenciais e interações com o ambiente sem influência de um viés predefinido. Ao contrário do aprendizado supervisionado, que depende de grandes conjuntos de dados rotulados, o aprendizado por reforço opera com feedback mínimo. Além disso, ele pode explorar ativamente o ambiente, adaptando-se a mudanças e generalizando políticas de decisão gerais, tornando-se mais robusto que métodos semi-supervisionados e não supervisionados.

Dessa forma, devido à falta de um conhecimento aprofundado dos processos

internos do veículo e à incerteza sobre quais ações devem ser tomadas em determinadas situações, é necessário um algoritmo capaz de explorar o ambiente e determinar, a partir de feedbacks limitados, a melhor ação para reduzir o consumo.

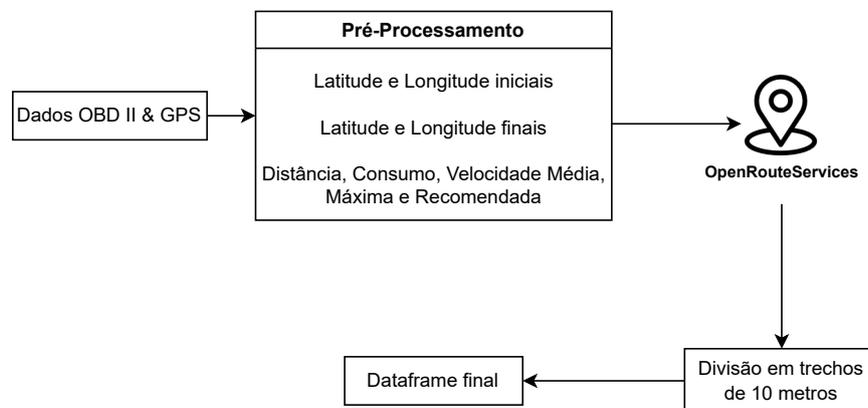
Este trabalho adota conceitos semelhantes aos apresentados por Liu et al. (2020), mas aborda um problema diferente. O objetivo é desenvolver um sistema que auxilie o condutor do veículo com sugestões relacionadas à aceleração, levando em consideração o relevo da pista à frente.

Além disso, o algoritmo escolhido é um ator-crítico. Apesar da semelhança com a rede dupla de Q-learning implementada no trabalho citado, o algoritmo ator-crítico apresentou uma redução de 87,5% no tempo de treinamento, e uma melhora de 23,3% na economia de combustível no trabalho de Wu et al. (2021).

3.3 BASE DE DADOS E TRATAMENTO

A fonte de dados e o pré-processamento dos dados são partes cruciais para o bom resultado do treinamento de uma Inteligência Artificial. A Figura 4 demonstra o processo de coleta e processamento dos dados utilizados para o treinamento da rede A3C desenvolvida.

Figura 4 – Coleta e Tratamento dos Dados



Fonte: Produzido pelo Autor

Como visto na Figura 4, os dados da tabela inicial foram utilizados apenas para coletar as coordenadas de início e fim do trajeto, além das estatísticas dos mesmos. A partir desses dados e da escolha do trajeto, uma Interface de Programação de Aplicações (API) foi utilizada e os dados foram segmentados para chegar ao dataframe final, que foi utilizado para o treinamento.

Visto que o objetivo do trabalho é o desenvolvimento do algoritmo, os dados reais utilizados foram coletados por Igor Silva (2022) com a utilização da porta OBD de automóveis e pelos dados do GPS durante o trajeto. Esses dados foram armazenados

em uma tabela com todas as informações coletadas por ambos (IGOR SILVA, 2022). A tabela foi construída a partir de 355 viagens diferentes, totalizando um tempo de 46 horas e 42 minutos de dados coletados. A tabela, além de apresentar os dados do percurso, inclui uma identificação do motorista e do veículo dos quais os dados foram coletados. Uma explicação de cada campo da tabela pode ser encontrada no Anexo A.

Para este trabalho, uma vez que o ambiente deve simular o real, os dados foram tratados de forma a permitir a flexibilização da tabela. Dessa forma, ao agente tomar uma decisão, é possível calcular a distância percorrida e simular o movimento dele na pista, reduzindo o dataframe inicial.

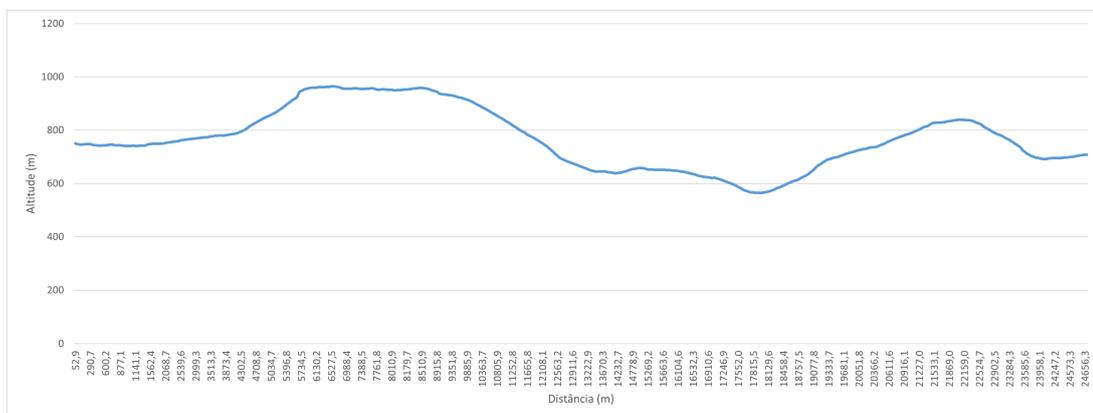
Inicialmente, é necessário definir os dados que serão enviados nas observações do agente no ambiente. Para solucionar o problema proposto, definiu-se que são necessários os dados de distância percorrida até o momento, distância faltante para o fim do trajeto, altitude atual, velocidade máxima, velocidade indicada, velocidade atual e diferença de altitude dos próximos 100 metros.

A definição da diferença de altitude dos próximos 100 metros se deu devido a uma limitação da API utilizada. A mesma retorna trechos não uniformes e, em geral, a altitude varia a cada 100 metros; a segmentação dentro dessa distância mantém a mesma altitude retornada para a primeira coordenada.

Portanto, os dados coletados foram agrupados por trajeto realizado, trazendo as coordenadas de início e fim do trajeto a partir das colunas Longitude e Latitude da fonte, a velocidade média do veículo (coluna Speed(OBD)(KM/h)), e a média da eficiência em km/L (coluna TripDistance(KM) dividida pela coluna FuelUsed(Trip)(l)). Além disso, optou-se por manter apenas trajetos mais longos.

O trajeto escolhido para o treinamento e coleta de resultados neste trabalho foi o trajeto de ID 287, pois apresenta um perfil de altitude mais adequado para o objetivo deste trabalho, com duas grandes subidas e três grandes descidas ao longo do trajeto, conforme ilustrado na Figura 5.

Figura 5 – Perfil de Altitude - Trajeto 287



Fonte: Produzido pelo Autor

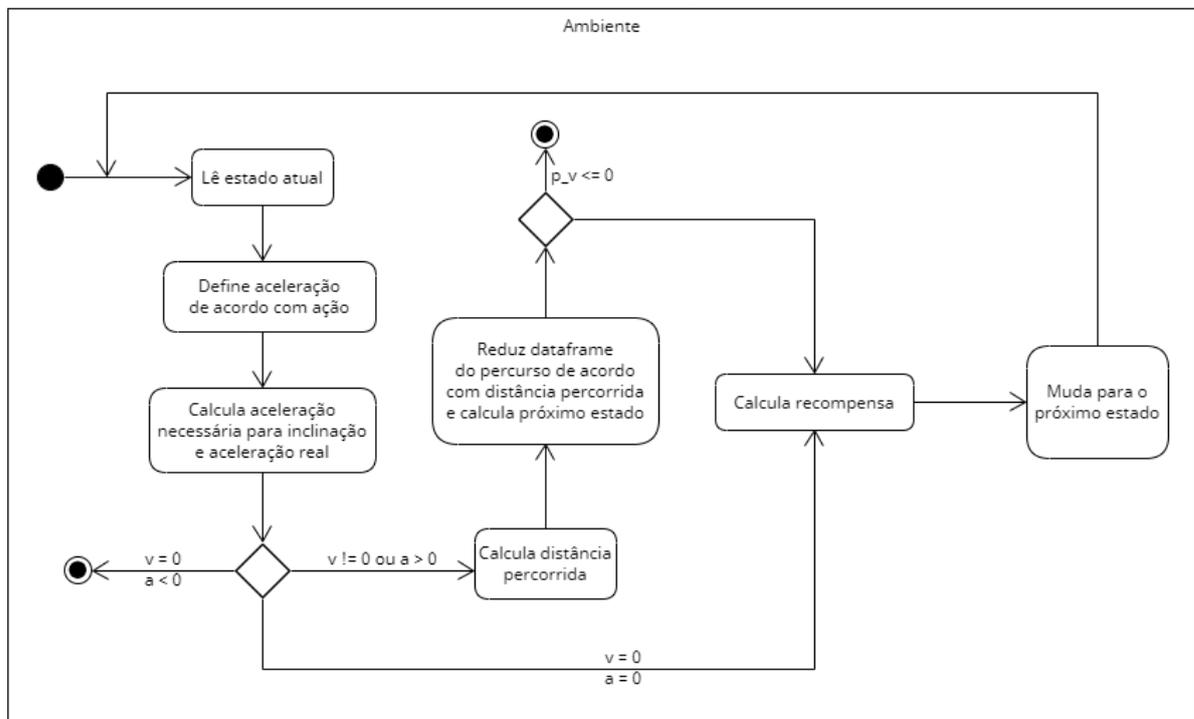
Conforme ilustrado na Figura 5, o trajeto possui uma distância total de 24,66 quilômetros, com variação de altitude entre 565 e 965 metros. O percurso começa com uma leve subida que se torna mais íngreme. Após um platô, a pista desce novamente, atingindo sua altitude mínima. Próximo aos 18 quilômetros, inicia-se uma nova subida, e o trajeto é concluído com uma descida.

Utilizando as coordenadas do trajeto, foi empregado o serviço de API da *openrouteservices* (HEIGIT GGMBH, 2022) para obter os dados de velocidade indicada e velocidade máxima. O resultado foi um dataframe com o trajeto segmentado a cada 10 metros, a fim de flexibilizar o cálculo do movimento de acordo com a lógica apresentada na Seção 3.4.

3.4 HEURÍSTICA DO AMBIENTE

O ambiente do agente deve representar o ambiente real da forma mais fidedigna possível. Para resolver o problema proposto por este trabalho, o ambiente deve simular um veículo percorrendo um trajeto pré-definido e receber observações de acordo com o percurso e a variação de altitude. A Figura 6 apresenta o fluxograma geral do comportamento e resposta do ambiente desenvolvido para o treinamento dos agentes.

Figura 6 – Fluxograma do Ambiente



Fonte: Produzido pelo Autor

Como mostrado na Figura 6, o ambiente inicia com o DataFrame do trajeto coletado e processado, conforme descrito na Seção 3.3. O primeiro passo consiste em

ler os parâmetros que definem o estado do agente, conforme apresentado na Seção 3.3; inicialmente, esses parâmetros representam o início do trajeto. O ambiente recebe como entrada a ação escolhida pelo agente, a partir da qual uma aceleração é definida. Com a informação da inclinação da pista, é possível calcular a aceleração necessária para superar essa inclinação, resultando na aceleração real do veículo.

Após os cálculos iniciais, se a ação tomada faz o veículo andar de ré, o ambiente é encerrado. Caso o veículo permaneça parado, a recompensa é calculada. Se o veículo estiver em movimento, é calculada a distância que ele percorre em 3 segundos, e o DataFrame é reduzido de acordo com essa distância.

Para definir a função recompensa, é necessário levar em conta o objetivo do trabalho. Assim, a recompensa proposta para o agente é diretamente proporcional à eficiência energética do agente em percorrer o trajeto por completo.

Porém, existem estados indesejáveis no ambiente real, como, por exemplo, o veículo parar no meio da subida ou andar de ré durante o trajeto. Esses estados também devem ser tratados para que o agente seja devidamente penalizado por tomar ações que afetariam negativamente o motorista no percurso.

A função de recompensas final é definida em partes e depende da velocidade, da aceleração escolhida pelo algoritmo, da distância percorrida momentânea e total. Para definir a recompensa final, são necessários alguns cálculos iniciais.

A distância percorrida pelo agente e sua próxima velocidade em um passo são calculadas através da função horária de posição e velocidade do movimento uniformemente variado, mostradas pelas Equações 3 e 4, respectivamente.

$$S = S_0 + V_0t + \frac{a_r t^2}{2} \quad (3)$$

$$V = V_0 + a_r t \quad (4)$$

A aceleração real a_r é definida pela aceleração escolhida pelo agente e a aceleração necessária para superar a inclinação dos próximos 100 metros. A Equação 5 demonstra a função por partes para a definição da aceleração em função da ação escolhida pelo agente.

$$a(action) = \begin{cases} -3m/s^2, & \text{se } action = 0 \\ -1m/s^2, & \text{se } action = 1 \\ 0m/s^2, & \text{se } action = 2 \\ 1m/s^2, & \text{se } action = 3 \\ 3m/s^2, & \text{se } action = 4 \end{cases} \quad (5)$$

O agente toma uma decisão baseada nas probabilidades, como visto no Capítulo 3. O espaço das ações é definido entre 0 e n-1, sendo n o número de ações possíveis para o agente tomar. Ao escolher a ação 0, o agente desacelera a

3 m/s^2 , com a ação 1, ele desacelera a 1 m/s^2 . A ação 2 mantém a velocidade do agente, enquanto as ações 3 e 4 aceleram o agente em 1 e 3 m/s^2 , respectivamente.

A aceleração escolhida pelo agente é a aceleração que age sobre o carro, porém existe a aceleração contrária ou a favor que é função da inclinação da pista. A Equação 6 demonstra o cálculo da aceleração necessária para o veículo superar a inclinação dos próximos 100 metros.

$$a_{slope} = g * \frac{diferenca_de_altitude_100m}{\sqrt{(diferenca_de_altitude_100m)^2 + (100)^2}} \quad (6)$$

A aceleração necessária para superar a inclinação dos próximos 100 metros, a_{slope} , é a aceleração da gravidade multiplicada pelo seno do ângulo de inclinação da pista. O seno da inclinação dos próximos 100 metros é a diferença de altitude dividida pela hipotenusa do triângulo retângulo formado pela distância (100 metros) e a diferença de altitude.

Com a aceleração escolhida pelo agente e o cálculo da aceleração necessária para superar a inclinação dos próximos 100 metros, calcula-se a aceleração real do veículo. A Equação 7 mostra o cálculo da aceleração real do veículo.

$$a_r = a - a_{slope} \quad (7)$$

A partir das Equações 3 e 4, e das definições de acelerações, é possível definir uma função recompensa. Esta é definida por partes, de acordo com a Equação 8.

$$r(v, v_{i+1}, a, d_m, d_t) = \begin{cases} -10, & \text{se } v = 0, a < 0 \text{ obs.: ambiente encerrado} \\ v_{i+1} - v, & \text{se } v_{i+1} \leq 0 \text{ obs.: ambiente encerrado} \\ -20, & \text{se } v = 0, a = 0 \\ f_V/d_m - penalidades, & \text{se } a \geq 0 \\ f_{cruise}/d_m - penalidades, & \text{se } a < 0 \\ 10000, & \text{se } d_t = trajetototal \end{cases} \quad (8)$$

A Equação 8 mostra que, caso o agente esteja parado e tome a ação para andar de ré, ou seja, escolha uma aceleração negativa, a recompensa é definida em -10 e o ambiente é encerrado. Também é encerrado caso a aceleração faça com que a velocidade no próximo passo, v_{i+1} , seja negativa ou zero, sendo a recompensa deste caso igual à variação da velocidade entre o passo atual e o próximo passo. Caso ele esteja parado e escolha se manter parado, a recompensa é de -20, porém o ambiente continua.

As recompensas positivas da Equação 8 são definidas de acordo com a eficiência do veículo e a finalização do trajeto. Caso o agente tenha velocidade positiva

e escolha acelerar ou manter, a recompensa é definida pelo consumo f_V dividido pela distância percorrida d_m , calculada pela Equação 3, e subtraída a penalidade.

Desacelerar um veículo não consome combustível, e, no caso de veículos à combustão, não recuperam energia. Porém, para fins deste trabalho, o agente recebe a recompensa de acordo com a eficiência definida pelo consumo em velocidade de cruzeiro f_{cruise} dividido pela distância d_m com a redução das penalidades.

A Equação 9 mostra que as penalidades são calculadas em função da velocidade máxima permitida no trajeto, v_{max} , da velocidade indicada pelo GPS, v_{rec} , da velocidade do agente, v , e da aceleração escolhida pelo agente, a .

$$p(v_{max}, v_{rec}, v, a) = \begin{cases} (v - v_{max})^2, & \text{se } v > v_{max}, a \geq 0 \\ (v_{rec} - v), & \text{se } v < v_{rec}, a \leq 0 \end{cases} \quad (9)$$

A Equação 9 estabelece que, se o agente estiver acima da velocidade máxima (60 km/h para o trajeto 287) e escolher manter ou acelerar, a penalidade é calculada pelo quadrado da diferença. Esse cálculo se baseia na equação da energia cinética (Equação 10), que mostra que a energia de um impacto é proporcional ao quadrado da velocidade.

$$E_c = \frac{mv^2}{2} \quad (10)$$

A penalidade por manter ou desacelerar a velocidades abaixo da velocidade indicada pelo GPS (53 km/h para o trajeto 287 no momento da coleta dos dados) é proporcional à diferença entre a velocidade indicada e a velocidade do agente. Após o cálculo da penalidade, ela é subtraída do valor da eficiência obtida pelo agente.

Para calcular a eficiência momentânea do agente, é necessário definir a quantidade de combustível consumida pelo agente para aquele trecho. Para isso, utilizaremos as equações definidas por Kamal et al. (2011), as quais podem ser observadas nas Equações 2 e 11.

$$f_{cruise} = b_0 + b_1v + b_2v^2 + b_3v^3 \quad (11)$$

O consumo para a velocidade de cruzeiro, f_{cruise} , é uma função de terceiro grau da velocidade do veículo. Já o consumo com aceleração é a mesma do cruzeiro, porém com a adição de um termo, que é uma função quadrática da velocidade, multiplicado pela aceleração do veículo.

Para o cálculo das Equações 2 e 11, é necessário a obtenção dos parâmetros $b_0, b_1, b_2, b_3, c_0, c_1, c_2$. Estes são parâmetros variáveis de acordo com o motor de cada modelo de veículo. Para este trabalho, os valores utilizados são os mesmos que os encontrados para o motor utilizado por Kamal et al. (2011). Os parâmetros e os

respectivos valores podem ser vistos na Tabela 1.

Tabela 1 – Parâmetros do Veículo

parâmetro	valor
b_0	0,1569
b_1	0,0245
b_2	-0,0007415
b_3	0,00005975
c_0	0,07224
c_1	0,09681
c_2	0,001075

Fonte: Kamal et al. (2011)

Os parâmetros apresentados na Tabela 1 foram obtidos a partir das curvas de torque e velocidade do motor do veículo utilizado por Kamal et al. (2011). Como os dados utilizados neste trabalho não incluíam os modelos específicos dos veículos, optou-se por utilizar os parâmetros previamente definidos.

Para simular o movimento do veículo ao longo do trajeto, o DataFrame do percurso é reduzido de acordo com a distância percorrida pelo agente.

Para recompensar positivamente o agente por finalizar o trajeto, foi definida uma recompensa muito mais alta do que aquelas obtidas em cada episódio, a fim de ser capaz de identificar quando o agente chega ao final. A recompensa, caso a distância total percorrida, d_t , seja atingida quando o agente chega ao final do DataFrame, é de 10000 pontos.

Em resumo, conforme ilustrado na Figura 6, a cada passo do agente no ambiente, se o agente possui velocidade ou aceleração positivas, o percurso é reduzido, e a recompensa é calculada, sendo proporcional à eficiência energética e à capacidade de se manter dentro do limite de velocidades do ambiente.

3.5 ALGORITMO A3C DESENVOLVIDO

Para implementar o algoritmo A3C, foi desenvolvida uma estrutura de agentes ator-crítico capaz de calcular a vantagem de tomar uma ação sob a política global e treinar diversos agentes de forma paralela. O algoritmo foi implementado utilizando a linguagem Python com as bibliotecas TensorFlow, Keras, Threading e Multiprocessing.

Conforme destacado no Código 3.1, foi elaborada uma classe que define o modelo global de uma rede ator-crítico, composta por duas camadas. A primeira camada atua como ator, retornando a probabilidade de cada ação no ambiente. Essa camada pode ser visualizada nas linhas 7, 12 e 13 do trecho de código fornecido. A segunda camada funciona como crítico, fornecendo o valor do quão benéfico é para o agente estar naquele estado. Essa camada foi criada na linha 8 e realiza o cálculo do

valor nas linhas 14 e 15. Ambas possuem 400 neurônios cada, como visto na criação de ambas.

```

1 class ActorCriticModel(keras.Model):
2     def __init__(self, state_size, action_size):
3         super(ActorCriticModel, self).__init__()
4         self.state_size = state_size
5         self.action_size = action_size
6         self.dense1 = layers.Dense(400, activation='relu')
7         self.policy_logits = layers.Dense(action_size)
8         self.dense2 = layers.Dense(400, activation='relu')
9         self.values = layers.Dense(1)
10
11     def call(self, inputs):
12         x = self.dense1(np.array(inputs).reshape(-1,7))
13         logits = self.policy_logits(x)
14         v1 = self.dense2(np.array(inputs).reshape(-1,7))
15         values = self.values(v1)
16         return logits, values

```

Listing 3.1 – Modelo Global Ator-Crítico

Em seguida, foi implementada a classe do agente principal, apresentada no Código 3.2, cujo modelo é baseado na classe do agente A3C, conforme evidenciado na linha 8 do trecho. Este agente principal realiza a atualização da política global com base no feedback dos agentes treinados no ambiente. Com os agentes sendo executados em paralelo, esta classe é responsável por manter a política que irá reger o treinamento do algoritmo, sendo atualizada pelos agentes e enviando a política atualizada para os mesmos.

```

1 class MasterAgent():
2     def __init__(self, state_size, action_size):
3         self.state_size = state_size
4         self.action_size = action_size
5         self.optimizer = tf.compat.v1.train.AdamOptimizer(learning_rate
6         =0.0001, use_locking=True)
7         print(self.state_size, self.action_size)
8
9         self.global_model = ActorCriticModel(self.state_size, self.
            action_size)
10        self.global_model(tf.convert_to_tensor(np.random.random((1, self
11        .state_size))), dtype=tf.float32))

```

Listing 3.2 – Agente Principal

Para assegurar a assincronicidade necessária para o algoritmo A3C, foi implementada a classe *Worker*, como apresentado no Código 3.3. Essa classe herda

da classe *Thread*, conforme evidenciado na linha 1 do código. A função *run* desta classe foi sobrescrita para permitir que os agentes fossem treinados em paralelo, sendo chamados pelo agente principal. A função sobrescrita pode ser observada das linhas 21 a 44 do Código 3.3.

Os agentes da classe *Worker* serão treinados em paralelo e irão atualizar suas políticas locais buscando aprimorá-la e, também, atualizar os pesos da política global. Com essas atualizações todos os agentes em paralelo irão receber a mesma política de decisões, treinando a rede da classe *MasterAgent*.

```

1 class Worker(threading.Thread):
2     global_episode = 0
3     global_moving_avg_reward = 0
4     best_score = 0
5     save_lock = threading.Lock()
6     def __init__(self, state_size, action_size, global_model, optimizer, idx,
7         save_dir='results/4'):
8         super(Worker, self).__init__()
9         self.state_size = state_size
10        self.action_size = action_size
11        self.global_model = global_model
12        self.env = RouteEnvironment(data)
13        self.optimizer = optimizer
14        self.local_model = ActorCriticModel(self.state_size, self.
15            action_size)
16        self.index = idx
17        self.save_dir = save_dir
18        self.ep_loss = 0.0
19        self.ep_reward = 0.
20        self.rewards = []
21        self.gamma = 0.99
22
23    def run(self):
24        while Worker.global_episode < max_global_episodes:
25            current_state = self.env.reset()
26            mem.clear()
27            best_score_mem.clear()
28            self.ep_reward = 0.
29            ep_steps = 0
30            self.ep_loss = 0
31            steps_count = 0
32
33            done = False
34            while not done:
35                logits, _ = self.local_model(
36                    tf.compat.v1.convert_to_tensor(current_state, dtype=
37                        tf.float32))

```

```

35         probs = tf.nn.log_softmax(logits)
36         action = np.random.choice(self.action_size, p=np.exp(
probs.numpy()[0]))
37
38         current_state, new_state, reward, done = self.env.step(
action)
39
40         self.ep_reward += reward
41         mem.store(current_state, action, reward)
42         best_score_mem.store(current_state, action, reward)
43         MasterAgent.rewards.append(reward)
44         steps_count += 1

```

Listing 3.3 – Agentes Workers

Além disso, para incluir a função de vantagem na rede A3C, foi criada a função de cálculo de perda na classe *Worker*. Nessa função, também foi adicionado o parâmetro de temperatura, que auxilia no controle do nível de exploração do ambiente pelo agente.

Finalmente, a função de execução do agente faz com que este escolha ações de acordo com a política local e, a cada passo, atualize a política global. Além disso, a política local é atualizada de acordo com a global a cada 15 passos, conforme ilustrado no Código 3.4.

```

1 if steps_count == 15 or done:
2     with tf.GradientTape() as tape:
3         total_loss = self.compute_loss(done, new_state,
mem, self.gamma)
4         self.ep_loss += total_loss
5         grads = tape.gradient(total_loss, self.local_model.
trainable_weights)
6         self.optimizer.apply_gradients(zip(grads, self.
global_model.trainable_weights))
7         self.local_model.set_weights(self.global_model.
get_weights())
8         mem.clear()

```

Listing 3.4 – Atualização da Política

O ambiente, por sua vez, é atualizado a cada passo (*step*), onde são calculadas as mudanças nos parâmetros do estado, como aceleração, distância percorrida, consumo de combustível, e assim por diante. As recompensas são então calculadas com base nessas mudanças, levando em consideração a eficiência do veículo em termos de quilômetros por litro, penalidades por velocidade inadequada e ações indesejadas. Esta função é vista no Trecho 3.5 de forma reduzida.

No código do ambiente apresentado no Trecho 3.5, são realizados os cálculos das recompensas recebidas pelo agente. As linhas de código de 10 a 16 definem a

aceleração com base na ação escolhida pelo agente. As linhas 22, 26, 35, 53, 54 e 64 representam os cálculos das recompensas, enquanto as linhas 41 e 43 correspondem às penalidades. Todos esses cálculos, assim como a lógica desenvolvida para o ambiente, foram detalhados na Seção 3.4.

```

1 def step(self, action):
2     #atribui o index do proximo passo
3     next_step = self.current_step + 1
4
5     #checa se passo atual e o ultimo ou o trajeto foi encerrado
6     if next_step < self.total_steps:
7         #recebe informacoes do estado: distancia total viajada,
8         distancia remanescente, altitude, diferenca de altitude para os
9         proximos 100m, velocidade atual, velocidade maxima e velocidade
10        indicada
11
12        #define aceleracao baseada na acao escolhida
13        acceleration = (
14            -3 if action == 0
15            else -1 if action == 1
16            else 0 if action == 2
17            else 1 if action == 3
18            else 3
19        )
20
21        #calcula aceleracao necessaria para passar inclinacao a
22        frente
23        #calcula aceleracao real
24
25        if speed == 0 and acceleration < 0:
26            reward = -10
27            next_state = None
28            done = True
29        elif speed == 0 and acceleration == 0:
30            reward = -20
31            next_state = state
32            done = False
33        else:
34            #se tem velocidade ou acelera, ou ambos, entao
35            #calcula distancia viajada no passo, dataframe dirigido,
36            atualiza total de passos e proxima velocidade
37
38            if next_speed <= 0:
39                #se a aceleracao fara o veiculo parar ou andar de re
40                reward = next_speed - speed
41                next_state = None

```

```

37         done = True
38     else:
39         #carro continua andando
40         if speed < min_speed and acceleration < 0:
41             self.penalty = min_speed - speed
42         if speed > max_speed and acceleration > 0:
43             self.penalty = pow((speed - max_speed),2)
44
45         #estima consumo em mL/s e passa para litros mL/s * s
46         /1000
47         if acceleration >=0:
48             consumption_mls = (b0+b1*speed+b2*pow(speed,2)+
49 b3*pow(speed,3)+acceleration*(c0+c1*speed+c2*pow(speed,2)))
50             consumption_l = consumption_mls*TIME_STEP/1000
51         else:
52             consumption_mls = (b0+b1*speed+b2*pow(speed,2)+
53 b3*pow(speed,3))
54             consumption_l = consumption_mls*TIME_STEP/1000
55
56         reward = distance_traveled_km/consumption_l
57         reward -= self.penalty
58
59         self.penalty = 0
60
61         #normaliza a velocidade calculada em relacao a
62         velocidade maxima permitida
63         #define proximo passo como proxima linha do
64         dataframe
65
66         done = False
67         self.current_step = next_step
68     else:
69         reward = 10000
70         next_state = None
71         done = True
72     return state, next_state, reward, done

```

Listing 3.5 – Ambiente (código reduzido)

O código desenvolvido está disponibilizado no GitHub¹.

¹ GitHub com o Código desenvolvido: <https://github.com/paulogorla/consumption-a3c>

4 RESULTADOS

Neste capítulo, serão apresentados o ambiente computacional no qual o algoritmo foi desenvolvido e treinado, discutindo a biblioteca e linguagem utilizadas, bem como os principais componentes que influenciam a execução do sistema. Além disso, o capítulo introduz os ambientes nos quais os agentes atuaram, conforme detalhado na Seção 3.4.

Após a contextualização dos ambientes de treinamento e coleta de resultados, será realizada uma análise e comparação entre eles, além de confrontá-los com os dados reais apresentados na Seção 3.3. Os parâmetros a serem avaliados incluem Eficiência Energética, Velocidade Média e Pontuação. Para melhor compreensão, será detalhada a abordagem ou métricas utilizadas na mensuração desses parâmetros.

4.1 TREINAMENTO E COLETA DE RESULTADOS

Os agentes foram desenvolvidos utilizando a biblioteca TensorFlow do Python. O Ambiente de Desenvolvimento Integrado (IDE) utilizado foi o Visual Studio Code, distribuído pela Microsoft. O notebook utilizado foi um Acer Nitro 5, equipado com um processador Intel Core i7 de 7^a geração com clock de 2800 MHz, 8 núcleos e uma placa de vídeo Nvidia GTX 1050 de 4GB. O algoritmo foi treinado sem prioridade no sistema operacional Windows com outros aplicativos abertos.

O número de agentes treinados depende da quantidade de núcleos disponíveis no processador do computador utilizado. Para este trabalho, foi possível paralelizar o treinamento de 8 agentes. Eles são executados em paralelo e treinados conjuntamente, acompanhando a evolução da política global. Para a coleta dos resultados dos episódios e a execução da atualização da política, foi criada uma classe de memória para armazenar o estado, ação e recompensa de cada etapa dos agentes em um episódio.

O treinamento da rede foi realizado ao longo de 30.000 episódios, equivalente a percorrer o trajeto 30 mil vezes. Durante o treinamento, além de realizar as atualizações de política conforme mencionado na Seção 3.5, o modelo com a maior pontuação atingida é salvo periodicamente. Além disso, uma planilha contendo as observações do agente, as ações tomadas e as recompensas do episódio de maior pontuação é registrada.

A planilha proporciona a capacidade de estimar o consumo final do agente para comparação com o consumo real, atendendo ao requisito de indicar as ações que o motorista deve tomar durante o percurso para aumentar a eficiência energética. Para a estimativa do consumo, são realizados cálculos com base nas Equações 2 e 11, levando em conta que este trabalho visa a eficiência em veículos à combustão, considerou-se que desacelerar o veículo não consome combustível.

Outro resultado significativo do treinamento é o número de episódios necessários para a convergência da rede A3C, o qual pode ser observado através do gráfico da média móvel de recompensas gerado pela classe de memória.

Os treinamentos foram conduzidos em dois ambientes com heurísticas similares, mas diferem no cálculo da penalidade por exceder a velocidade máxima permitida pela pista, representada por v_{max} . A Tabela 2 apresenta o cálculo da penalidade para cada ambiente de treinamento.

Tabela 2 – Ambientes de Treinamento

	Penalidade por $v > v_{max}$
Ambiente 1	$(v - v_{max})^2$
Ambiente 2	$10 \times (v - v_{max})$

Fonte: Produzido pelo Autor

O Ambiente 1 segue a descrição apresentada na Seção 3.4. Já o Ambiente 2 é uma variação do Ambiente 1, onde o cálculo da penalidade por ultrapassar a velocidade, mostrado na Equação 9, é 10 vezes a diferença das velocidades, em vez do quadrado. Essa alteração foi feita para analisar o impacto da penalidade na convergência do agente.

4.2 RESULTADOS

Para medir os resultados do ganho de eficiência obtido pela rede A3C em um percurso pré-definido, será comparado o episódio com a maior recompensa obtida pelos agentes. Esse episódio é salvo em uma tabela, a qual seria utilizada para fazer as recomendações. A Tabela 3 mostra a comparação entre o consumo obtido pelo motorista e o agente nos dois ambientes definidos.

Tabela 3 – Comparativo

	Eficiência ECU	Eficiência Estimada	Velocidade Média
Real	8.23 km/L	14.25 km/L	62.9 km/h
A3C Ambiente 1	-	15.46 km/L	60.0 km/h
A3C Ambiente 2	-	15.57 km/L	58.2 km/h

Fonte: Produzido pelo Autor

A Tabela 3 mostra as três amostras a serem comparadas, sendo o Real obtido pelo motorista, as eficiências do algoritmo A3C treinados nos Ambientes 1 e 2, conforme descrito na Seção 4.1. A coluna "Eficiência ECU" mostra a eficiência medida pela própria ECU do veículo utilizado no percurso. Já a "Eficiência Estimada" é o cálculo da eficiência de acordo com as Equações 2 e 11. A "Velocidade Média" é a média das velocidades momentâneas coletadas em um intervalo de 3 segundos.

Além da eficiência obtida pelo agente, é necessário medir sua capacidade de se manter entre a velocidade máxima e indicada na determinada pista. Para que seja possível a comparação entre os resultados do algoritmo A3C e os dados reais, foi calculada a eficiência estimada com as Equações 2 e 11. Já para o comparativo entre as velocidades médias, para o ambiente real, foi utilizada a velocidade média medida pela ECU do veículo. Para o algoritmo, a velocidade média é a média das velocidades calculadas pela Equação 4.

A eficiência observada pela porta OBD foi de 8,23 km/L. Por outro lado, a eficiência estimada pelas Equações 2 e 11 foi de 14,25 km/L. A velocidade média do trajeto foi de 62,9 km/h, o que está acima da velocidade máxima permitida pela pista de 60 km/h. Portanto, o motorista do ambiente real onde os dados foram coletados, por si só, não foi capaz de se manter dentro do limite de velocidade.

O melhor episódio, obtido pelo treinamento do algoritmo A3C no Ambiente 1, apresentou um consumo médio estimado de 15,46 km/L, considerando que desacelerar não consome combustível, e uma velocidade média de 60,01 km/h. Comparando ao consumo estimado dos dados reais, houve um aumento de eficiência de 8,5%. Além disso, manteve a média de velocidade dentro do limite estipulado, apesar de ultrapassá-la em alguns trechos. Ou seja, em geral, o desempenho do agente no episódio de maior recompensa no Ambiente 1 foi melhor que o apresentado pelo motorista no cenário real.

Além do resultado em relação à eficiência e velocidade média, outro parâmetro a ser analisado é a convergência do algoritmo. Para isso, foi traçado um gráfico para as recompensas ao longo do treinamento. A Figura 7 mostra a média móvel, com uma janela de 50 episódios, da recompensa no treinamento no Ambiente 1.

A Figura 7 mostra que o agente 0 passa a finalizar o trajeto apenas após 2.500 tentativas, ou seja, o agente precisa tentar percorrer o trajeto 2.500 vezes para aprender a completá-lo a maioria das vezes. O agente não foi capaz de alcançar a média móvel em um valor de 90% do episódio de maior recompensa, ou seja, precisaria de um treinamento mais longo para alcançar a convergência.

Apesar de não alcançar a convergência, o algoritmo foi capaz de atingir grandes pontuações. A pontuação final total do agente no melhor episódio do treinamento no Ambiente 1 foi de 15.744 pontos em um total de 455 observações. Para analisar o desempenho do agente neste episódio, podemos verificar o somatório total das recompensas ao longo do episódio para avaliar se o agente foi capaz de obter recompensas positivas consistentemente.

Utilizando a tabela salva pelo episódio da melhor recompensa, pode-se obter a recompensa para cada observação e ação do agente no ambiente, ou seja, as pontuações momentâneas recebidas pelo agente. A soma da recompensa ao longo

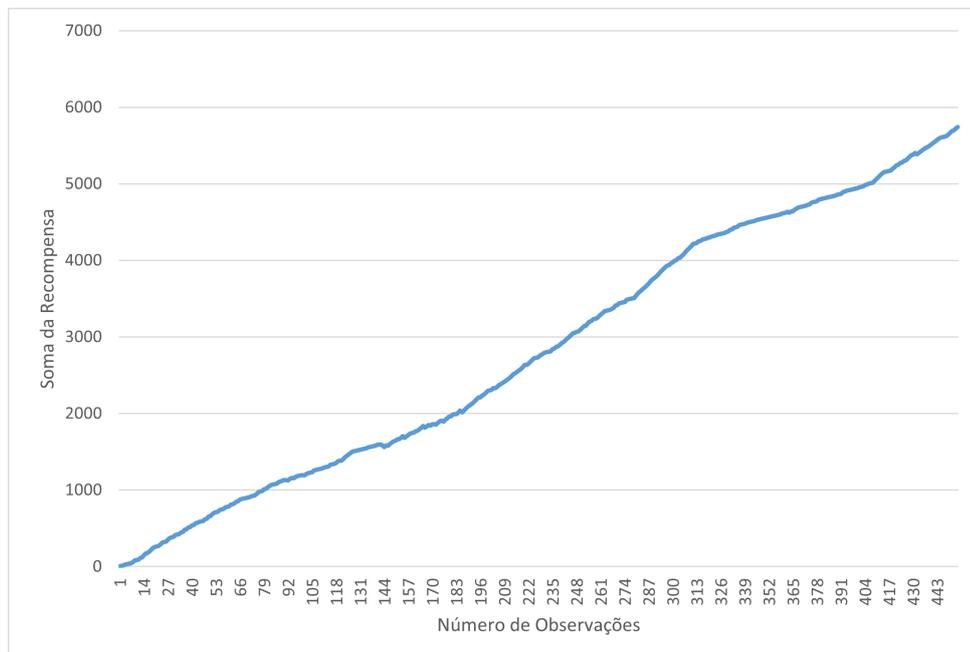
Figura 7 – Recompensa por episódio *Worker 0* Ambiente 1



Fonte: Produzida pelo Autor

do episódio gravado pelo algoritmo, excluindo a recompensa por finalizar o trajeto, é mostrada na Figura 8.

Figura 8 – Recompensa do melhor episódio Ambiente 1



Fonte: Produzida pelo Autor

Pela Figura 8, nota-se que, de forma geral, o agente foi capaz de tomar decisões que trouxeram recompensas positivas na maioria das vezes. Isso mostra que o agente saiu dos limites de velocidade em alguns instantes, porém, mesmo assim, o agente

visou alcançar uma eficiência maior a longo prazo.

As ações tomadas pelo agente no episódio com a melhor recompensa do treinamento no Ambiente 1 podem ser utilizadas para recomendar ações ao motorista com o objetivo de aumentar a eficiência. Para isso, as ações são representadas para o condutor por meio de setas, tornando a decisão o mais intuitiva possível. As ações tomadas pelo agente no melhor episódio do primeiro teste são demonstradas para o usuário no painel, conforme visto na Tabela 4.

Tabela 4 – Ações - Ambiente 1

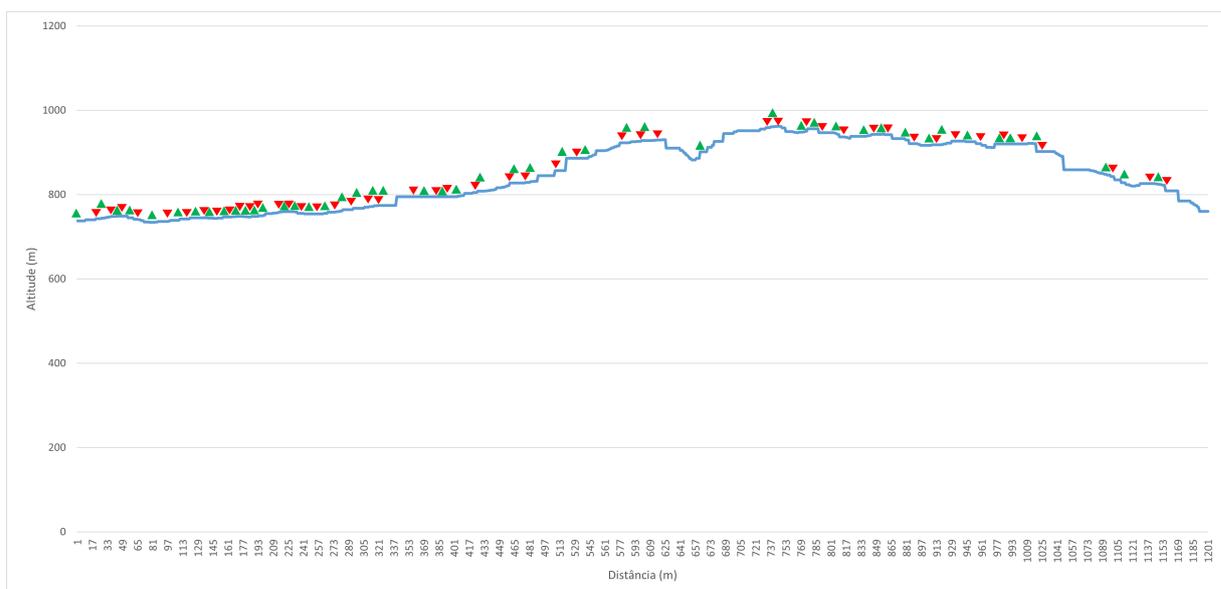
action	acceleration	interface
1	$-1m/s$	▼
3	$1m/s$	▲

Fonte: Produzido pelo Autor

A Tabela 4 mostra as ações únicas que o agente tomou, a aceleração correspondente no ambiente de acordo com a Equação 8, e como cada ação seria mostrada para o motorista. Nota-se que o agente optou apenas por acelerações e desacelerações leves e, em nenhum momento, manteve a velocidade constante nem acelerou bruscamente.

As ações tomadas pelo agente variam de acordo com a observação feita por ele em cada instante. Ou seja, o agente deve definir ações com base no trajeto já percorrido e no que ainda está por percorrer. Para verificar a coerência das ações ao longo do percurso, a Figura 9 mostra as ações do agente durante a primeira metade do trajeto no Ambiente 1.

Figura 9 – Recomendações Primeira Metade do Trajeto - Ambiente 1



Fonte: Produzida pelo Autor

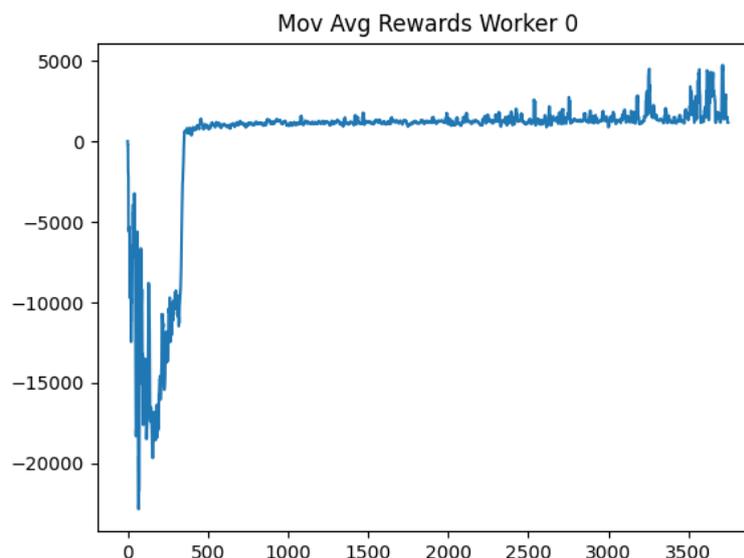
A Figura 9 apresenta a altitude do trajeto 287 ao longo da distância percorrida. As setas indicam as mudanças de ação do agente, ou seja, todo o trecho entre duas setas apresenta a mesma ação da primeira seta. Por exemplo, no primeiro trecho, o agente acelerou a uma taxa de 1 m/s^2 até os 240 metros, desacelerou por 10 metros e, em seguida, acelerou novamente.

A análise da figura permite observar que, nos trechos mais planos, o agente realizou diversas trocas de aceleração para manter a velocidade dentro do limite permitido. Em subidas, o agente acelerou para aumentar a velocidade, enquanto em descidas, o agente desacelerou para diminuir a velocidade.

A coleta dos resultados do treinamento no Ambiente 2 foi realizada da mesma forma que no Ambiente 1. O melhor episódio deste ambiente apresentou uma eficiência média estimada de 15,57 km/L e uma velocidade média de 58,20 km/h. Em comparação com os dados reais, a eficiência estimada foi 9,3% maior, e 0,7% maior do que o melhor episódio do Ambiente 1. Além disso, o agente conseguiu manter a velocidade média dentro do limite, e, ainda, inferior à do treinamento no primeiro ambiente.

Para analisar a convergência do algoritmo no Ambiente 2, foi traçado um gráfico da média móvel da recompensa por episódio do agente 0, assim como o traçado para o Ambiente 1. A Figura 10 mostra a média móvel da recompensa no treinamento no Ambiente 2, com uma janela de 50 episódios.

Figura 10 – Recompensa por episódio *Worker 0* Ambiente 2



Fonte: Produzida pelo Autor

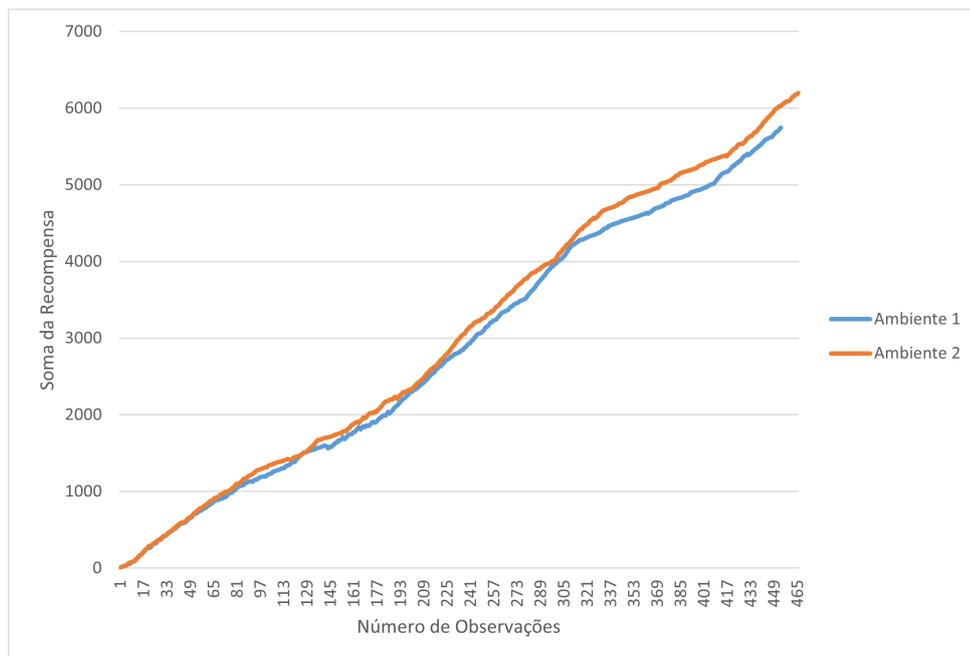
Pela Figura 10, observa-se que o agente não conseguiu finalizar o trajeto na maioria das vezes, e a média móvel da recompensa nunca atingiu 90% do valor do melhor episódio. Portanto, seria necessário um treinamento mais longo ou a alteração

de parâmetros para melhorar o desempenho do agente.

Comparando a Figura 10 com a Figura 7, nota-se que, ao final dos 30.000 episódios de treinamento, o algoritmo chegou mais próximo de alcançar uma convergência no Ambiente 1, ao se utilizar o quadrado da diferença das velocidades como penalidade. Mesmo variando quadraticamente, a penalidade, em geral, é menor que no Ambiente 2, onde a diferença das velocidades é multiplicada por 10, o que auxiliou na convergência do algoritmo.

Apesar de ter obtido uma convergência mais lenta, o algoritmo no Ambiente 2 conseguiu gerar um episódio com pontuação mais alta que o treinado no Ambiente 1. A Figura 11 apresenta um comparativo da soma da recompensa total do melhor episódio de cada Ambiente ao longo das observações feitas e ações tomadas pelo agente, sem considerar o último passo onde recebe a recompensa de 10.000 por finalizar o trajeto.

Figura 11 – Comparativo da Soma da Recompensa no Melhor Episódio



Fonte: Produzida pelo Autor

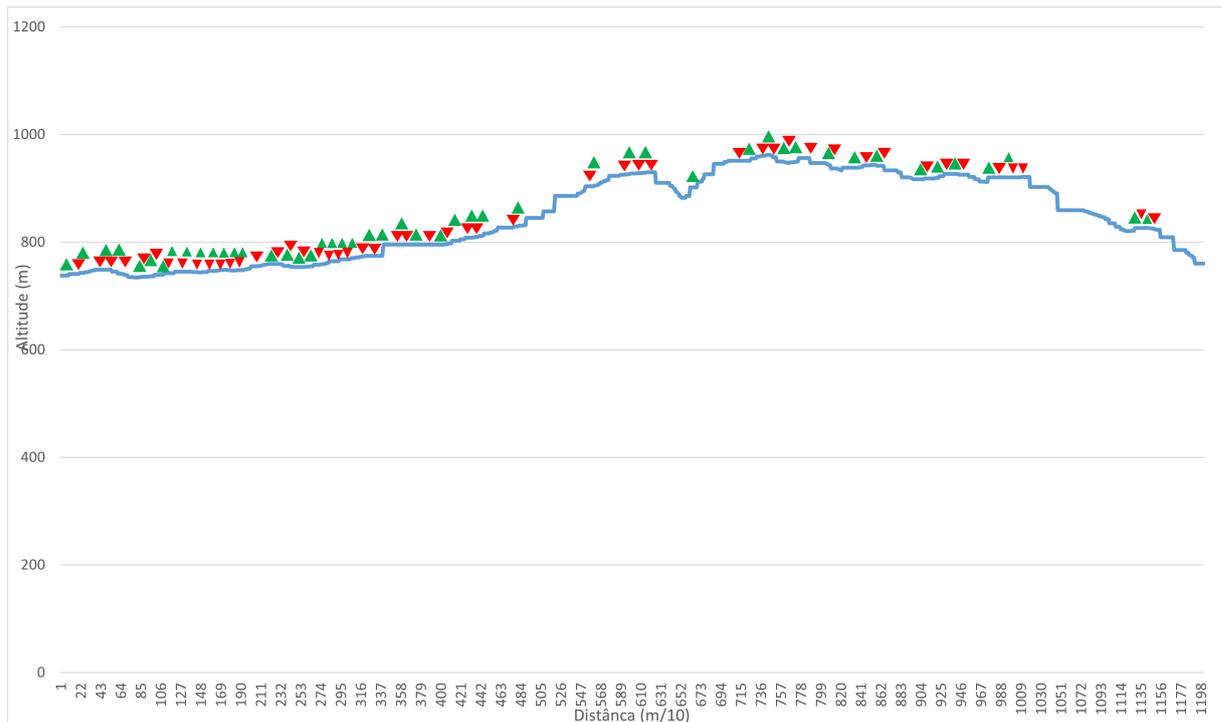
Como visto na Figura 11, o treinamento no Ambiente 2 foi capaz de gerar um episódio com pontuação total de 16.201, 7,9% maior que no Ambiente 1 quando removemos os 10.000 de pontuação por finalizar o trajeto. No Ambiente 2, o agente precisou de mais tempo para finalizar o trajeto, o que é mostrado pelo maior número de observações. Mesmo com essa pequena diferença, é possível notar que na maior parte do trajeto o agente alcançou pontuações momentâneas maiores no Ambiente 2.

Esta pontuação maior se deve por ter alcançado uma eficiência maior, mas também por ter sido capaz de se manter dentro dos limites de velocidade em mais trechos. As ações escolhidas pelo agente no melhor episódio do Ambiente 2 também foram apenas as 2 apresentadas na Tabela 4, ou seja, o agente tomou as mesmas

decisões, mas em momentos que favoreciam a eficiência do veículo.

Além das ações tomadas nos momentos que favoreciam a eficiência, a ordem das decisões também possibilitou ao veículo permanecer em velocidades que reduziram as penalidades recebidas. Para verificar a diferença entre as tomadas de decisões, a Figura 12 mostra as ações tomadas pelo agente ao longo da primeira metade do trajeto no melhor episódio do treinamento no Ambiente 2.

Figura 12 – Recomendações Primeira Metade do Trajeto - Ambiente 2



Fonte: Produzida pelo Autor

Pela Figura 12, nota-se que, assim como no Ambiente 1, em trechos planos o agente variou várias vezes a aceleração, visando manter-se dentro dos limites de velocidades. Da mesma forma, acelerou em subidas e desacelerou em descidas. Houve também pouca diferença na quantidade de trocas de aceleração, passando de 96 para 100 trocas na primeira metade do trajeto. Portanto, o agente foi mais eficiente na escolha dos momentos de troca de aceleração.

Ademais, visto que, em ambos os ambientes, os agentes se mantiveram dentro do limite de velocidades, o trajeto percorrido de forma mais eficiente não apresentou impactos significativos no tempo de trajeto. No Ambiente 1, considerando a velocidade média obtida, o agente demorou 1 minuto a mais para finalizar o trajeto que o motorista no ambiente real. Já no Ambiente 2, este impacto foi de aproximadamente 2 minutos.

5 CONCLUSÕES

O aumento do uso de Inteligência Artificial em diversas áreas decorre da capacidade da IA em resolver problemas de alta complexidade de maneira eficiente e com pouco conhecimento prévio sobre o processo complexo. No entanto, para atingir tal eficácia, é necessário um poder computacional suficiente, a coleta de dados coerentes com o problema a ser solucionado e a modelagem artificial desse problema.

Com base na utilização de dados coletados por GPS e pela porta OBD de automóveis, juntamente com estimativas obtidas por meio do estudo do consumo em trajetos com desníveis, foi criado um ambiente simulado para representar um trajeto com variações de elevação. Neste cenário, foram desenvolvidos agentes utilizando o algoritmo A3C, com a finalidade de interagir com o ambiente e buscar a maximização da eficiência energética.

Com o treinamento dos agentes nos ambientes projetados, o algoritmo salvou os episódios com as maiores pontuações em cada ambiente. Isso possibilitou a comparação entre os dados coletados por um motorista no trajeto real e a capacidade dos agentes em alcançar uma eficiência superior ao definir a aceleração ou desaceleração considerando as variações de elevação do trajeto.

O trabalho apresenta resultados expressivos na aplicação do algoritmo A3C para aprimorar a eficiência energética em trajetos com desníveis. Os agentes treinados demonstraram um aumento notável de 8,5% e 9,3% na eficiência nos Ambientes 1 e 2, respectivamente, comparados ao motorista real. Além disso, os agentes conseguiram manter a velocidade média dentro dos limites estipulados, indicando uma adaptação eficaz ao relevo do trajeto, desta forma não teve impacto significativo no tempo de trajeto. Apesar de desafios na convergência do algoritmo, os agentes alcançaram pontuações elevadas, sugerindo uma consistente capacidade de tomar decisões que resultam em recompensas positivas ao longo dos episódios. Esses resultados destacam o potencial da IA em contribuir significativamente para a eficiência energética veicular, promovendo não apenas economia de combustível, mas também segurança e conformidade com regulamentações de trânsito.

Conclui-se que a aplicação do algoritmo A3C apresenta-se como uma ferramenta promissora para otimizar as decisões dos motoristas em trajetos com desníveis, resultando em significativos ganhos de eficiência energética. Apesar das limitações discutidas no capítulo, os benefícios potenciais dessa abordagem estendem-se não apenas aos consumidores, mas também têm impactos positivos na indústria e no governo. A melhoria na eficiência veicular não apenas reduz os custos operacionais para os condutores, mas também contribui para a sustentabilidade e conformidade com regulamentações ambientais, promovendo uma condução mais econômica e

ecologicamente consciente.

Apesar dos resultados positivos, é crucial reconhecer a existência de dados adicionais não considerados neste estudo, como curvas, condições específicas da pista e fatores climáticos, que desempenham um papel crucial nas decisões de aceleração dos motoristas. Além disso, diversos parâmetros no modelo A3C e na simulação do ambiente podem ser ajustados, impactando os resultados de acordo com a aplicação específica.

O trabalho também aponta para oportunidades significativas de otimização, tanto no aprimoramento do modelo do algoritmo A3C quanto na modelagem do ambiente real. Isso inclui a busca por uma função de estimativa do consumo mais precisa, a consideração de uma variedade mais ampla de cenários de condução, e a exploração de melhorias nos parâmetros do modelo, como o aumento da quantidade de agentes treinados simultaneamente, a variação no número de neurônios artificiais, a experimentação com diferentes temperaturas e taxas de aprendizagem, entre outros. Essas possibilidades de otimização oferecem caminhos promissores para aprimorar ainda mais a eficácia do algoritmo e sua aplicabilidade em cenários automotivos diversos.

Uma perspectiva interessante para futuras pesquisas seria o desenvolvimento de uma Interface Homem-Máquina (IHM) destinada a criar uma aplicação prática para os motoristas. Ao integrar essa IHM em um sistema embarcado equipado com um algoritmo A3C treinado, seria possível fornecer ao condutor orientações intuitivas, visando otimizar a eficiência energética durante a condução. Essa abordagem promissora poderia contribuir para a implementação prática de soluções baseadas em inteligência artificial no contexto automotivo, oferecendo benefícios tangíveis aos motoristas em termos de eficiência e sustentabilidade.

REFERÊNCIAS

- ALMANSOUR, M. Electric vehicles (ev) and sustainability: Consumer response to twin transition, the role of e-businesses and digital marketing. **Technology in Society**, v. 71, p. 102–135, 2022.
- BISHOP, C. M.; NASRABADI, N. M. **Pattern recognition and machine learning**. Heidelberg: Springer, 2006.
- CHAPELLE, O.; SCHOLKOPF, B.; ZIEN, A. Semi-supervised learning. 2006. **The MIT Press View Article**, v. 2, p. 542–542, 2006.
- CUSENZA, M. A. et al. Energy and environmental assessment of a traction lithium-ion battery pack for plug-in hybrid electric vehicles. **Journal of cleaner production**, v. 215, p. 634–649, 2019. Acesso em: 18 jun. 2023.
- FAIZAN, M.; PILLAI, A. S. Dynamic task allocation and scheduling for multicore electronics control unit (ECU). *in*: **Proceedings of the 2019 3RD INTERNATIONAL CONFERENCE ON ELECTRONICS, COMMUNICATION AND AEROSPACE TECHNOLOGY (ICECA)**. p. 821–826, 2019. Disponível em: <https://doi.org/10.1109/ICECA.2019.8822086>. Acesso em: 18 jun. 2023.
- GÉRON, A. **Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow**. Sebastopol: O'Reilly Media, 2022.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. Heidelberg: Springer, 2009.
- HEIGIT GGBMH. **Open Route Service**. Heidelberg, 2022. Disponível em: <https://openrouteservice.org/>. Acesso em: 07 nov. 2023.
- HELFAND, G.; WOLVERTON, A. Evaluating the consumer response to fuel economy: A review of the literature. U.S. Environmental Protection Agency, 2009. Disponível em: <http://dx.doi.org/10.22004/ag.econ.280877>. Acesso em: 18 jun. 2023.
- HEYWOOD, J. B. **Internal combustion engine fundamentals**. New York: McGraw-Hill Education, 2018.
- IGOR SILVA. **Vehicular Dataset**. Minas Gerais, 2022. Disponível em: <https://www.kaggle.com/datasets/igorsilva02/vehicular-dataset?select=Dataset.csv>. Acesso em: 07 nov. 2023.
- KAELBLING, L. P.; LITTMAN, M. L.; MOORE, A. W. Reinforcement learning: A survey. **Journal of artificial intelligence research**, v. 4, p. 237–285, 1996. Acesso em: 12 out. 2023.
- KAMAL, M. et al. Ecological vehicle control on roads with up-down slopes. **Intelligent Transportation Systems**, v. 12, p. 783–794, oct. 2011. Acesso em: 18 jun. 2023.
- KAMAL, M. et al. Model predictive control of vehicles on urban roads for improved fuel economy. **Control Systems Technology, IEEE Transactions on**, v. 21, p. 831–841, 05 2013. Acesso em: 18 jun. 2023.

- LARMINIE, J.; LOWRY, J. **Electric vehicle technology explained**. New Jersey: John Wiley & Sons, 2012.
- LEVINE, S. et al. Offline reinforcement learning: Tutorial, review and perspectives on open problems. **arXiv e-prints**, v. 5, 2020. Acesso em: 18 jun. 2023.
- LI, L. Cognitive cars: A new frontier for adas research. **IEEE Trans Intell Trans Syst**, v. 1, p. 1–13, jan. 2011. Acesso em: 18 jun. 2023.
- LIU, X. et al. Enhancing the fuel-economy of v2i-assisted autonomous driving: A reinforcement learning approach. **IEEE Transactions on Vehicular Technology**, v. 69, n. 8, p. 8329–8342, 2020. Acesso em: 18 jun. 2023.
- MICHALEK, J. J.; PAPALAMBROS, P. Y.; SKERLOS, S. J. A study of fuel efficiency and emission policy impact on optimal vehicle design decisions. **Journal of Mechanical Design**, v. 126, n. 6, p. 1062–1070, 2004. Acesso em: 18 jun. 2023.
- MITCHELL, T. M. et al. **Machine learning**. New York: McGraw-hill, 2007.
- MNIH, V. et al. Asynchronous methods for deep reinforcement learning. In: **International conference on machine learning**. Nova Iorque: JMLR, 2016. p. 1928–1937. Disponível em: <http://proceedings.mlr.press/v48/mniha16.pdf>. Acesso em: 12 out. 2023.
- PERERA, A.; KAMALARUBAN, P. Applications of reinforcement learning in energy systems. **Renewable and Sustainable Energy Reviews**, v. 137, p. 110–618, 2021. Acesso em: 18 jun. 2023.
- POWELL, W. B. **Reinforcement Learning and Stochastic Optimization: A unified framework for sequential decisions**. Nova Jersey: John Wiley & Sons, 2022.
- REIF, K. **Brakes, Brake Control and Driver Assistance Systems: Function, Regulation and Components**. Wiesbaden: Springer, 2014.
- SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. Cambridge: MIT press, 2018.
- TUNNELL, J. et al. Toward improving vehicle fuel economy with adas. **SAE International Journal of Connected and Automated Vehicles**, v. 1, p. 81–92, 2018.
- WIERING, M.; OTTERLO, M. van. **Reinforcement Learning: State-of-the-Art**. Heidelberg: Springer, 2012.
- WINNER, H. et al. **Handbook of driver assistance systems**. Amsterdam: Springer International Publishing, 2014.
- WU, J. et al. Battery thermal- and health-constrained energy management for hybrid electric bus based on soft actor-critic drl algorithm. **IEEE Transactions on Industrial Informatics**, v. 17, n. 6, p. 3751–3761, 2021. Acesso em: 18 jun. 2023.
- ZANETTE, A.; WAINWRIGHT, M. J.; BRUNSKILL, E. Provable benefits of actor-critic methods for offline reinforcement learning. **Advances in neural information processing systems**, v. 34, p. 13626–13640, 2021. Acesso em: 18 jun. 2023.

ZHOU, M.; JIN, H.; WANG, W. A review of vehicle fuel consumption models to evaluate eco-driving and eco-routing. **Transportation Research Part D: Transport and Environment**, v. 49, p. 203–218, 2016. Acesso em: 18 jun. 2023.

ZHU, X.; GOLDBERG, A. B. **Introduction to semi-supervised learning**. London: Springer Nature, 2022.

ANEXO A - COLUNAS DO DATASET

Dado	Descrição	Unidade / Saída	Fonte
Id	Identificação da viagem	Inteiro	Rotulação
GPSTime	Tempo no GPS	Data e hora	OBD
DeviceTime	Tempo no Dispositivo	Data e hora	OBD
Longitude	Longitude	Graus	OBD
Latitude	Latitude	Graus	OBD
GPSSpeed(Meters/second)	Velocidade no GPS	Metros por segundo	OBD
HorizontalDilutionofPrecision	Grau de degradação do nível de acuracidade da posição horizontal	Graus	OBD
Altitude	Altitude	Metros	OBD
Bearing	Orientação	Graus	OBD
G(x)	Aceleração no eixo X	m/s ²	OBD
G(y)	Aceleração no eixo Y	m/s ²	OBD
G(z)	Aceleração no eixo Z	m/s ²	OBD
G(calibrated)	Valor de G combinado	m/s ²	OBD
0-100kphTime(s)	Desempenho	Segundo	OBD
0-100mphTime(s)	Desempenho	Segundo	OBD
0-200kphTime(s)	Desempenho	Segundo	OBD
0-30mphTime(s)	Desempenho	Segundo	OBD
0-60mphTime(s)	Desempenho	Segundo	OBD
1/4MileTime(s)	Desempenho	Segundo	OBD
1/8MileTime(s)	Desempenho	Segundo	OBD
100-0KphTime(s)	Desempenho	Segundo	OBD
100-200KphTime(s)	Desempenho	Segundo	OBD
40-60MphTime(s)	Desempenho	Segundo	OBD
60-0MphTime(s)	Desempenho	Segundo	OBD
60-120MphTime(s)	Desempenho	Segundo	OBD
60-130MphTime(s)	Desempenho	Segundo	OBD
60-80MphTime(s)	Desempenho	Segundo	OBD
80-100MphTime(s)	Desempenho	Segundo	OBD
80-120KphTime(s)	Desempenho	Segundo	OBD
AbsoluteThrottlePosition B(%)	Posição de aceleração absoluta		OBD
AccelerationSensor(Total)(g)	Sensores de Aceleração (Total)	g	OBD
AccelerationSensor(Xaxis)(g)	Sensores de Aceleração (Eixo X)	g	OBD

AccelerationSensor(Yaxis)(g)	Sensores de Aceleração (Eixo Y)	g	OBD
AccelerationSensor(Zaxis)(g)	Sensores de Aceleração (Eixo Z)	g	OBD
AcceleratorPedalPositionD(%)	Posição do pedal do acelerador	%	OBD
AcceleratorPedalPositionE(%)	Posição do pedal do acelerador	%	OBD
ActualEngine%Torque(%)	Torque real do motor	%	OBD
AirFuelRatio(Commanded):(1)	Relação de combustível de ar (comandada)		OBD
AirFuelRatio(Measured):(1)	Relação de combustível de ar (mensurado)		OBD
AmbientAirTemp(°C)	Temperatura do ar ambiente	Graus Celsius	OBD
AndroidDeviceBatteryLevel(%)	Nível da bateria do dispositivo Android	%	OBD
AverageTripSpeed(WhilstMovingOnly)(km/h)	Velocidade média de viagem (apenas em movimento)	km/h	OBD
AverageTripSpeed(WhilstStoppedorMoving)(km/h)	Velocidade média de viagem (enquanto parado ou em movimento)	km/h	OBD
BarometricPressure(fromVehicle)(psi)	Pressão do ar medida pela ECU	psi	OBD
CatalystTemperature(Bank1Sensor 1)(°C)	Temperatura do catalisador	Graus Celsius	OBD
ChargeAirCoolerTemperature (CACT)(°C)	Temperatura do Resfriador de Ar de Carga	Graus Celsius	OBD
CommandedEquivalenceRatio(lambda)	Razão de Equivalência Comandada	Lambda	OBD
CostPerMile/km(Instant)(R\$/km)	Custo por milha / km (instantâneo)	R\$/Km	OBD
CostPerMile/km(Trip)(R\$/km)	Custo por milha / km (Viagem)	R\$/Km	OBD
CO2(Average)(g/km)	CO ₂ (Média)	g/Km	OBD
CO2(Instantaneous)(g/km)	CO ₂ (Instantâneo)	g/Km	OBD
DistanceToEmpty(Estimated)(km)	Distância para acabar a gasolina (Estimativa)	km	OBD
DistanceTravelledSinceCodesCleared(km)	Distância percorrida desde que os códigos foram apagados	km	OBD
DistanceTravelledWithMIL/CEL(km)	Distância percorrida com MIL / CEL	litros por km	OBD
DPFBank1DeltaPressure(psi)	Pressão DPF do banco 1	psi	OBD
DPFBank1InletPressure(psi)	Pressão de Entrada do banco 1	psi	OBD
DPFBank1InletTemperature(°C)	Temperatura DPF	Graus Celsius	OBD
DriversDemandEngine%Torque(%)	Torque demandada pelo motorista	%	OBD
EGRCommanded(%)	Recirculação de gases de escape (comandado)	%	OBD
EGRError(%)	Erro de recirculação de gases de escape	%	OBD
EngineCoolantTemperature(°C)	Temperatura do líquido de arrefecimento do motor	Graus Celsius	OBD
EngineKW(AtTheWheels)(KW)	Quilowatt do motor (nas rodas)	KW	OBD
EngineLoad(%)	Carga do motor	%	OBD
EngineLoad(Absolute)(%)	Carga do motor (absoluta)	%	OBD
EngineOilTemperature(°C)	Temperatura do óleo do motor	Graus Celsius	OBD

EngineReferenceTorque(Nm)	Torque de referência do motor	Newton-metro	OBD
EngineRPM(RPM)	rotações por minuto do motor	RPM	OBD
EthanolFuel(%)	Combustível etanol	%	OBD
EvapSystemVapourPressure(Pa)	Pressão de Vapor do Sistema Evap	Pascal	OBD
ExhaustPressureBank1(psi)	Banco de Pressão de Escape	psi	OBD
FuelCost(Trip)(Cost)	Custo do combustível na viagem	R\$	OBD
FuelFlowRate/Hour(l/hr)	Taxa de fluxo de combustível por hora	Litros por hora	OBD
FuelFlowRate/Minute(cc/min)	Taxa de fluxo de combustível por minuto	Litros por minuto	OBD
FuelLevel(FromEngineECU)(%)	Nível de combustível (da ECU do motor)	%	OBD
FuelPressure(psi)	Pressão do Combustível	psi	OBD
FuelRailPressure(psi)	Pressão do trilho de combustível	psi	OBD
FuelRailPressure(RelativeToManifoldVacuum)(psi)	Pressão do trilho de combustível (relativa ao vácuo do coletor)	psi	OBD
FuelRate(DirectFromECU)	Taxa de combustível (diretamente da ECU)		OBD
FuelRemaining(CalculatedFromVehicleProfile)(%)	Combustível Restante(Calculado a Partir do Perfil do Veículo)	&	OBD
FuelTrimBank1LongTerm(%)	Banco de Combustível 1 (Longo prazo)	%	OBD
FuelTrimBank1ShortTerm(%)	Banco de Combustível 1 (Curto prazo)	%	OBD
FuelUsed(Trip)(l)	Gasolina utilizada na viagem	Litros	OBD
GPSAccuracy(m)	GPS Accuracy	Metros	OBD
GPSAltitude(m)	Altitude GPS	Metros	OBD
GPSBearing(°)	Orientação GPS	Graus	OBD
GPSLatitude(°)	Latitude GPS	Graus	OBD
GPSLongitude(°)	Longitude GPS	Graus	OBD
GPSvsOBDSpeedDifference(KM/h)	Diferença de velocidade GPS vs OBD	km/h	OBD
Horsepower(AtTheWheels)(hp)	Potência (Nas Rodas)	Cavalo vapor	OBD
HybridBatteryCharge(%)	Carga da bateria híbrida	%	OBD
IntakeAirTemperature(°C)	Temperatura do ar de admissão	Graus	OBD
IntakeManifoldPressure(psi)	Pressão do Coletor de Admissão	psi	OBD
KilometersPerLitre(Instant)(kpl)	Quilômetros por litro (instantâneo)	km/l	OBD
KilometersPerLitre(LongTermAverage)(kpl)	Quilômetros por litro (Média de longo prazo)	km/l	OBD
LitresPer100Kilometer(Instant)(l/100km)	Litros por 100 Quilômetros (Instantâneo)	litros por 100km	OBD
LitresPer100Kilometer(LongTermAverage)(l/100km)	Litros por 100 Quilômetros (Média de longo prazo)	litros por 100km	OBD
MassAirFlowRate	Taxa de fluxo de massa de ar	Gramas	OBD
MilesPerGallon(Instant)(mpg)	Milhas por Galão (Instantâneo)	Milhas por galão	OBD

MilesPerGallon(LongTermAverage)(mpg)	Milhas por Galão (Média de longo prazo)	Milhas por galão	OBD
PercentageOfCityDriving(%)	Porcentagem de Condução Urbana	%	OBD
PercentageOfHighwayDriving(%)	Porcentagem de Condução Rodoviária	%	OBD
PercentageOfIdleDriving(%)	Porcentagem de condução ociosa	%	OBD
RelativeAcceleratorPedalPosition(%)	Posição relativa do pedal do acelerador	%	OBD
RelativeThrottlePosition(%)	Posição relativa do pedal do acelerador	%	OBD
RunTimeSinceEngineStart(s)	Tempo de execução desde a(s) partida(s) do motor	Segundo	OBD
Speed(GPS)(KM/h)	Velocidade no GPS	km/h	OBD
Speed(OBD)(KM/h)	Velocidade no OBD	km/h	OBD
ThrottlePosition(Manifold)(%)	Posição do acelerador (Múltiplo)	%	OBD
TimingAdvance(°)	Tempo avançado	Graus	OBD
Torque(Nm)	Torque	Newton-metro	OBD
TripAverageKPL(kpl)	Média de quilômetros por litro	km/l	OBD
TripAverageLitres/100KM(l/100KM)	Média de quilômetros por litro por 100km	km/l	OBD
TripAverageMPG(mpg)	Média de galão por milhas	Milhas por galão	OBD
TripDistance(KM)	Distância de viagem	km	OBD
TripDistance(StoredInVehicleProfile)(KM)	Distância da viagem (armazenada no perfil do veículo)	km	OBD
TripTime(SinceJourneyStart)(s)	Tempo de viagem (desde o início da viagem)	Segundo	OBD
TripTime(WhilstMoving)(s)	Tempo de viagem (em movimento)	Segundo	OBD
TripTime(WhilstStationary)(s)	Tempo de viagem (enquanto parado)	Segundo	OBD
TurboBoost&VacuumGauge(psi)	Turbo Boost e medidor de vácuo	psi	OBD
Voltage(ControlModule)(V)	Voltagem (Módulo de Controle)	Volts	OBD
Voltage(OBDAdapter)(V)	Voltagem (Adaptador OBD)	Volts	OBD
VolumetricEfficiency(Calculated)(%)	Eficiência Volumétrica (Calculada)	%	OBD
Temperature	Temperatura	Graus Celsius	API Weather Channel
WeatherDescriptions	Descrição do tempo	Texto	API Weather Channel
WindSpeed	Velocidade do vento	km/h	API Weather Channel

WindDegree	Direção do vento	Inteiro	API Weather Channel
WindDir	Direção do vento	Texto	API Weather Channel
Pressure	Pressão atmosférica	Pascal	API Weather Channel
Precip	Precipitação	mm	API Weather Channel
Humidity	Humidade	kg/m ³	API Weather Channel
CloudCover	Cobertura de nuvens	%	API Weather Channel
FeelsLike	Sensação Térmica	Graus Celsius	API Weather Channel
UvIndex	Índice de UV	Inteiro	API Weather Channel
Visibility	Visibilidade no momento (Entre 0 e 10)	Inteiro	API Weather Channel
IsDay	Define se é dia ou não	Texto	API Weather Channel
HereMapsSpeedUncapped	Velocidade prevista na via (Pode ser acima do permitido)	km/h	API Here WeGo

HereMapsSpeed	Velocidade prevista na via	km/h	API Here WeGo
HereMapsFreeFlow	Velocidade de referência da via quando não há tráfego	km/h	API Here WeGo
HereMapsJamFactor	Quantidade de tráfego na via (Valor de 0 a 10, quanto maior o número, maior o tráfego)	Inteiro	API Here WeGo
HereMapsDuration	Tempo estimado para a viagem	Minutos	API Here WeGo
HereMapsDistance	Distância a ser percorrida na viagem	km	API Here WeGo
GoogleMapsDistance	Distância a ser percorrida na viagem	km	API Google Maps
GoogleMapsDuration	Tempo estimado para a viagem	Minutos	API Google Maps
Hour	Hora no momento da coleta	Hora	Rotulação
Driver	Motorista que está dirigindo o veículo	Inteiro	Rotulação
Vehicle	Veículo que está sendo utilizado	Inteiro	Rotulação
Scenary	Cenário de coleta de dados	Inteiro	Rotulação