Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM)
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)
Associação Instituto Tecnológico Vale (ITV)

Dissertação

MAPEAMENTO AUTOMATIZADO DE GASES COM ROBÔS

Maurício Souza Sathler

Ouro Preto
Minas Gerais, Brasil
2023

Maurício Souza Sathler

MAPEAMENTO AUTOMATIZADO DE GASES COM ROBÔS

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Gustavo Pessin, Ph.D.
Coorientador: Prof. Maíra Saboia, Ph.D.

Ouro Preto
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S253m Sathler, Mauricio Souza.
Mapeamento automatizado de gases com robôs. [manuscrito] / Mauricio Souza Sathler. - 2023.
65 f.: il.: color., gráf., tab., mapa.

Orientador: Prof. Dr. Gustavo Pessin.
Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.
Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Global Land Survey (GLS). 2. Robótica - Robótica Móvel. 3. Aprendizado do computador - Aprendizado por reforço. I. Pessin, Gustavo. II. Universidade Federal de Ouro Preto. III. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Maristela Sanches Lima Mesquita - CRB-1716

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
PROGR. POS GRAD. PROF. INST. CONT. E AUT.
PROCESSOS DE MIN.

FOLHA DE APROVAÇÃO

Maurício Souza Sathler

Mapeamento automatizado de gases com robôs

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração

Aprovada em 10 de outubro de 2023

Membros da banca

Doutor - Gustavo Pessin - Orientador - Instituto Tecnológico Vale
Doutor - Gustavo Medeiros Freitas - Universidade Federal de Minas Gerais
Doutor - Héctor Ignacio Azpúrua Perez-Imaz - Meta

Gustavo Pessin, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 10/12/2023

Documento assinado eletronicamente por **Bruno Nazário Coelho**, **COORDENADOR(A) DE CURSO DE PÓS-GRADUAÇÃO EM INST. CONTROLE AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 21/12/2023, às 11:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **0645283** e o código CRC **4CAAD4DC**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.017089/2023-35 SEI nº 0645283

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3552-7352 - www.ufop.br

Agradecimentos

Este trabalho é o resultado de um ciclo de muito esforço, dedicação e aprendizado, que não seria possível sem o apoio de muitas pessoas que me acompanharam nesta jornada. Gostaria de expressar minha profunda gratidão a todos eles.

Aos meus pais, Jucelena e Kleber, que me deram a vida e me ensinaram os valores que me guiaram até aqui. Vocês são minha maior inspiração e motivação. À minha irmã, Nathália, que sempre esteve ao meu lado, me apoiando nos momentos difíceis e compartilhando as alegrias. À minha namorada, Raissa sempre me inspirando e motivando, que me apoia para tudo. Ao meu primo, Gabriel, ao meu irmão de faculdade e mestrado, Bidas, e ao Alexandre, meus grandes amigos que me ajudaram a superar os desafios com maior facilidade e leveza. Os amigos de BH, Victor, Vitor, Arthur, Caique, Gabriel, David e a Ord, pelos momentos vividos.

Aos meus colegas de trabalho e do Laboratório do ITV, Hector, Lucas, Mário, Jacó, Speto, Nilton, Filipe, Buneku, Levi, Matheus, Fred, Saulo, André, Gustavo e Renato, que formaram uma equipe incrível e colaborativa. Aprendi muito com vocês e me diverti muito também.

Aos meus orientadores, Gustavo Pessin e Maíra, que me orientaram com sabedoria e paciência. Vocês foram fundamentais para a realização deste trabalho e para o meu crescimento profissional e pessoal.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

MAPEAMENTO AUTOMATIZADO DE GASES COM ROBÔS

Maurício Souza Sathler

Outubro/2023

Orientadores: Gustavo Pessin
Maíra Saboia

As operações de inspeção são fundamentais para assegurar a segurança operacional na indústria de mineração. Quando realizados em ambientes confinados, esses processos exigem atenção devido aos vários riscos envolvidos, incluindo a presença de gases nocivos e a falta de oxigênio. A presente dissertação compara diferentes modelos de entrada e arquiteturas de redes para o Deep Q Learning (DQN), uma técnica de aprendizado profundo por reforço. Isso inclui camadas totalmente conectadas, camadas convolucionais e convoluções consolidadas na literatura. O objetivo da dissertação é realizar o rastreamento de dióxido de carbono, configurando diferentes plumas em vários ambientes simulados, utilizando robôs móveis. Além das configurações necessárias para a simulação de uma pluma de gás, a dissertação detalha as configurações, parâmetros e arquitetura da rede de aprendizado profundo utilizada. Os resultados obtidos em ambientes simulados representativos demonstram a viabilidade da técnica proposta, abrindo caminho para futuras aplicações com robôs reais.

Palavras-chave: GSL, Robótica Móvel, Aprendizado Profundo por Reforço.

Macrotema: Mina; **Linha de Pesquisa:** Robótica Aplicada à Mineração; **Tema:** Saúde e Segurança; **Área Relacionada da Vale:** Instituto Tecnológico Vale.

Abstract

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

THESIS TITLE

Maurício Souza Sathler

October/2023

Advisors: Gustavo Pessin
Maíra Saboia

Inspection operations are essential to ensure operational safety in the mining industry. When carried out in confined environments, these processes require attention due to the various risks involved, including the presence of harmful gases and the lack of oxygen. This dissertation compares different input models and network architectures for Deep Q Learning (DQN), a deep reinforcement learning technique. This includes fully connected layers, convolutional layers and convolutions consolidated in the literature. The objective of the dissertation is to perform carbon dioxide tracking, setting up different plumes in various simulated environments, using mobile robots. In addition to the settings required for the simulation of a gas plume, the dissertation details the settings, parameters and architecture of the deep learning network used. The results obtained in representative simulated environments demonstrate the feasibility of the proposed technique, paving the way for future applications with real robots.

Keywords: GSL, Mobile Robotics, Deep Reinforcement Learning.

Macrotheme: Mine; **Research Line:** Robotics Applied to Mining; **Theme:** Health and Safety; **Related Area of Vale:**Vale Institute of Technology.

Lista de Figuras

Figura 1.1 Acidentes recentes relacionados a gases nas usinas da Gerdau e Usiminas. . 11
Figura 1.2 Robôs e Sistemas Autônomos. 13
Figura 1.3 EspeleoRobô realizando inspeção em ambiente real e simulado, usando reconstrução 3D por fotogrametria (Mina do Veloso, Ouro Preto-MG). 14
Figura 1.4 *Louisville Mega Cavern*. 16

Figura 2.1 Diagrama de Venn-Euler das categorias dos algoritmos. 20
Figura 2.2 Dissipação de uma pluma de gás. 22
Figura 2.3 Modificação do kernel baseado em leituras de vento. a) leitura do anemômetro = 0; b) c) d) leituras do anemômetro > 0 24
Figura 2.4 Comparação dos métodos GMRF e KernelDM+V 24
Figura 2.5 Diagrama de Venn-Euler das áreas do aprendizado de máquina. 25
Figura 2.6 Sistema de iterações no aprendizado por reforço. 28
Figura 2.7 Ambientes de exemplo PyRep. 29
Figura 2.8 Arquitetura da ResNet-18 30

Figura 3.1 Fluxo de execução das atividades 31
Figura 3.2 Pluma não turbulenta simulada. (Roxo ≈ 400, Vermelho ≈ 5) 32
Figura 3.3 Ambiente e EspeleoRobô simulados no Coppeliasim. 32
Figura 3.4 Arquitetura da rede utilizada. 33
Figura 3.5 Ambiente de treinamento com posições iniciais (P=preto) e fonte (vermelho) 38
Figura 3.6 Gráficos de resultado simulação com obstáculos. 39
Figura 3.7 Trajetória realizada por agente treinado. 40

Figura 4.1 Fluxo de execução das atividades para ambiente com plumas turbulêntas. . . 41
Figura 4.2 Ambientes gerados. 43
Figura 4.3 Mapeamento de concentração de gás em ambiente real utilizando o KernelDM+V adaptado (vermelho: alta concentração. azul: baixa concentração). . . . 44
Figura 4.4 Mapas de entrada da CNN. 46
Figura 4.5 Convolução proposta. 48

Figura 4.6 Posições de fontes de gás simuladas. Em vermelho posições utilizadas para treinamento e em verde posições utilizadas para testes (Teste1 = esquerda, Teste2 = direita). 48
Figura 4.7 Resultados gerados para o treinamento com o robô e a fonte em posições aleatórias. 49
Figura 4.8 Fonte gerada em diferentes posições para o treinamento. 50
Figura 4.9 Trajetórias feitas pelo agente treinado no ambiente 1. 51
Figura 4.10 Estrutura com a DenseNet. 52
Figura 4.11 Posições de fontes de gás simuladas. Em vermelho posições utilizadas para treinamento e em verde posições utilizadas para testes. (Teste1 = esquerda, Teste2= direita) 52
Figura 4.12 Resultado do treinamento no ambiente 2. 54
Figura 4.13 Movimentação do treinamento no ambiente 2. 55

Lista de Tabelas

Tabela 4.1 Tabela com parâmetros utilizados para a simulação. 42
Tabela 4.2 Comparação tempo para convergir por arquitetura. 49
Tabela 4.3 Resultados dos testes realizados no ambiente 1. 50
Tabela 4.4 Resultados dos testes realizados no ambiente 2. 53

Sumário

1 Introdução **11**

1.1 Contextualização . . . 12

1.2 Objetivo . . . 17

1.2.1 Objetivos Específicos . . . 17

1.3 Contribuições . . . 17

1.4 Local de Realização . . . 19

1.5 Organização . . . 19

2 Referencial teórico e fundamentação científica **20**

2.1 Localização de Fontes de Gás (GSL) . . . 20

2.1.1 Pluma de gás . . . 21

2.1.2 Simulação de robôs . . . 22

2.1.3 Simuladores de Dispersão de Gás . . . 23

2.2 Técnicas de Mapeamento . . . 24

2.3 Aprendizado de Máquina . . . 25

2.3.1 Aprendizado por Reforço . . . 25

2.3.2 Redes Convolucionais . . . 29

3 Rastreamento de fonte de gás em plumas não turbulentas **31**

3.1 Ambiente e Arquitetura . . . 31

3.1.1 Estado . . . 32

3.1.2 Sensor de gás simulado . . . 34

3.1.3 Recompensas . . . 34

3.1.4 Normalização . . . 35

3.1.5 *Exploration* e *Exploitation* . . . 36

3.1.6 Função de Custo . . . 36

3.1.7 Parâmetros . . . 36

3.2 Treinamento e Resultados . . . 37

4 Rastreamento de pontos de origem de gases em pluma turbulenta **41**

4.1 Simulador de robótica e dispersão de gás . . . 41

4.2 Técnica de Mapeamento 42
4.3 Parâmetros e Configurações 44
4.4 Treinamento, Ajustes e Resultados 47
4.4.1 Ambiente 1 47
4.4.2 Ambiente 2 52

5 Considerações finais e trabalhos futuros **56**
5.1 Trabalho Futuros 56

Referências Bibliográficas **57**

1. Introdução

O monitoramento de fonte de gás é uma tarefa importante na mineração, no entanto, é complexo, pelo fato desses locais serem extensos e em grande maioria não possuem capacidade para instalação de equipamentos fixos de monitoramento. Assim, se torna um problema complexo, e quando existe a possibilidade, ela é onerosa, pois a disposição de sensores de gás pelo ambiente demanda infraestrutura e grande quantidade de sensores. Isso fortalece a utilização de robôs móveis para substituir equipamentos fixos e pessoas nesses ambientes perigosos. A possibilidade de utilizar robôs móveis oferece uma ampla variedade de oportunidades, não somente no âmbito de monitoramento da qualidade do ar, como o sistema de sensores citados anteriormente, mas também é possível realizar diversas outras atividades, como: filmar, gerar mapas, monitoramento preditivo, exploração de novos ambientes, tirar informações sobre possíveis problemas, etc.

Além desses pontos os robôs também podem ser utilizados em situações de acidentes / desastres. Com o envio do robô em situação de desastre é possível detectar e atuar no foco do acidente. Outra possibilidade é explorar e mapear ambientes inexplorados, como cavidades naturais sem colocar em risco a vida humana. Por fim, as informações captadas pelos robôs por serem armazenadas, servem de insumo para outras áreas da indústria 4.0, como o *big data* e análise de dados.

Alguns processos da mineração utilizam, geram ou ocorrem em locais suscetíveis a gases, desde inertes até tóxicos e explosivos (SATHLER, 2021), o que dificulta o seu monitoramento. Historicamente, são conhecidos relatos de acidentes em ambientes industriais do setor mineral, em 2016 uma explosão de gasoduto causou a morte de 3 pessoas na Gerdau (Figura 1.1a) e em 2018 ocorreu uma grande explosão num gasômetro da Usiminas, deixando 34 feridos (Figura 1.1b), dentre vários outros acidentes (GLOBO, 2020) (GLOBO, 2022), (TEMPO, 2022) (DO VALE, 2023), demonstrando a importância do monitoramento dos gases na indústria como chave para manter a saúde e segurança operacional.

Figura 1.1: Acidentes recentes relacionados a gases nas usinas da Gerdau e Usiminas.

(a) Acidente na Gerdau. (b) Usina da Usiminas.

Fonte: (GLOBO, 2018) e (DO AÇO, 2016).

1.1. Contextualização

O campo da robótica faz parte do avanço industrial, mas robôs não são novidade no setor produtivo. Desde a década de 60 os robôs industriais vêm ganhando espaço, mas existe uma diferença entre os robôs que operavam antigamente e os atuais.

Antigamente as atividades exercidas por sistemas robóticos eram baseadas em movimentos repetitivos e limitados. Atualmente está acontecendo uma disruptura nessa forma de trabalhar com robôs, onde cada vez mais temos a presença de robôs em atividades que demandam alta percepção e interação com o ambiente, o que implica um alto nível de complexidade. Anteriormente existiam situações predefinidas e uma sequência de movimentos para cada uma delas.

Atualmente, o campo da inteligência artificial tem revolucionado o cenário da robótica. Nesse novo contexto, um objetivo é definido para o sistema inteligente, que por sua vez decide quais ações tomar para alcançar esse objetivo.

É importante notar que o resultado encontrado pelo sistema inteligente não é necessariamente conhecido pelo programador. Isso significa que não é necessário ter todas as possíveis situações e comportamentos programados, conforme destacado por Mitchell *et al.* (1990).

Esta área de estudo é denominada *Robotics and Autonomous Systems* (RAS) (WONG *et al.*, 2017). Alguns exemplos desses sistemas são apresentados na Figura 1.2, que apresenta diferentes tipos de robôs móveis, quadrúpedes, bípedes, caminhões, etc.

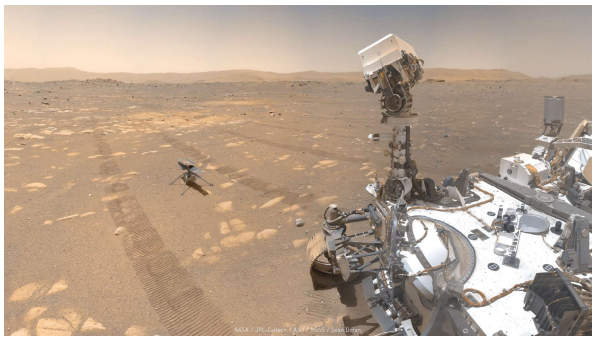
O projeto EspeleoRobô surgiu em 2014, como uma iniciativa da equipe de Espeleologia da Vale, para desenvolver um dispositivo robótico que pudesse realizar inspeções em áreas confinadas com mais segurança operacional. A princípio foi adquirido um robô comercial inspirado no robô hexápode RHex da Boston Dynamics. No entanto, este robô possuía algumas limitações, tais como seu sistema de locomoção mal sincronizado, e um sistema de comunicação de baixa eficiência, necessitando de dois receptores separados para a transmissão das imagens e um utilizado apenas para o envio de comandos ao robô (ROCHA *et al.*, 2019)

Por isso, o robô foi encaminhado ao Instituto Tecnológico Vale (ITV) para aprimoramento. Ao longo dos anos o ITV desenvolveu seu próprio robô, o EspeleoRobô, que manteve a inspiração no RHex, mas com modificações necessárias para suas aplicações. O EspeleoRobô possui um sistema intercambiável de locomoção e diversos sensores acoplados na parte superior, como câmeras RGB e LiDAR, que permitem o mapeamento 3D e 2D das cavidades (Figura 1.3).

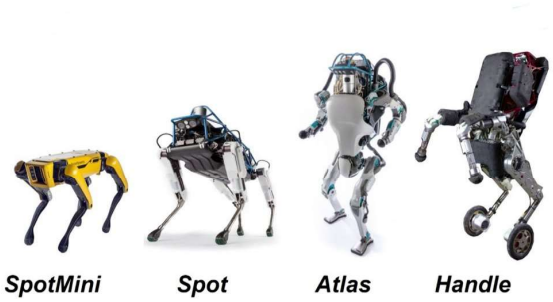
Durante o desenvolvimento do projeto, surgiram novas áreas de inspeção para o robô além de cavernas e outros ambientes naturais. Segundo Cid *et al.* (2020), algumas das tarefas mais demandantes em ambientes industriais são inspeções e mapeamento. Particularmente, regiões como tubulações, cavernas e galerias de barragens são áreas que exigem inspeções e mapeamentos frequentes.

Com o objetivo de promover a segurança nas operações de inspeção, foi realizado um estudo para identificar os gases nocivos presentes em mineração e cavidades naturais. Com

Figura 1.2: Robôs e Sistemas Autônomos.



(a) Rover perseverance e drone ingenuity



(b) Robôs da Boston Dynamics

(c) Veículos autônomos para transporte de pessoas e mercadorias.

Fonte: Dynamics (2020); Space (2022); Waymo (2022)

base nesse estudo foi desenvolvido um módulo de sensores de gás embarcado no EspeleoRobô (SATHLER, 2021). O robô apresenta a vantagem de poder realizar medições com alta precisão sem a necessidade de expor operadores a ambientes perigosos. A partir das demandas de inspeção, visando saúde e segurança foi realizado um estudo para mapear os gases nocivos que existem em ambientes de mineração e cavidades naturais.

Figura 1.3: EspeleoRobô realizando inspeção em ambiente real e simulado, usando reconstrução 3D por fotogrametria (Mina do Veloso, Ouro Preto-MG).

(a) Plataforma real na mina.

(b) Robô virtual na reconstrução 3D da mina no simulador.

Fonte: Cid *et al.* (2020)

O EspeleoRobô é operado por teleoperação. Apesar disso, ele já conta com algumas funcionalidades autônomas, demonstrando um avanço significativo em sua capacidade operacional. A meta para o futuro é tornar o EspeleoRobô completamente autônomo. Este objetivo representa um marco importante no desenvolvimento do robô, pois permitirá uma maior eficiência e independência em suas operações.

Uma vez que a estrutura do robô, incluindo os sensores de gás, esteja pronta, surgirá uma nova oportunidade de pesquisa. Esta oportunidade envolve a adição de mais uma capacidade autônoma ao EspeleoRobô: a habilidade de rastrear fontes de gases tóxicos. Isso ampliará ainda mais o potencial do robô em termos de segurança e eficácia.

Para o desenvolvimento da funcionalidade do EspeleoRobô de localizar a fonte de um gás tóxico no ambiente de forma autônoma é necessário utilizar técnicas de *Mobile Robot Olfaction* (MRO), que consistem em integrar sensores de gás e químicos em plataformas móveis e explorar maneiras de rastrear ou localizar a fonte de gás (MONROY *et al.*, 2017b). Dentro dessa área, uma das técnicas mais utilizadas é o *Gas Source Localization (GSL)* ou *Odor Source Localization* (OSL), que envia comandos para o robô, com o objetivo de encontrar a posição de uma fonte gás ou química de interesse no ambiente (KOWADLO e RUSSELL, 2008). Essa técnica se divide

em três etapas: (i) busca pela pluma, que é a região do ambiente onde há presença relevante do gás alvo; (ii) rastreamento da pluma, que consiste em seguir o gradiente de concentração do gás até a fonte; (iii) declaração da fonte, que é o momento em que o robô identifica a localização da fonte com base em algum critério. Outra técnica relacionada é o *Gas Distribution Model* (GDM), que não envia comandos ao robô, mas tem como proposta gerar algum tipo de mapa que representa as posições onde se encontram as concentrações ou variações de concentração do gás (LILIENTHAL *et al.*, 2007). Essa técnica pode ser usada para auxiliar o GSL ou para fins de análise e visualização dos dados.

Este é um problema complexo onde as soluções existentes atualmente são incapazes de performar bem em uma variedade de ambientes. Isso está diretamente ligado a característica de dispersão do gás, pois há diferentes a maneiras de dispersão, além de que ainda não existe um modelo matemático capaz de representar fielmente a dispersão dos gases.

Levando em consideração a complexidade matemática do modelo de dispersão dos gases, técnicas para rastreamento não são tão eficientes devido à grande variação do modelo de dispersão de gases nos diversos ambientes.

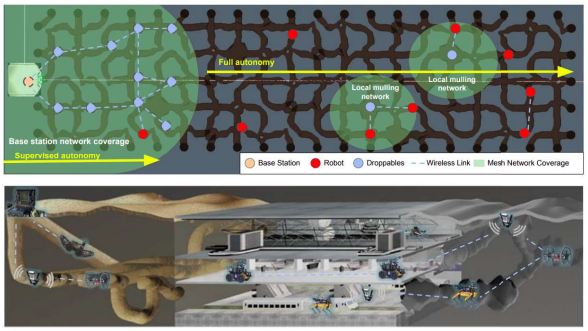
A *Defense Advanced Research Projects Agency* (DARPA), uma instituição pioneira no desenvolvimento de tecnologias inovadoras e responsável pela criação da primeira rede de computadores em 1969, tem incentivado o avanço da robótica inteligente. Entre 2018 e 2021, a DARPA promoveu o Subterranean Challenge (AGENCY, 2021), um desafio que estimulou o desenvolvimento de robôs autônomos capazes de explorar ambientes subterrâneos perigosos, como locais abandonados e minas subterrâneas. Além da exploração, os robôs deveriam realizar tarefas adicionais, como a localização de diversos artefatos, incluindo a detecção de fontes de gás CO_2.

A presença da atividade de detecção de fontes de gás no DARPA destaca a complexidade e a aplicabilidade prática desta tarefa. Esta dissertação contribuiu para este campo ao colaborar com a equipe CoStar (JPL, 2023), um grupo multidisciplinar composto principalmente por membros do Laboratório da NASA, o JPL, e da Universidade do MIT. Foi utilizado um algoritmo de mapeamento, que é apresentado nesta dissertação, para mapear a concentração de gás. No desafio foi necessário detectar uma fonte de gás artificial na caverna de *Louisville Mega Cavern* no estado de *Kentucky*, Estados Unidos da America (Figura 1.4a). Após o mapeamento, este mapa era então fornecido ao operador, que determinava a localização da fonte de gás, alcançando uma taxa de sucesso de 66% nas tentativas.

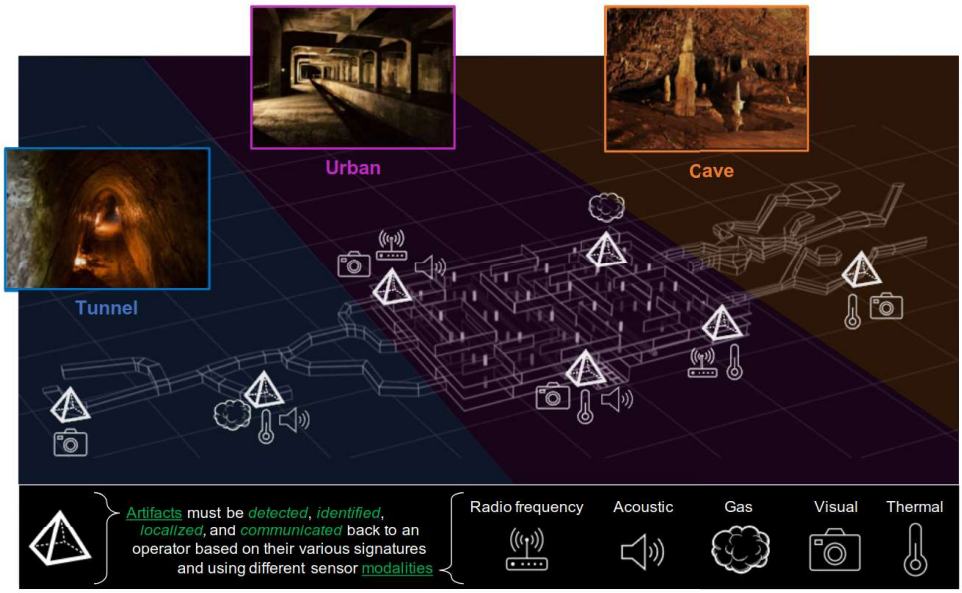
No entanto, o objetivo principal é permitir que o robô identifique autonomamente a localização do gás. Em algumas situações de navegação, o contato com o operador pode não ser possível, tornando o reconhecimento autônomo da fonte de gás uma característica desejável para o robô. Isso possibilita que o robô tome decisões autônomas para mitigar ou resolver problemas. Para isso, é proposto o uso da técnica de aprendizado por reforço profundo, que permite com que o robô aprenda a rastrear a fonte de gás a partir de exemplos e se adapte às condições do ambiente, melhorando a precisão e eficiência do rastreamento.

Figura 1.4: *Louisville Mega Cavern*.

(a) *Louisville Mega Cavern*



(b) Agha *et al.* (2021).



(c) DARPA SubT Challenge.

Fonte: Tranzatto *et al.* (2022).

Neste trabalho serão apresentados dois algoritmos de rastreamento de fontes de gás, um que utiliza informações de leitura e posição do robô para o rastreamento e o outro que utiliza mapas (GDM) para o rastreamento de fonte de dióxido de carbono (CO_2), tendo uma plataforma robótica móvel para explorar o ambiente.

1.2. Objetivo

O objetivo geral deste trabalho é o desenvolvimento de um método baseado em aprendizado por reforço para robôs móveis rastrearem fontes de gás.

1.2.1. Objetivos Específicos

- Estudar, propor, desenvolver modelos de dispersão de gás.
- Estudar, propor, desenvolver simuladores de robótica possíveis de adaptação para o problema.
- Desenvolver um ambiente de simulação capaz de representar bem o problema.
- Estudar, propor técnicas para mapeamento de gases.
- Utilizar/adaptar técnicas para mapeamento de gases.
- Desenvolver e avaliar modelos de inteligência artificial, baseados em aprendizado por reforço que seja capaz de rastrear fontes de gás em simulação.

1.3. Contribuições

A principal contribuição deste trabalho é a proposta e desenvolvimento de um modelo de aprendizado por reforço para robôs localizar fontes de gás. Além dessa contribuição, outras contribuições técnicas e científicas foram obtidas, e foram descritas nos documentos abaixo:

- *Marques, M.; Sathler, M. S.; Campos, S. P.; Santos, L. F.; Serrantola, W.; Pessin, G.; Lopes, B. E.; Falqueto, P.* . **Prognóstico de falha em correia transportadora: proposta e avaliação de dispositivos (Hardware).** 2021.
- *Marques, M.; Sathler, M. S.; Campos, S. P.; Santos, L. F.; Serrantola, W.; Pessin, G.; Lopes, B. E.; Falqueto, P.*. **Prognóstico de falha em correia transportadora: investigação em simulação e testes em correia de pequena escala.** 2021.
- *Freitas, G.; Rezende, A. M. C; Miranda, V.; Cruz Júnior, G.; Silva, R.; ARAÚJO, A.; Fany, C.; Marques, D.; Coutinho, D.; Matos, L. V. C. ; Leão, L.G.; Azpurua, H.; Domingues, J. D.; Barros, L. G. D.; Cid, A. L. M.; Sousa, F. L. M.; Torres, M. C. D.; Coelho, M.*

N.; Sathler, M. S.; França, P. H. T.; Teotonio, R.; Pessin, G. **Quarto relatório técnico de acompanhamento do projeto Dispositivo Robótico de Inspeção de Ambientes Restritos e Confinados.** 2021.

- *Freitas, G.; Rezende, A. M. C; Miranda, V.; Cruz Júnior, G.; Silva, R.; ARAÚJO, A.; Fany, C.; Marques, D.; Coutinho, D.; Matos, L. V. C. ; Leão, L.G.; Azpurua, H.; Domingues, J. D.; Barros, L. G. D.; Cid, A. L. M.; Sousa, F. L. M.; Torres, M. C. D.; Coelho, M. N.; Sathler, M. S.; França, P. H. T.; Teotônio, R.; Pessin, G.*. **Dispositivo robótico para inspeção de ambientes restritos e confinados.** 2021.

Adicional aos relatórios técnicos, no decorrer do projeto também foi publicado o seguintes trabalhos em congresso e depósitos de patente:

- *Sathler, M. S.; Cid, A. M.; Oliveira, G. O.; Freitas, G.; Pessin, G.; Azpurua, H.* **Localização de Fontes de Gás Usando Aprendizagem por Reforço com Robôs Móveis na Indústria da Mineração.** 2021, Simpósio Brasileiro de Automação Inteligente (SBAI).

- A. Cid; M. Sathler, M. Delunardo; J. Domingues; H. Azpúrua; G. Pessin; G. Freitas. **Path planning for ground robots based on radio signal strength.** 2022, Latin American Robotics Symposium (LARS), 2022 Brazilian Symposium on Robotics (SBR) and 2022 Workshop on Robotics in Education (WRE).

- *Pessin, G.; Marques, M.; Sathler, M. S.; Coelho, B. N.* **Dispositivo ejetor, veículo aéreo não-tripulado, e,método para ejeção de emergência.** 2021, Brasil.

  Patente: Privilégio de Inovação. Número do registro: BR10202101400, título: ”DISPOSITIVO EJETOR, VEÍCULO AÉREO NÃO-TRIPULADO, E,MÉTODO PARA EJEÇÃO DE EMERGÊNCIA”, Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial. Depósito: 15/07/2021. Instituição(ões) financiadora(s): Vale SA, UFOP.

- *Pessin, G.; Marques, M.; Sathler, M.; Dias, J.* **Sistema e Método Implementado por Computador de Monitoramento para Detecção e Estimação da Posição de Desalinhamento Lateral de Correrias Transportadoras, e Meio Legível Por Computador.** 2023, Brasil.

  Patente: Privilégio de Inovação. Número do registro: BR102023010884, título: ”SISTEMA E MÉTODO IMPLEMENTADO POR COMPUTADOR DE MONITORAMENTO PARA DETECÇÃO E ESTIMAÇÃO DE POSIÇÃO DE DESALINHAMENTO LATERAL DE CORREIAS TRANSPORTADORAS, E MEIO LEGÍVEL POR COMPUTADOR”, Instituição de registro: INPI - Instituto Nacional da Propriedade Industrial. Depósito: 02/05/2023. Instituição(ões) financiadora(s): Vale SA, UFOP.

1.4. Local de Realização

Este projeto foi desenvolvido no Laboratório de Robótica e Controle (ITVRoC) do Instituto Tecnológico Vale (ITV), em Ouro Preto, dentro do contexto do projeto EspeleoRobô.

1.5. Organização

O restante do trabalho tem a seguinte organização: No Capítulo 2 é apresentado o compilado de revisão teórica e trabalhos relacionados. O Capítulo 3 abordará o desvio de obstáculos em conjunto rastreamento de fontes de gás em plumas não turbulentas, no Capítulo 4 será apresentado o rastreamento de fontes de gases em plumas turbulentas em conjunto com o desvio de obstáculos, duas maneiras de treinamento de agentes inteligentes por aprendizado por reforço, seu desenvolvimento e resultados. Por fim, o Capítulo 5 conclui o trabalho e apresenta os próximos passos que podem ser seguidos nesta linha de pesquisa.

2. Referencial teórico e fundamentação científica

Este capítulo apresentará o embasamento teórico para a resolução do problema proposto e os trabalhos relacionados.

2.1. Localização de Fontes de Gás (GSL)

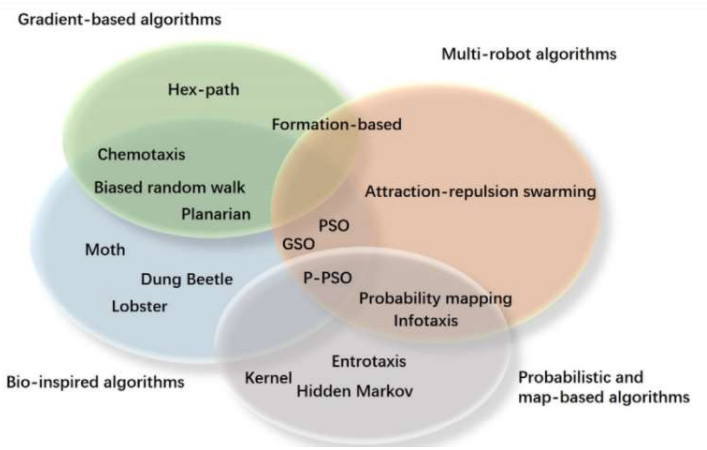
Durante as últimas três décadas, a localização de fontes de gás se tornou um campo de pesquisa popular, pois possui diversas potenciais aplicações, como a detecção de substâncias perigosas ou ilegais, localizar vazamentos em tubos de gás e assistência em missões de salvamento.

Existem diversos algoritmos sendo propostos atualmente. Chen e Huang (2018) apresentam um diagrama de Venn-Euler que ilustra as áreas de estudo, conforme mostrado na Figura 2.1.

Esses algoritmos podem ser categorizados em quatro grupos distintos, proporcionando uma visão abrangente e estruturada do campo de estudo.

- Baseados em gradiente;
- Bioinspirados;
- Multirrobô;
- Probabilísticos.

Figura 2.1: Diagrama de Venn-Euler das categorias dos algoritmos.



Fonte: Chen e Huang (2018).

As técnicas puramente baseadas em gradiente (DHARIWAL *et al.*, 2004) foram as primeiras a serem implementadas e se baseiam em deslocar o robô no sentido do gradiente de concentração. Essa técnica apresenta bom desempenho em ambientes de distribuição contínua, pois em caso de pequenas descontinuidades na dispersão da pluma esta técnica já perde efetividade. As estratégias bioinspiradas têm como fundamento, reproduzir o comportamento de animais que utilizam algum tipo de pluma de odor para busca de alimento ou até para reprodução (DAVIES *et al.*, 2015).

Tanto as técnicas baseadas em gradiente e bioinspiradas têm vantagem de possuir baixo custo computacional, no entanto, elas são limitadas em relação aos diferentes tipos de plumas e não levam em consideração o ambiente ao redor, sendo difícil a integração com algoritmos de navegação que acabam atrapalhando o próprio algoritmo de busca de fonte de gás. Os algoritmos de *Swarm* ou multirrobô utilizam de compartilhamento de informações, técnicas de exploração e alguma métrica para buscar a fonte, em Duisterhof *et al.* (2021) é apresentado uma estratégia de *Swarm* utilizando drones e o *Particle Swarm Optimization* (PSO).

Já as probabilísticas possuem maior custo computacional como desvantagem, e como vantagem é mais fácil a integração com algoritmos de navegação e consegue compreender melhor os diversos tipos de plumas.

As estratégias probabilísticas, que são fundamentadas em mapas, modelam a localização da fonte do odor. Elas representam essa localização como uma distribuição de probabilidade, derivada das leituras obtidas pelo sensor de gás.

Nos últimos anos os estudos na área de algoritmos probabilísticos têm ganhado força, pelo aumento do poder computacional técnicas antes que eram inviáveis computacionalmente. Hoje são possíveis de serem processadas por aparelhos móveis, assim, as abordagens baseadas em aprendizado de máquina atualmente apresenta melhor desempenho em muitos casos, com isso, ganhou espaço.

Contudo, até o momento existem poucos trabalhos baseados em aprendizado por reforço para resolver o GSL. Niu *et al.* (2017) e Hu *et al.* (2019) propuseram estratégias de rastreamento de pluma turbulenta estática para veículos aquáticos baseados em aprendizado profundo por reforço, onde as técnicas propostas são comparadas com estratégias convencionais. Singh *et al.* (2020) apresentam uma rede baseada em aprendizado profundo por reforço para resolver o problema de rastreamento de fonte de gás para plumas no ambiente aéreo. Adicionalmente, é mostrada a variação do aprendizado, onde é fornecido um vetor de estados anteriores como entrada do modelo de aprendizado, representando a ideia de memória.

No restante desta subseção serão apresentados alguns temas importantes para o entendimento do problema de localização de fontes de gás.

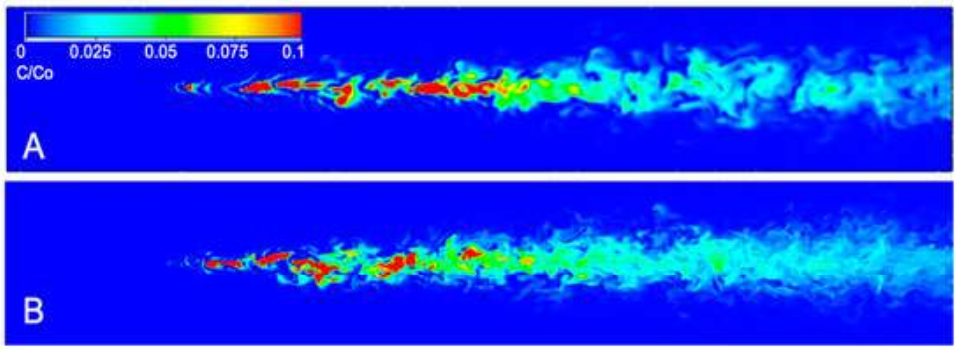
2.1.1. Pluma de gás

Uma pluma é uma coluna de fluido em movimento dentro de outro fluido. No caso das plumas gases nocivos geralmente estão em movimento dentro do ar atmosférico, esses fluidos podem ser movimentados por difusão, pressão e momento. As plumas de gás/odor têm como classificações turbulentas e não turbulentas. O parâmetro de classificação é o número de Reynolds, plumas não turbulentas ocorrem quando as forças viscosas são dominantes (fluxo lento, baixo número de Reynolds).

Estas forças são suficientes para manter todas as partículas de fluido alinhadas, resultando em um fluxo laminar, geralmente ocorrem em locais fechados e com pouco fluxo de vento como

cavernas. Já em plumas turbulentas (Figura 2.2) as forças inerciais atuam sobre as forças viscosas (fluxo rápido, número de Reynolds maior), esse comportamento é comum em ambientes abertos, com grande influência de ventos.

Figura 2.2: Dissipação de uma pluma de gás.



Fonte: Leathers *et al.* (2020).

2.1.2. Simulação de robôs

É importante ressaltar a dificuldade de realizar os experimentos para localização de fontes de gás, pois é necessário um ambiente grande, preparado, alto custo financeiro e o risco de trabalhar com gases perigosos. Com isso, as ferramentas computacionais são de grande valia.

A grande maioria dos trabalhos de GSL utilizam algum simulador de pluma de gás para o desenvolvimento dos algoritmos, porém, vale relembrar que ainda não existe um modelo que simule perfeitamente todos os movimentos das partículas, isto implica que após o desenvolvimento do algoritmo em simulação, é importante a validação em ambiente real.

Os ambientes simulados têm como função básica de reproduzir com fidelidade o ambiente e o robô com seus determinados atributos, sendo uma ferramenta indispensável para prototipagem de sistemas robóticos. Um simulador possibilita testar ideias em ambientes seguros e controlados, evitando danos à estrutura do robô. Escolher o simulador apropriado é importante, pois diferentes ambientes de simulação oferecem desempenhos, detalhamentos e caraterísticas variadas.

A utilização destes ambientes é de grande auxílio para o desenvolvimento de algoritmos que são capazes de localizar fontes de gás. Segundo Cid *et al.* (2020), alguns dos simuladores mais utilizados na robótica são:

- O **Gazebo** é um simulador desenvolvido pela *Open Robotics* para ser compatível com Robot Operating System (ROS), de código aberto e gratuito (GAZEBO, 2019). O simulador de dinâmica padrão é o ODE, e simuladores opcionais podem ser adicionados em forma de *plugins*;

- O **Webots** é um simulador de robótica, lançado pela empresa suíça (CYBERBOTICS, 2020), que visa a facilidade no uso. Este é um simulador gratuito de código aberto que

também pode ser controlado externamente usando a API ou qualquer rede TCP/IP padrão, tem suporte ao ROS por padrão e possui o motor de física baseado no ODE;

- *Autonomous Robots Go Swarming* (**ARGoS**) foi desenvolvido no projeto *Swarmanoid* especificamente para simulação de enxames e possui parte gráfica simples com desempenho aprimorado (PINCIROLI *et al.*, 2012). Com arquitetura baseada em *plugins*, é possível realizar a troca do motor de física e adicionar sensores via estes módulos;

- O **CoppeliaSim**, sucessor do VREP, é um simulador de robótica muito utilizado para prototipagem, desenvolvimento e teste de algoritmos, verificação de segurança e para fins educacionais na área da robótica. Ele é um software de código fechado comercial que pode ser obtido gratuitamente em sua versão educacional (COPPELIAROBOTICS, 2019). Seu ambiente de desenvolvimento integrado é baseado em arquitetura de controle distribuído, permitindo que os objetos e modelos sejam controlados de forma individual via *scripts*. Este simulador é bem reconhecido por ser bem documentado, ter um time de suporte presente e um eficiente manual de instruções *offline*;

- O **Player** e **Stage** (VAUGHAN, 2008) é um simulador mais antigo e limitado, porém, fornece boas funcionalidades.

2.1.3. Simuladores de Dispersão de Gás

Os simuladores de dispersão de gás, como o nome sugere, têm como objetivo reproduzir dispersões de gases de maneira realista. No entanto, existem alguns desafios. O primeiro é que não há uma formulação que descreva exatamente a dispersão dos gases, o que significa que todos os simuladores representam apenas uma aproximação da realidade. O segundo desafio é o tempo necessário para calcular a dispersão do gás. A geração de apenas alguns minutos de simulação pode levar cerca de uma hora .

Para a segunda parte do trabalho, será utilizado o simulador GADEN Monroy *et al.* (2017a). Este simulador, que opera no ROS, é capaz de processar modelos de dispersão de diferentes tipos de gases. Ele realiza o pré-processamento da simulação, salvando os dados obtidos em arquivos binários que só precisam ser lidos ao executar a simulação. Além disso, o GADEN gera modelos em ambientes tridimensionais que levam em consideração obstáculos, o que o diferencia de outros simuladores.

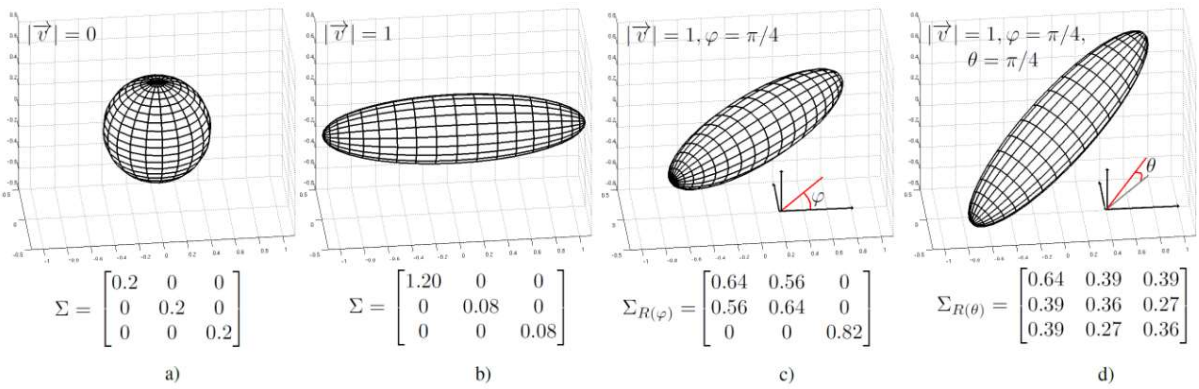
No entanto, devido à sua abrangência, o GADEN apresenta um alto nível de complexidade para a geração de novos ambientes. Isso requer a execução de uma série de atividades: (i) criação de um modelo *Computer Aided Design* (CAD) dos ambientes; (ii) simulação dos campos vetoriais (SimScale); (iii) geração de um arquivo com campos vetoriais; e (iv) simulação e criação dos arquivos binários (GADEN).

2.2. Técnicas de Mapeamento

As técnicas de mapeamento de gás ou *Gas Distribution Model* (GDM) têm como objetivo gerar mapas representativos da distribuição de gás, essas técnicas necessitam de algumas informações, como leituras de sensor de gás, posição onde a leitura foi realizada e em alguns casos específicos utilizam o mapa do ambiente e leituras de sensor anemômetro.

Com base nessas leituras, existem várias abordagens para processar e analisar esses dados. Em Reggente e Lilienthal (2010) é apresentado um algoritmo capaz de representar plumas turbulentas em ambientes tridimensionais, esse algoritmo se baseia em utilizar convolução de um kernel tridimensional que tem seus pesos ajustados pela leitura do anemômetro (Figura 2.3)

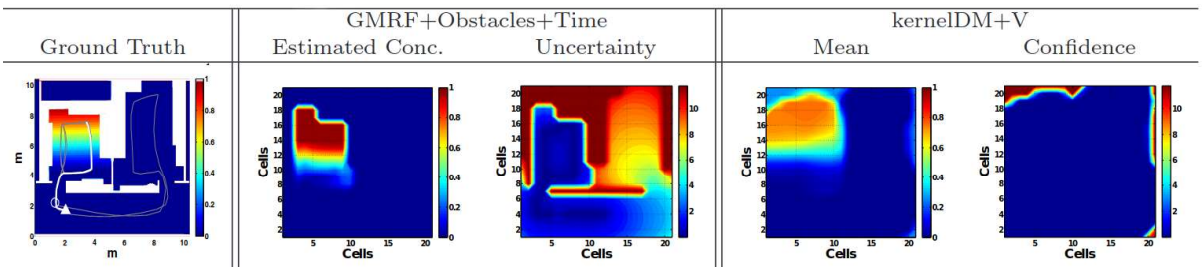
Figura 2.3: Modificação do kernel baseado em leituras de vento. a) leitura do anemômetro = 0; b) c) d) leituras do anemômetro > 0



Fonte: Reggente e Lilienthal (2010).

Monroy *et al.* (2015) apresentam uma técnica de mapeamento baseado em *Gaussian Markov Random Field* (GMRF) em que obstáculos afetam a distribuição bidimensional do gás. Isso é bom por representar mais fielmente o ambiente como apresentado na Figura 2.4, porém, é mais custoso computacionalmente.

Figura 2.4: Comparação dos métodos GMRF e KernelDM+V

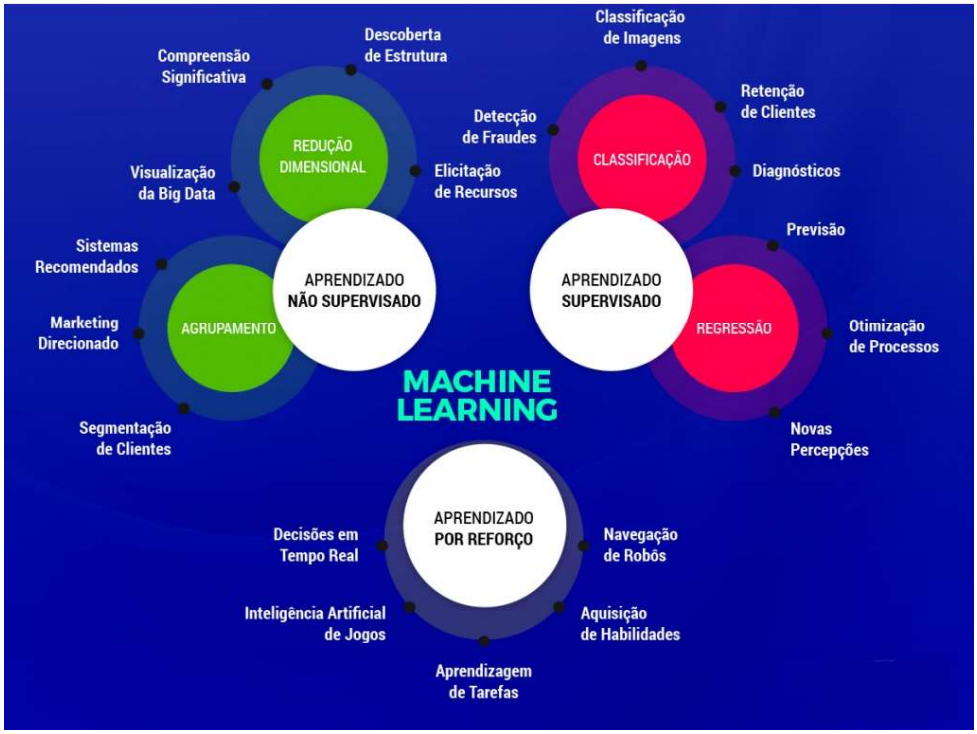


Fonte: Monroy *et al.* (2015).

2.3. Aprendizado de Máquina

O aprendizado de máquina ou *machine learning* é subdividido em três subcategorias (Figura 2.5) que são: (i) aprendizado supervisionado; (ii) aprendizado não-supervisionado; e (iii) aprendizado por reforço (MURPHY, 2012).

Figura 2.5: Diagrama de Venn-Euler das áreas do aprendizado de máquina.



Fonte: programaria (2020).

A categoria do aprendizado supervisionado tem como fundamento criar um modelo preciso que consiga dividir os rótulos das classes por características dos dados fornecidos a partir de uma base de dados fornecida já rotulada (KOTSIANTIS *et al.*, 2006). O aprendizado não supervisionado é um subgrupo onde seu propósito é agrupar conjunto de dados que não possuem rótulos ou classes pré-definidas(BARLOW, 1989). Os quais buscam encontrar padrões ou similaridades nos dados sem intervenção humana. Este trabalho é fundamentado em aprendizado por reforço, que será detalhado na Seção 2.3.1, no entanto, também serão utilizadas técnicas que são conhecidas pela sua utilização no aprendizado supervisionado, como a utilização de redes convolucionais para extração de *features*.

2.3.1. Aprendizado por Reforço

O aprendizado por reforço é um dos três principais subgrupos de aprendizado de máquina (Figura 2.5). Esse campo é bem conhecido por conseguir performar melhor do que o ser humano em alguns jogos como os do Atari (WANG *et al.*, 2016), alphaGO (SILVER *et al.*, 2017), xadrez (SILVER *et al.*, 2018), entre outros, porém, também tem sido muito utilizado na robótica para

encontrar uma solução viável para problemas tradicionalmente intratáveis dado o grande número de parâmetros e variáveis envolvidas.

Os problemas da robótica geralmente são bastante complexos de modelar matematicamente e os algoritmos de RL se tornam uma boa alternativa. Haarnoja *et al.* (2019) apresentam aprendizado profundo por reforço (DRL) para que um robô quadrupede aprenda a andar. Em *a deep reinforcement learning approach for dynamically stable inverse kinematics of humanoid robot* foi utilizado o DRL a fim de manter o equilíbrio de um robô, após treinamento foi capaz de resolver a cinemática inversa com 90% de precisão para parte superior do corpo altamente articulada de um modelo humanoide com 27 graus de liberdade (DoF).

A ideia do aprendizado por reforço (RL) é ensinar agente como realizar ações em determinadas situações (ambiente), isto de modo a maximizar uma recompensa numérica (Figura 2.6). Ao agente não é dito quais ações tomar, ao invés disso ele deve descobrir por meio de iterações, quais ações retornam recompensas maiores, se tornando um problema de otimização que busca maximizar as recompensas (R) se baseando no estado atual (S'), estado anterior (S), ação (A) (S, A, R, S'). Em casos mais complexos essas ações podem não só afetar a recompensa imediata, mas também as situações e recompensas seguintes, gerando um efeito cascata. Essas duas características: (i) Busca por tentativa e erro (*trial-and-error search*) e (ii) Atraso de recompensa (*delayed reward*), são as mais importantes características distintivas da aprendizagem de reforço de entre outros tipos de aprendizado.

Q-learning e Deep Q-learning

O *Q-learning* Watkins e Dayan (1992) é um algoritmo que busca encontrar a melhor ação para o estado em que se encontra. Tal como os primeiros algoritmos de RL o Q-learning é baseado na tabela Q, tabela essa que tem como princípio mapear todos as ações por estados (AxS). Conforme o agente interage com o ambiente, é atribuído valor de qualidade a tabela, ou seja, esse valor demonstra a qualidade da ação, em que ações com maior valor de qualidade resultam em um comportamento mais próximo do esperado para o agente.

Os algoritmos de RL podem ser divididos em dois grupos, os baseados em valor e os baseados e política. Essa divisão se dá pela maneira que o algoritmo busca aprender, o *Q-learning* é um algoritmo baseado em valor que utilizada a função de Bellman (Equação 2.1), que é uma função de valor, para preencher os valores de qualidade. Já os baseados em política (KIMURA *et al.*, 1995; SCHULMAN *et al.*, 2017) ao invés de tentar estimas os valores, tem como princípio estimar funções de valores, sempre se baseando em políticas gananciosas:

$$q^{new}(s,a) = (1-\alpha)q(s,a) + \alpha(R_{t+1} + \gamma max_{a'} q(s',a')), \tag{2.1}$$

onde:

- $q(s',a')$: valor de qualidade atual;

- $q(s,a)$: valor de qualidade em um período de tempo atrás;
- α: taxa de aprendizado;
- γ: fator de desconto.

Agentes baseados no Q-learning apresentam um bom desempenho e conseguem resolver uma ampla variedade de problemas. No entanto, o Q-learning também apresenta alguns desafios. O principal problema é a tabela Q, que armazena todos os estados e ações. Quando se trabalha em ambientes contínuos, o tamanho da tabela pode se tornar extremamente grande. Isso pode levar a uma inviabilidade computacional, devido à quantidade excessiva de recursos necessários para processar e armazenar esses dados.

A partir disso, o Deep-Q-learning soluciona esse problema com a substituição da tabela Q por uma rede neural. A ideia é que partir das iterações com o ambiente realizar o *fit* da rede, para que ela consiga representar bem o ambiente e assim, retornar os valores de ação de cada estado o que é o papel da tabela Q (função de aproximação da tabela Q), no entanto, é computacionalmente mais eficiente, pois foi substituída uma matriz extremamente grande por uma rede neural, que são basicamente funções. O restante é igual ao valor da tabela Q será enviado para a função de Bellman e esse valor serve de ajuste para a rede.

O DQN resolve o problema computacional da tabela Q, mas gera outros problemas no processo de treinamento que devem ser solucionados para que ocorra um aprendizado de convirja e mais estável (THRUN e SCHWARTZ, 1993). Isso se deve ao fato do *Q-learning* aprender com cada uma das tuplas de experiência $(Estado_Anterior, Acao, Recompensa, Estado)$ em ordem sequencial, com isso pode ser influenciado pelos efeitos dessa correlação, e também se deve a superestimação sistemática dos valores. No entanto, é possível estabilizar o treinamento com duas técnicas chaves:

- **Experience Replay:** Possibilita a reutilização de experiências anteriores para o aprendizado, pois sem essa técnica, após a utilização da experiência ela é descartada, já no replay buffer não. Ele armazena as experiências passadas em um buffer e as misturam dentro deste, quebrando a correlação entre experiências e melhorando dois casos do aprendizado por reforço: (i) estabilização do aprendizado; (ii) reutilização de experiências passadas;
- **Target network:** Quando utilizadas as redes neurais para representar a função de valor, diferente do *Q-learning* que se atualiza o exato valor de estado, no DQN se ajusta os parâmetros de toda a rede que pode afetar no valor de ação do próximo estado. A *target network* propõe resolver esse problema com outra DQN de mesma configuração, essa rede *target* se atualizará com uma frequência menor, a ideia é fixar os valores de ação dos próximos estados por um determinado período de tempo, estabilizando o aprendizado.

  Existem dois métodos de atualização da rede. O *full update* que atualiza direto toda a rede e o *soft update* em que a rede de destino é atualizada com maior frequência, mas

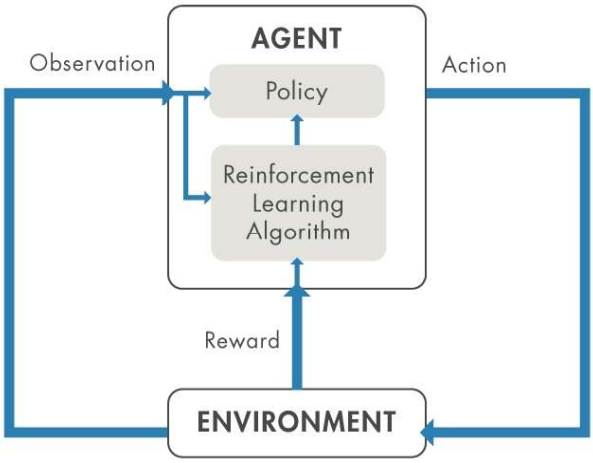
em pequenas partes e em escala menor, assim, a rede *target* irá se mover lentamente, sua atualização deve ser contínua para que o efeito seja perceptível. O valor de τ é usado na fórmula de *soft update* como mostra a equação 2.2, onde θ são os parâmetros da rede destino, θ^- os parâmetros da rede target e τ é uma constante de atualização.

$$\theta^- = (\theta * \tau) + (\theta^- * (1 - \tau)). \tag{2.2}$$

Ferramentas para Treinamento do Aprendizado por Reforço

Como apresentado na Figura 2.6 o aprendizado por reforço tem seu padrão de funcionamento e necessidade de muitas iterações para conseguir convergir a melhor política. Dito isso, existem ferramentas que têm como objetivo facilitar o desenvolvimento de algoritmos e técnicas de aprendizado, provendo um *framework* base e padronizado pelo qual começar os treinamentos e onde outros algoritmos e implementações já validados podem ser adaptados.

Figura 2.6: Sistema de iterações no aprendizado por reforço.



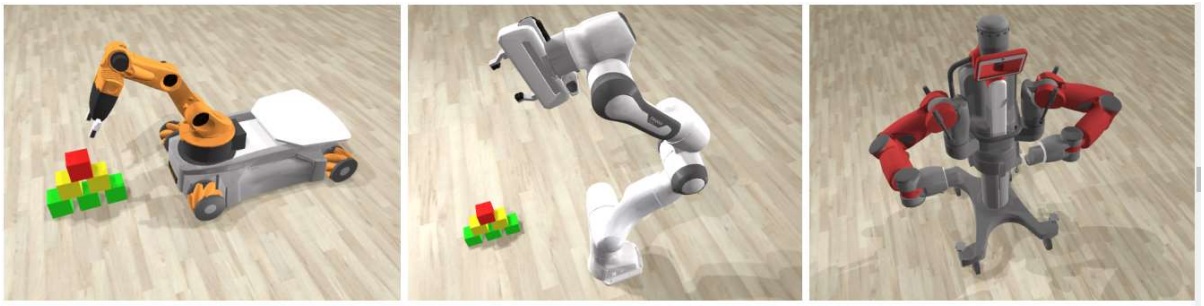
Fonte: Mathworks (2021).

OpenAI Gym é um kit de ferramentas para pesquisa em aprendizado de máquina. Este kit de ferramentas oferece diversos ambientes e agentes preparados para implementações do aprendizado por reforço. Segundo Brockman *et al.* (2016) os algoritmos de RL necessitavam de um ambiente de simulação comum, para que as pessoas possam compartilhar, comparar e analisar os resultados das diferentes técnicas de RL nestes ambientes. O intuito é desenvolver uma base para entender quais as melhores técnicas para determinados tipos de ambientes e assim, acelerar o desenvolvimento da área.

PyRep é um kit de ferramentas para pesquisa de aprendizagem de robôs, construído para o CoppeliaSim / V-REP. Foi construído de forma a ser otimizado com o Coppeliasim com aumento

de velocidade de 10000 vezes em relação com o Python Remote API anterior, propôs um novo mecanismo de renderização e possui uma API simples para controle do robô e manipulação de cena (JAMES *et al.*, 2019). Nesse pacote já vêm alguns exemplos (Figura 2.7) e maneiras de integrar novos robôs a sistemas de aprendizado de máquina.

Figura 2.7: Ambientes de exemplo PyRep.



Fonte: James *et al.* (2019).

2.3.2. Redes Convolucionais

As redes convolucionais, também conhecidas como redes neurais convolucionais (CNNs), são uma classe especializada de algoritmos de aprendizado profundo amplamente utilizados em tarefas de processamento de imagens e reconhecimento de padrões. Eles são particularmente eficazes na extração de recursos de dados visuais.

Os dados da imagem são representados como uma grade bidimensional de *pixels*, que pode ser monocromática ou colorida. Portanto, cada pixel corresponde a um ou mais valores numéricos. Em *Multi Layer Perceptrons* (MLP), essas estruturas são tratadas como vetores numéricos, achatando a imagem independentemente da relação espacial entre os *pixels*. Esta abordagem não é satisfatória para o processamento de imagens, com isso (LECUN *et al.*, 1995) em 1995 apresentou as redes convolucionais. A ideia básica por trás de uma rede convolucional é aplicar uma operação de convolução aos dados de entrada. A convolução é um processo matemático que envolve a combinação de uma matriz de filtros com dados de entrada, resultando em uma representação transformada. Essa operação permite que as redes convolucionais capturem com eficiência recursos locais e invariantes, como arestas, textura e forma.

Uma das principais vantagens das redes convolucionais é a capacidade de compartilhar parâmetros entre as camadas. Isso significa que o mesmo filtro é aplicado a diferentes partes da imagem, reduzindo a quantidade de parâmetros necessários e permitindo que a rede aprenda com eficiência os recursos comuns em diferentes partes da imagem.

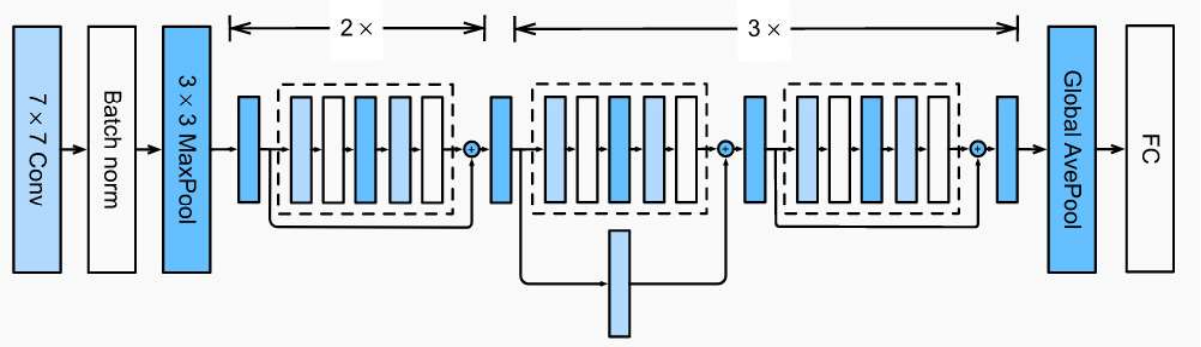
Uma rede convolucional consiste em várias camadas, incluindo camadas convolucionais, camadas de agrupamento (*pooling*) e camadas totalmente conectadas ou uma camada de *Max Average Pooling*. As camadas convolucionais são responsáveis pela extração de recursos, enquanto as camadas de agrupamento reduzem a dimensionalidade e ajudam a preservar recursos

importantes. As camadas totalmente conectadas e a camada de *Max Average Pooling* são responsáveis pela classificação final ou regressão com base nas representações aprendidas.

As redes convolucionais tiveram um grande impacto em muitos campos, incluindo visão computacional, reconhecimento de fala, processamento de linguagem natural e muito mais. Eles têm sido usados para tarefas desafiadoras, como detecção de objetos, reconhecimento facial, segmentação de imagens e até mesmo criação de obras de arte geradas por computador.

O desafio da ImageNet (2021), também conhecido como ILSVRC (ImageNet Large Scale Visual Recognition Challenge) ocorreu entre 2010 e 2017. A ideia principal por trás desse desafio era usar um grande conjunto de dados chamado ImageNet, que continha milhões de imagens marcadas com vários objetos em mais de mil categorias. Com o objetivo de desenvolver algoritmos que pudessem classificar corretamente as imagens nas categorias apropriadas a ImageNet foi responsável por impulsionar o desenvolvimento no campo de visão computacional, gerando arquiteturas famosas (Figura 2.8) na literatura, como: AlexNet (KRIZHEVSKY *et al.*, 2012), NiN (LIN *et al.*, 2013), VGG (SIMONYAN e ZISSERMAN, 2014), GoogLetNet (SZEGEDY *et al.*, 2015), ResNet (HE *et al.*, 2016), DenseNet (HUANG *et al.*, 2018).

Figura 2.8: Arquitetura da ResNet-18



Fonte: Zhang *et al.* (2023).

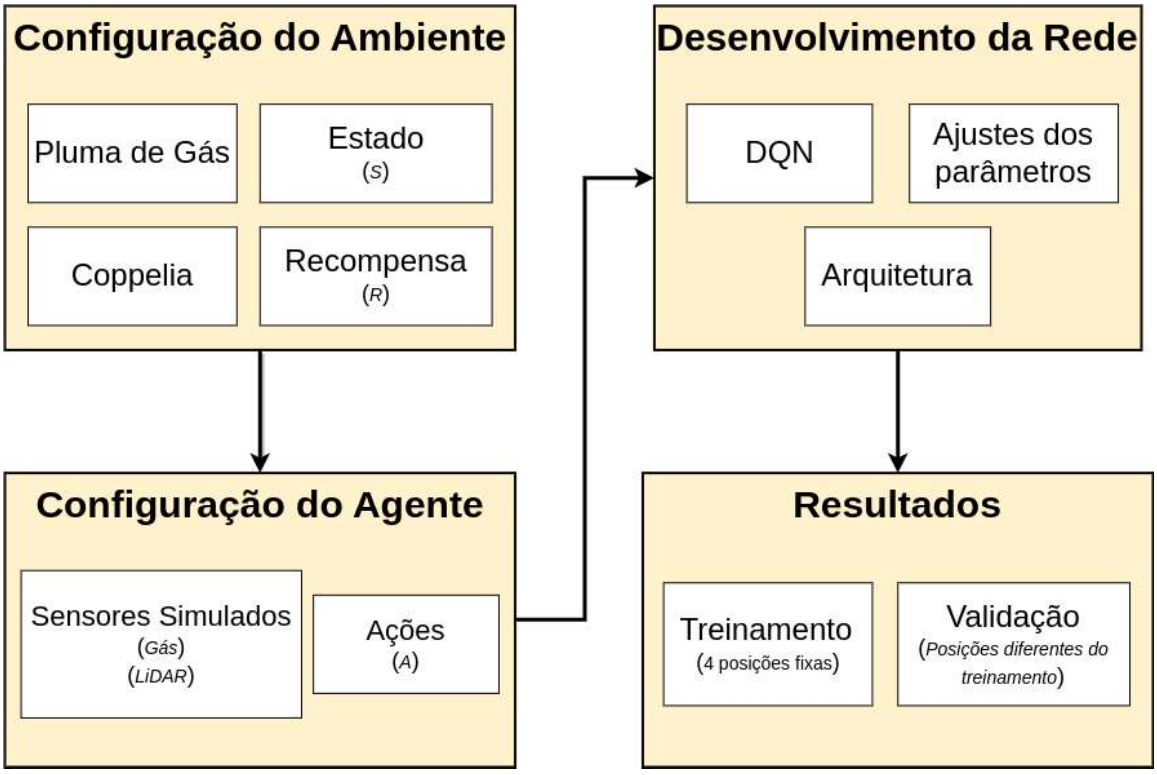
3. Rastreamento de fonte de gás em plumas não turbulentas

Este capitulo apresenta a continuação do trabalho iniciado em Sathler *et al.* (2021), que utilizou aprendizado por reforço para rastrear uma fonte de gás não turbulenta. Com objetivo de deixar o sistema cada vez mais realístico, adicionamos obstáculos. Dessa forma, o método de aprendizado por reforço deve ser capaz de realizar desvio de obstáculos.

Para esta etapa é utilizado o simulador Coppeliasim em conjunto com o *plugin* de aprendizado por reforço no PyRep (JAMES *et al.*, 2019). Este *plugin* facilita o treinamento, pois com ele é possível reiniciar o ambiente, remover a renderização gráfica. Consequentemente é possível acelerar bastante a simulação e treinamento.

A Figura 3.1 apresenta o fluxo das atividades, configurações e desenvolvimento/ajuste da rede necessários para a realização do treinamento e validação do algoritmo de aprendizado por reforço.

Figura 3.1: Fluxo de execução das atividades

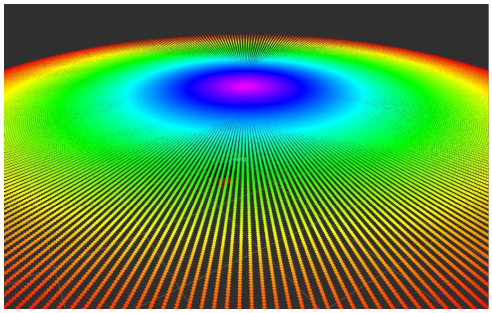


Fonte: O Autor.

3.1. Ambiente e Arquitetura

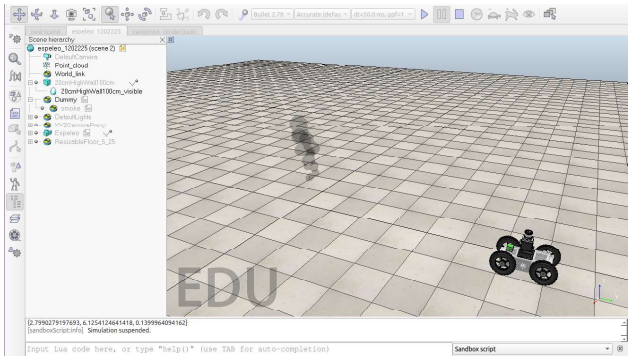
Foi utilizado um ambiente que possui obstáculos, uma pluma não turbulenta e o robô. A pluma de gás simulada tem um decaimento linear conforme apresentado na Figura 3.2, onde em roxo é a maior concentração de gás (400) e vermelho a menor concentração (1). A plataforma utilizada para a simulação foi o EspeleoRobô (AZPÚRUA *et al.*, 2021; CID *et al.*, 2020).

Figura 3.2: Pluma não turbulenta simulada. (Roxo ≈ 400, Vermelho ≈ 5)

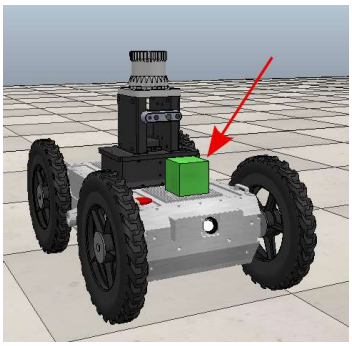


Fonte: Sathler *et al.* (2021).

Figura 3.3: Ambiente e EspeleoRobô simulados no Coppeliasim.



(a) Ambiente com pluma de gás e robô.



(b) Plataforma robótica EspeleoRobô.

A arquitetura utilizada é baseada no trabalho de Sathler *et al.* (2021), onde foi adicionado a entrada de leituras do laser de distância (Figura 3.4).

3.1.1. Estado

Em um ambiente de aprendizado por reforço, a cada passo dado pelo agente o ambiente deve gerar o próximo estado, calcular a recompensa e verificar se a condição de encerramento do episódio aconteceu. Nesse ambiente os estados definidos foram:

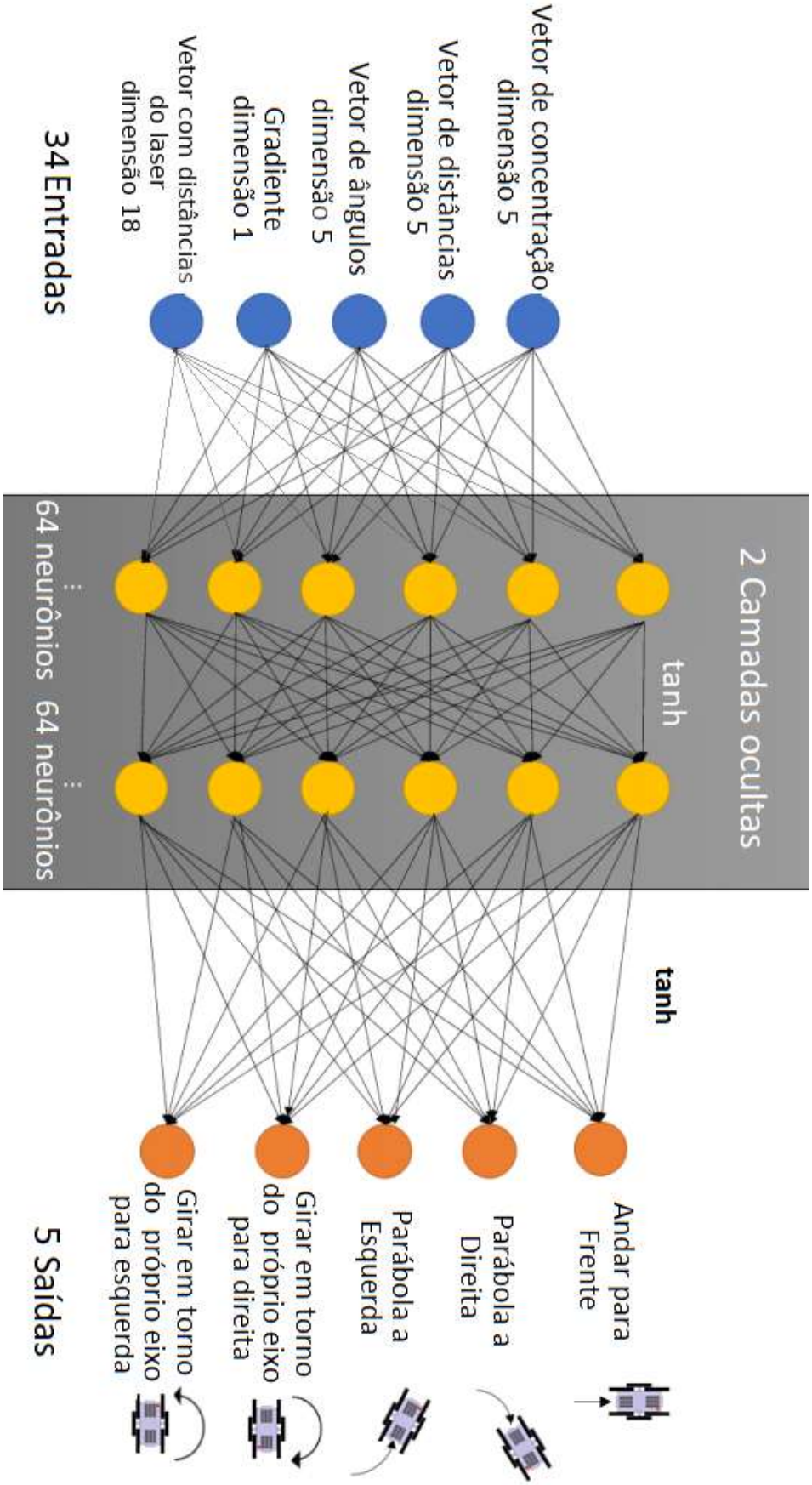
- Concentração de gás.
- Distância deslocada entre a posição atual para posterior do movimento. Representado na equação 3.1, onde D, x, y, t consecutivamente correspondem a distância relativa, posição x referente ao ponto de referência do simulador, posição y correspondente ao ponto de referência do simulador e intervalo de tempo:

$$D(t) = \sqrt{(x(t) - x(t-1))^2 + (y(t) - y(t-1))^2} \quad \{D \in \mathbb{Q} \text{ — } 0 \leq D \leq 0.2\}. \tag{3.1}$$

- Ângulo entre a pose atual para posterior do movimento (Equação 3.2):

$$\theta(t) = yaw \qquad \{\theta \in \mathbb{Q} \text{ — } -2\pi \leq \theta \leq +2\pi\}. \tag{3.2}$$

Figura 3.4: Arquitetura da rede utilizada.



Fonte: O Autor.

- Gradiente de da concentração e gás (0 ou 1).
- Leituras de distância, a partir do sensor simulado *velodyne*, foram utilizados só os feixes planares e selecionado 1 leitura a cada 20 graus, isto se deve ao fato de evitar a sobrecarga da rede com muita informação para processar.

3.1.2. Sensor de gás simulado

O sensor de gás é representado pela caixa verde acoplada ao robô, conforme a Figura 3.3b. Para coleta da concentração de gás pelo sensor, foi desenvolvido um algoritmo de busca da seguinte maneira: Na pluma de odor radial, os pontos distribuídos no ambiente possuem listas de posição (x, y, z) e concentração ($\mathbb{P}$), relativos a um ponto fixo (objeto *Dummy*) denominado como "*world*".

Para saber se as partículas de gás se encontram dentro dos limites do sensor, primeiramente é utilizado um método que busca na lista com informações da posição (x, y, z) e a intensidade de cada partícula. Vale ressaltar que o sensor simulado tem formato cúbico com lados de 8 centímetros. Em seguida, as partículas são analisadas de forma que os pontos lidos pelo sensor são os que satisfazem as condições seguintes:

$$\phi_x(t) - \frac{T}{2} <= \psi_x(t) <= \phi_x(t) + \frac{T}{2}, \tag{3.3}$$

$$\phi_y(t) - \frac{T}{2} <= \psi_y(t) <= \phi_y(t) + \frac{T}{2}, \tag{3.4}$$

onde $\psi_{\{x|y\}}$ é a posição global da partícula química, $\phi_{\{x|y\}}$ é a posição global do sensor de gás no eixo x ou y no instante t, e T o tamanho de um lado do sensor.

Para o cálculo de concentração $\mathbb{C}$ das partículas $\mathbb{P}$ que se encontram dentro do sensor foi utilizada a fórmula:

$$\mathbb{C} = \frac{\sum_{n=1}^{p} \mathbb{P}_n}{|\mathbb{P}|}, \tag{3.5}$$

onde $|\mathbb{P}|$ é o número total de partículas dentro do sensor.

3.1.3. Recompensas

Para obtenção do comportamento desejado no RL é importante a definição adequada das recompensas (Equação 3.6) e o peso de cada uma. Com isso, a função de recompensa deve fornecer um *feedback* positivo em situações em que o robô está dentro da pluma, seguindo o gradiente de concentração e encontro da fonte, caso o robô saia da pluma, colida ou siga um gradiente negativo de concentração de gás o *feedback* deve ser negativo.

$$r(s,a) = \begin{cases} 1 & \text{Robô dentro da pluma} \\ -1 & \text{Para cada iteração do agente com o ambiente} \\ -100 & \text{10 iterações seguidas fora da pluma (Condição de parada)} \\ -100 & \text{Colisão com obstáculos (Condição de parada)} \\ 1000 & \text{Encontre a fonte (Condição de parada)} \end{cases} \tag{3.6}$$

3.1.4. Normalização

A normalização é importante para igualar o peso de cada dado de entrada na rede, pois, dados com faixas de valores muito diferentes dificultam os otimizadores e gera oscilação no aprendizado resultando em acréscimo de tempo para convergência. Assim foi adotado o padrão de normalização que é compreender todos os dados entre 0 e 1. As entradas da rede foram normalizadas conforme as equações:

- Equação 3.7: No caso da pluma radial, a leitura do sensor (L) pode variar de 0 a 400, assim, para normalizar (L_N) foi dividido o valor de leitura por 400:

$$L_N = \frac{L}{400}. \tag{3.7}$$

- Equação 3.8: A maior distância percorrida pelo robô em um único movimento chega perto de 0.2, por isto foi dividida a distância caminhada por 0.2. Para calcular a distância é utilizado a posição atual em x (x_n) e y (y_n) e a posição passada de x (x_{n-1}) e y (y_{n-1}):

$$D_N = \frac{\sqrt{(x_n - x_{n-1})^2 + (y_n - y_{n-1})^2}}{0.2}. \tag{3.8}$$

- Equação 3.9: O ângulo da pose do robô (α_t) está em radianos que compreende de -2π a $+2\pi$, com isso, a normalização (α_N) foi feita adicionado π e dividido por 2π:

$$\alpha_N = \frac{\alpha_t + \pi}{2 * \pi}. \tag{3.9}$$

- Equação 3.10: Foi definido que a distância máxima que o robô terá de informação é 2,5 metros (essa limitação foi realizada com intuito do robô não conseguir ter informação de todo o ambiente), a normalização é a divisão da distância retornada pelo sensor dividido pela distância máxima:

$$Leitura_Laser(n) = \frac{distancia(n)}{2,5}. \tag{3.10}$$

3.1.5. *Exploration* e *Exploitation*

Em RL para o agente aprender inicialmente tem que haver uma maneira de exploração inicial do ambiente, para a obtenção de informação e a partir dessas informações desenvolver uma base de informação que descreve melhor o ambiente. Em RL o comportamento em que o agente prioriza realizar ações que não são as melhores possíveis até o momento é definido como *exploration*, e esse comportamento é essencial para que o algoritmo de treinamento não fique preso em máximos locais que ocorre quando é utilizado apenas o *exploitation*, comportamento o qual faz com que o agente utilize sempre as melhores ações para os estados.

Existem algumas estratégias de *exploration* a mais comum e utilizada é o épsilon-greedy. A ideia do épsilon-greedy é que a cada ação que seja tomada exista uma chance de o agente escolher uma ação aleatória, isso é bem importante nos primeiros momentos do aprendizado, e menos interessando após o agente treinado, por isso é utilizada uma função de decaimento da chance de escolha de ações aleatórias.

Para este trabalho, tanto o treino 1 quanto o treino 2, o decaimento do épsilon-greedy será sempre de 0,99 por episódio (épsilon = épsilon * 0,99), e o valor inicial será alterado conforme necessidade e apresentado no restante do trabalho.

3.1.6. Função de Custo

No aprendizado de máquina, a função de perda é determinada como a diferença entre a saída real e a saída prevista do modelo para o exemplo de treinamento único, enquanto a média da função de perda para todo o exemplo de treinamento é denominada como a função de custo. Esta diferença calculada das funções de perda é denominada como o valor de erro. No caso, foi utilizado como função de custo o erro quadrático médio (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i')^2, \tag{3.11}$$

onde n é o número total de dados, y_i o valor real e y_i' o valor previsto.

3.1.7. Parâmetros

Além da rede neural profunda também é necessário o ajuste dos parâmetros de aprendizagem para que este aprendizado ocorra da melhor maneira possível. Os parâmetros utilizados nos treinamentos foram:

- Buffer Size = 1.000.000, é o tamanho to buffer para armazenar experiências, este valor foi definido com intuito de armazenar boa parte das experiencias, o episódio tem tamanho máximo de 150 experiências, dividindo 1.000.000 por 150 resulta em 6.600 episódios armazenados.

- Batch Size = 3.200, representa o tamanho do lote de experiências que serão utilizadas para o aprendizado.
- Gamma = 0,95, é o fator de desconto.
- Tau = 0,001, parâmetro de interpolação da função *soft update* para atualizar os parâmetros da rede target.
- Learning Rate = 0,0003, Taxa de aprendizado.
- Update_every = 1, Taxa de atualização, no caso atualiza a rede target a cada nova experiência.

3.2. Treinamento e Resultados

Os treinamentos foram executados em um computador com 16GB de memória RAM, processador Intel i5, placa gráfica NVIDIA Quadro P400 2GB, com sistema operacional Ubuntu 18.04 LTS.

Para o treinamento as posições iniciais (Figura 3.5) foram configuradas a partir de posições dentro da pluma, onde o robô aparece em diferentes orientações. Essa abordagem apenas apresenta o rastreio da pluma, não é apresentada a declaração da fonte, no algoritmo de RL é considerado que a fonte foi encontrada após a leitura do sensor ser maior que 360.

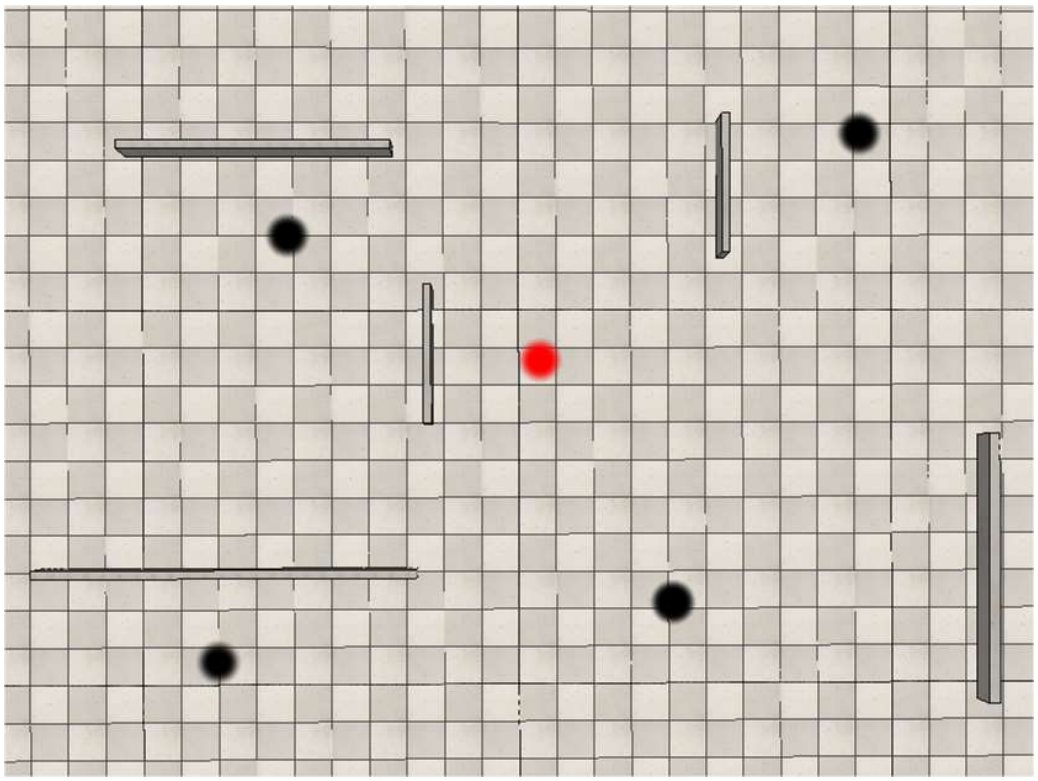
A partir dos parâmetros configurados, no treinamento 1 (Figura 3.6a) o agente inicia com épsilon igual a 1.0, para ter características exploratórias, após 150 episódios o épsilon que estava em 0,2 foi aumentado para 0,4. É perceptível o aprendizado no treinamento 1, mas o agente não conseguiu mais progredir no aprendizado, oscilando as recompensas por volta de 780.

A partir do resultado do treinamento 1, o treinamento 2 (Figura 3.6b) utilizou os pesos da rede do treinamento 1, e foi configurado para iniciar com épsilon igual a 0.8 e após 150 episódios foi configurado para 0.4. O resultado apresentado confirma que o agente convergiu e conseguiu aprender a rastrear a fonte de gás desviando dos obstáculos para aquela situação, mas é necessário validar esse aprendizado em situações diferentes do treinamento, pois pode ocorrer que o agente tenha aprendido apenas aquela determinada situação e não o comportamento, como é esperado nessa situação.

Para isso foram realizados testes com o robô iniciando em posições diferentes das realizadas no treinamento. Nesta situação o agente pode encontrar a fonte de gás em 92% dos testes. O comportamento do agente treinado pode ser observado na Figura 3.7 e em vídeo[1].

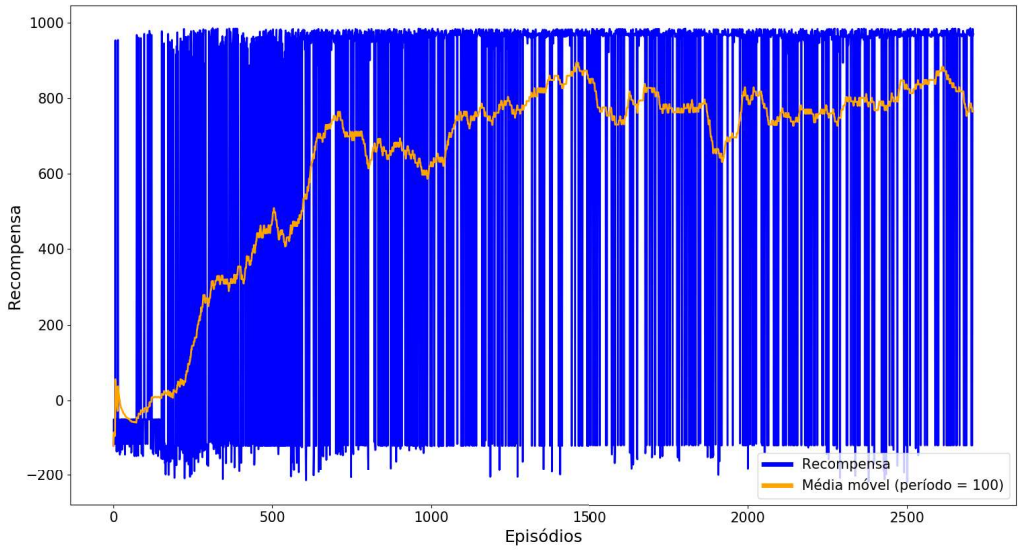
[1] https://www.youtube.com/watch?v=qWwLojPjE_g

Figura 3.5: Ambiente de treinamento com posições iniciais (P=preto) e fonte (vermelho)

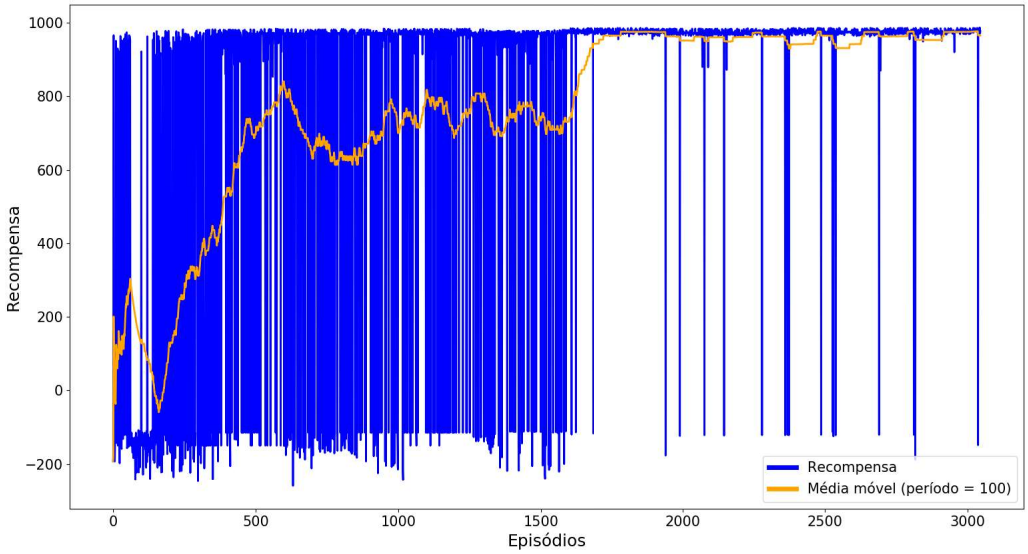


Fonte: O Autor.

Figura 3.6: Gráficos de resultado simulação com obstáculos.



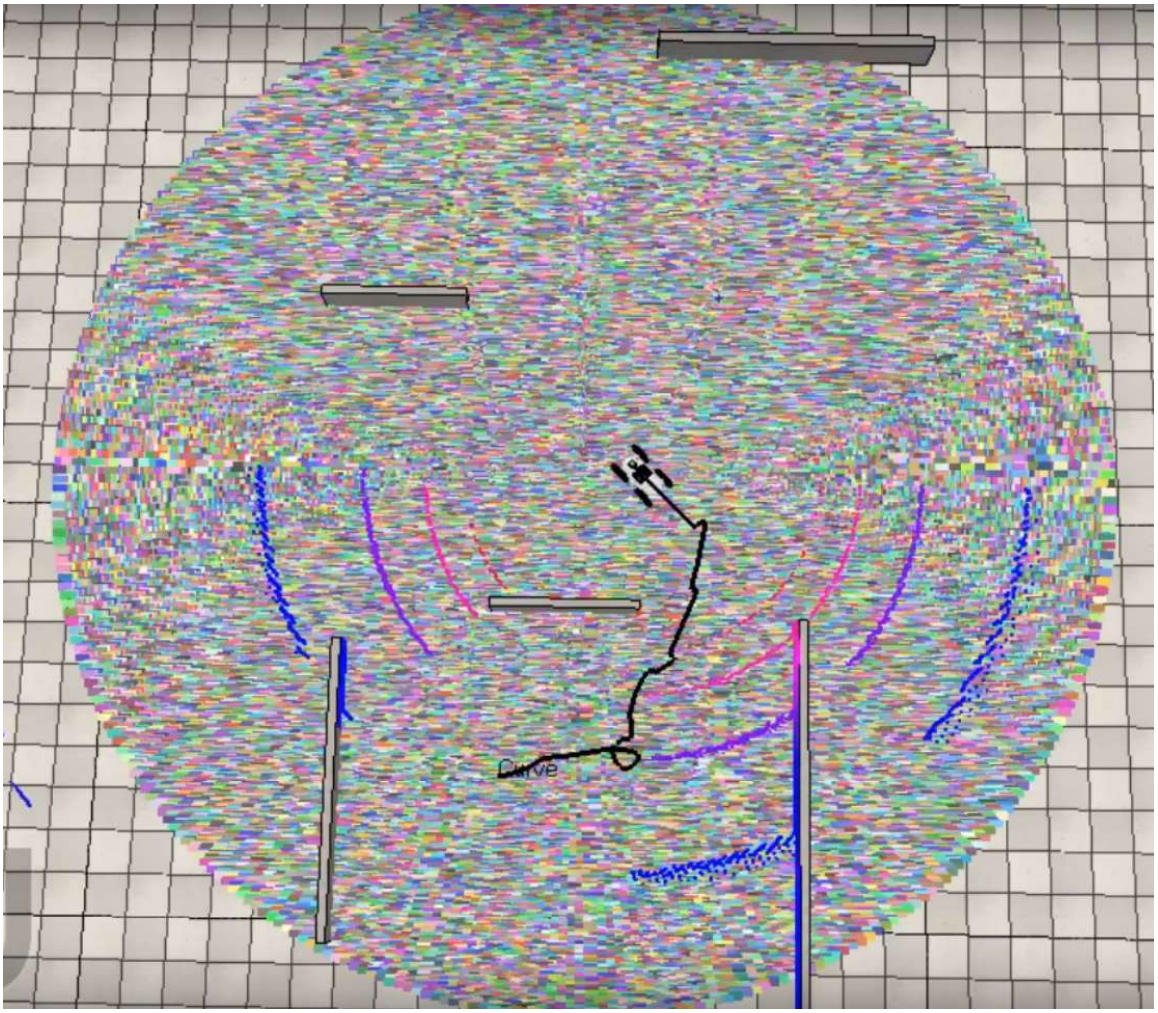
(a) Gráfico com primeira parte do treinamento.



(b) Gráfico com segunda parte do treinamento.

Fonte: O Autor.

Figura 3.7: Trajetória realizada por agente treinado.



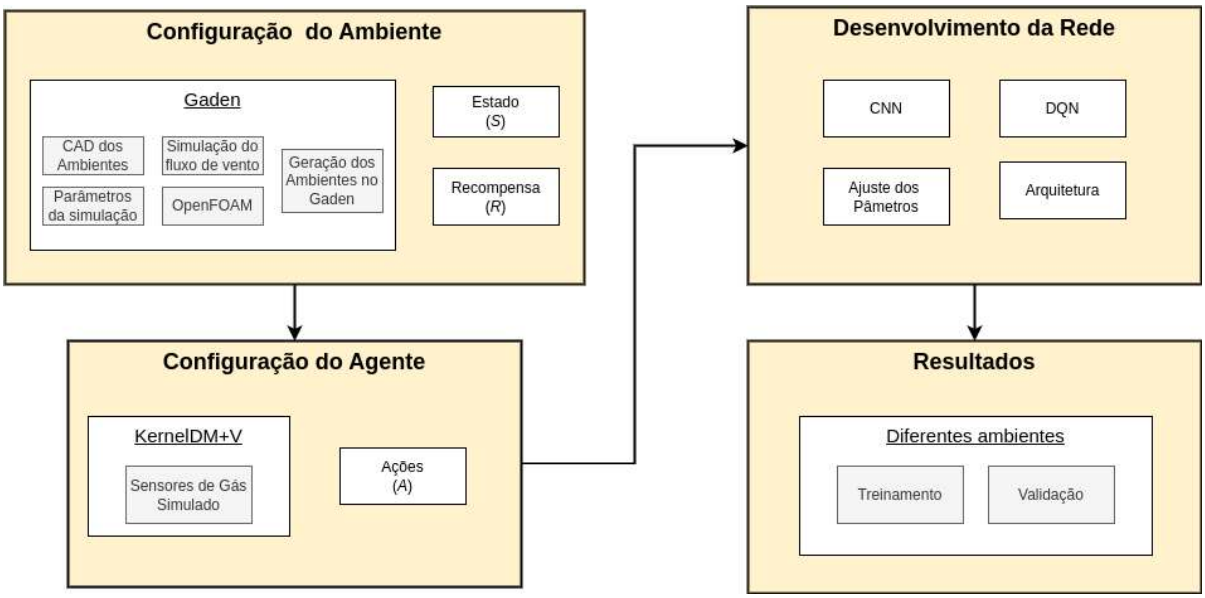
Fonte: O Autor.

4. Rastreamento de pontos de origem de gases em pluma turbulenta

Após os experimentos percebeu-se a necessidade de que a pluma de gás seja mais completa, com melhor modelo de dispersão, onde a pluma interaja com o ambiente, paredes, janelas, fluxo de vento, etc.

Dito isto, esta parte do trabalho têm como atividades (Figura 4.1) o desenvolvimento do ambiente simulado que utiliza o simulador GADEN, o qual é baseado no simulador Stage (GERKEY *et al.*, 2003), que é capaz de representar bem o ambiente real. Em seguida, é apresentado o método de mapeamento (GDM) e os modelos de DQN são apresentados a seguir. Em vez de utilizar apenas redes totalmente conectadas, as redes de convoluções (CNNs) também são analisadas. Por fim, o modelo de aprendizado por reforço é treinado para explorar, localizar e rastrear a fonte de gás.

Figura 4.1: Fluxo de execução das atividades para ambiente com plumas turbulêntas.



Fonte: O Autor.

4.1. Simulador de robótica e dispersão de gás

Existem diversos simuladores de dispersão de gás (POMPY, 2018), porém, todos os trabalhos mais recentes e relevantes de localização de fontes de gás (OJEDA *et al.*, 2020)(DUISTERHOF *et al.*, 2021)(CHEN *et al.*, 2021)(PRABOWO *et al.*, 2020) utilizam o simulador de dispersão *Gas Dispersion Simulator for MobileRobot Olfaction in Realistic Environments* (GADEN) (MONROY *et al.*, 2017a). Simulador este desenvolvido em *Robot Operating System* (ROS) o qual tem intuito de auxiliar na validação de sistemas robóticos e algoritmos de detecção de gás em ambientes realistas.

O GADEN foi capaz de trazer uma grande quantidade de melhorias, como um modelo de dispersão de pluma turbulenta que considera a densidade do gás em ambientes tridimensionais. O qual também considera a colisão das partículas com paredes e objetos pré adicionados a simulação e sua dispersão é variante no tempo. A velocidade de simulação também é um fator diferencial do GADEN, pois ele faz um pré-processamento da dispersão e armazena os dados em um intervalo de tempo configurável em arquivos binários, assim, no momento da simulação é necessário apenas ler os arquivos do pré-processados, isso é essencial para a dinâmica de treinamento do aprendizado pro reforço.

Ele também simula os sensores e suas características, como saturação e velocidade de leitura. Para esse trabalho foram desenvolvidos ambientes em CAD (OnShape), gerados os campos vetoriais de fluxo de vento no SimScale[1], tratados os dados no OpenFOAM[2] e simulados no GADEN (Figura 4.2). Por ser um simulador completo, sua configuração é complexa, assim, foi gerado um tutorial[3] que apresenta passo a passo da configuração de um novo ambiente.

Por estes e outros motivos, o GADEN foi selecionado para ser utilizado neste trabalho. Os ambientes mostrados nas Figuras 4.2c, 4.2i e 4.2l foram gerados para esta dissertação, onde o grau de dificuldade é aumentado conforme avança de ambiente. Os parâmetros da simulação por ambiente são apresentados na Tabela 4.1

Tabela 4.1: Tabela com parâmetros utilizados para a simulação.

Parâmetros/Ambientes	Ambiente 1	Ambiente 2	Ambiente 3	Ambiente 4
Densidade (kg/m^3)	1.1965	1.1965	1.1965	1.1965
Viscosidade (m^2/s)	0.000015295	0.000015295	0.000015295	0.000015295
Ux (m/s)	0.003	1.5	0.07	0.1
Uy (m/s)	0.003	0.6	0	0
Uz (m/s)	0	0	0	0
Pressão de saída (janela) (Kpa)	10	80	0	0
Número de filamentos liberados por segundo	0 a 10	0 a 100	0 a 100	0 a 100
Temperatura do ambiente(k)	303	303	303	303
Pressão do ambiente (atm)	1	1	1	1

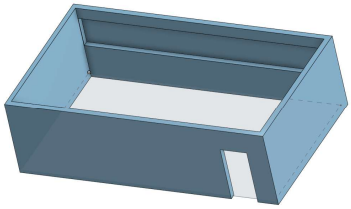
Fonte: O Autor.

4.2. Técnica de Mapeamento

Para que um algoritmo de aprendizado de máquina obtenha sucesso no treinamento é necessária uma boa rede que consiga abstrair o problema, configurar bons parâmetros e que essa rede tenha informação suficiente para poder extrair as informações.

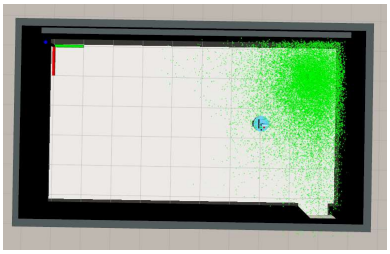
[1] www.simscale.com
[2] www.openfoam.org
[3] https://github.com/MSathler/tutorial_gaden_environment/blob/main/Gaden_Environment_Tutorial.pdf

Figura 4.2: Ambientes gerados.

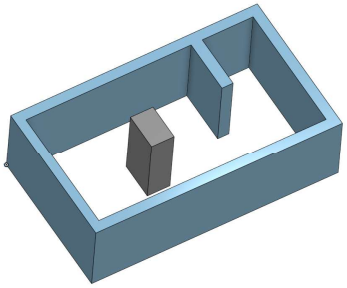


(a) CAD de uma sala vazia com porta e janelas.

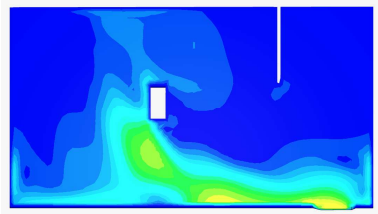
(b) Simulação no de fluxo de vento.



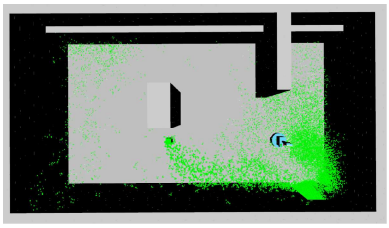
(c) Simulação de dispersão de gás visualizado no Rviz ambiente 1.



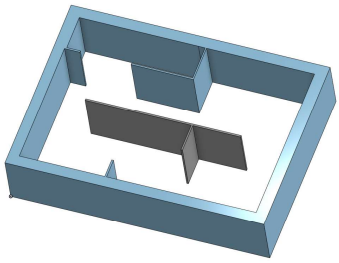
(d) CAD de uma sala com alguns obstáculos.



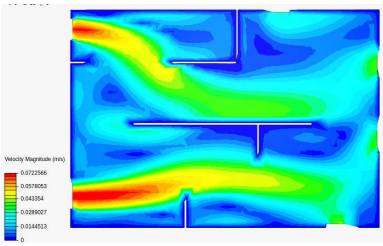
(e) Simulação no de fluxo de vento.



(f) Simulação de dispersão de gás visualizado no Rviz ambiente 2.

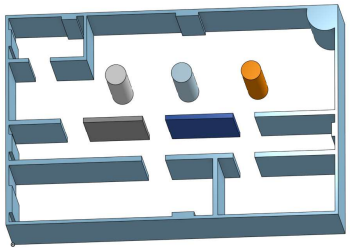


(g) CAD de um ambiente com alguns cômodos.

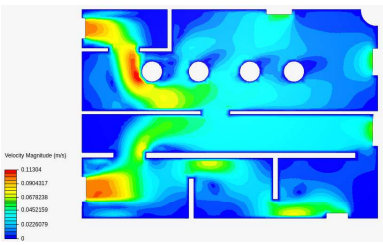


(h) Simulação no de fluxo de vento.

(i) Simulação de dispersão de gás visualizado no Rviz ambiente 3.



(j) CAD de um ambiente extenso com diversos cômodos.



(k) Simulação no de fluxo de vento.

(l) Simulação de dispersão de gás visualizado no Rviz ambiente 4.

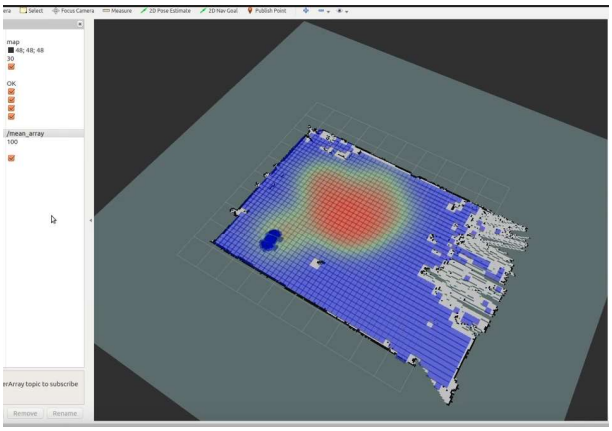
Fonte: O Autor.

Dentre os pontos de configuração de um bom algoritmo de aprendizado de máquina, o ponto de melhoria analisado referente ao Capítulo 2.1 é a possibilidade de aumentar o nível de informação de entrada utilizada na DQN, e com isso obter um nível maior de entendimento/abstração do ambiente pelo agente. Com esse aumento do nível de informação é possível realizar o rastreamento de fontes de gás em ambientes mais complexos, como: ambientes que possuam plumas turbulentas e ambientes maiores e mais complexos.

Para que seja possível aumentar o nível de informação de entrada da DQN foi proposto substituir as entradas anteriores (posição e orientação relativa, leitura do sensor de gás, leitura do sensor de distância) por mapas de calor, onde esses mapas contêm as seguintes informações: (i) Mapa de concentração média (adaptativo): esse mapa tem como intuito exibir onde estão as concentrações do gás.Este é o mapa principal, que retorna mais informações uteis; (ii) Mapa de confiança: apresenta por onde o robô já passou, é bem útil para o algoritmo ter característica exploratória e percorrer melhor o ambiente; (iii) mapa de ocupação, que apresenta as características físicas do ambiente, como os obstáculos e onde o robô se encontra.

Para gerar os mapas (i) e (ii) será utilizado o método de mapeamento KernelDM+V[4]. Método esse que foi adaptado para a aplicação (Figura 4.3), onde em nosso problema não será utilizado sensor anemômetro, e feitas algumas adaptações de parâmetros e nova forma de visualização pelo *ROS Visualization* (RViz).

Figura 4.3: Mapeamento de concentração de gás em ambiente real utilizando o KernelDM+V adaptado (vermelho: alta concentração. azul: baixa concentração).



Fonte: O Autor.

4.3. Parâmetros e Configurações

O modelo que primariamente será testado é o de *Deep Q Learning* (DQN), modelo esse baseado nas técnicas tradicionais de aprendizado por reforço, esse método possui algumas limitações como trabalhar com saídas discretas e maior tempo para convergir. No entanto, sua vantagem é ser simples e tem bons resultados, como pode ser analisado em Baselines (2020).

[4] https://github.com/h3ct0r/gas_mapping_kerneldm

Como mencionado na Seção 4.2, as entradas da rede serão os mapas gerados pelo algoritmo kernelDM+V (Figuras 4.4a e 4.4b). Além disso, um mapa de ocupação (Figura 4.4c) gerado pelo *gmapping* adaptado será incluído. Este mapa apresenta os obstáculos (preto) e a posição do robô (verde).

As posições de saída da rede, ou seja, as ações, também são adicionadas ao mapa. Se a ação não resultar em colisão, ela é representada em azul; caso contrário, é representada em amarelo. Vale ressaltar que esses mapas são matrizes grandes. Por conta disso, um ponto de análise desse trabalho é verificar a performance das redes convolucionais. Utilizadas em conjunto com o aprendizado por reforço, essas redes têm o potencial de reduzir a informação, permitindo que o modelo opere de maneira mais eficiente.

Para otimização de recursos computacionais é necessário a redução dos mapas de entrada, assim, reduzindo o consumo de memória RAM do computador. A redução das entradas da rede não pode ser exagerada, pois com grandes reduções, os mapas perdem as características. Assim, os mapas terão resolução de (52x31).

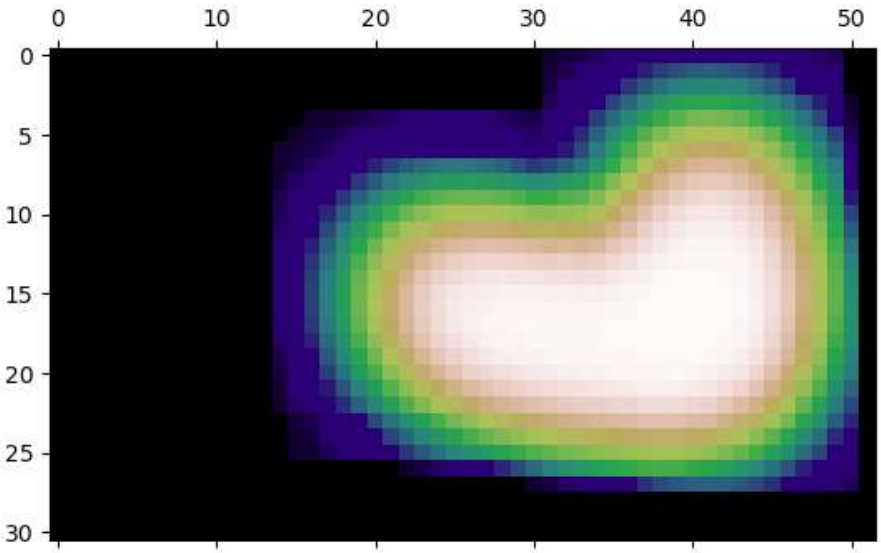
Os parâmetros utilizados nos treinamentos foram:

- Buffer Size: 14.200, este é responsável pelo armazenamento de experiências e teve seu valor definido com intuito de armazenar todas elas. O buffer size é armazenado na memória RAM, então um grande valor de buffer size exige maior quantidade de memória RAM.
- Batch Size: 128, representa o tamanho do lote de experiências que serão utilizadas para o aprendizado. O batch size é processado em GPU, então, o seu tamanho precisa ser dimensionado com o tamanho da VRAM.
- Gamma: 0,95, é o fator de desconto.
- Tau: 0,001, é o parâmetro de interpolação da função *soft update* para atualizar os parâmetros da rede target.
- Learning Rate: 0,0003, é a taxa de aprendizado.
- Update_every: 1, é a taxa de atualização, onde a rede target é atualizada a cada nova experiência.
- Otimizador: Adam (ALGORITHMIA, 2018).

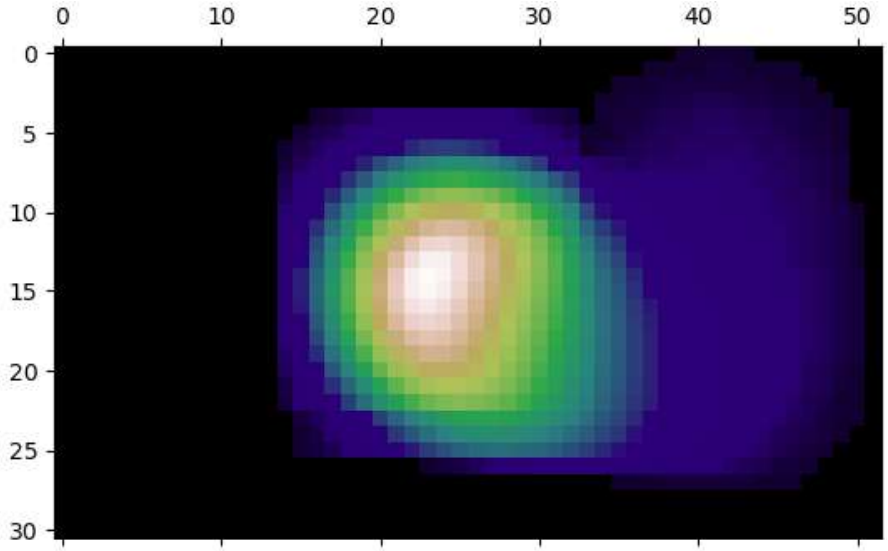
As saídas da rede serão 4 posições de navegação que são 20 cm deslocados nas direções norte, sul, leste e oeste. O algoritmo que realizará a navegação é o *navigation* (ROS, 2020), que é um pacote do ROS, onde basta adaptar os tópicos e já está pronto para usar.

As recompensas e condições de parada são apresentadas na Equação 4.1:

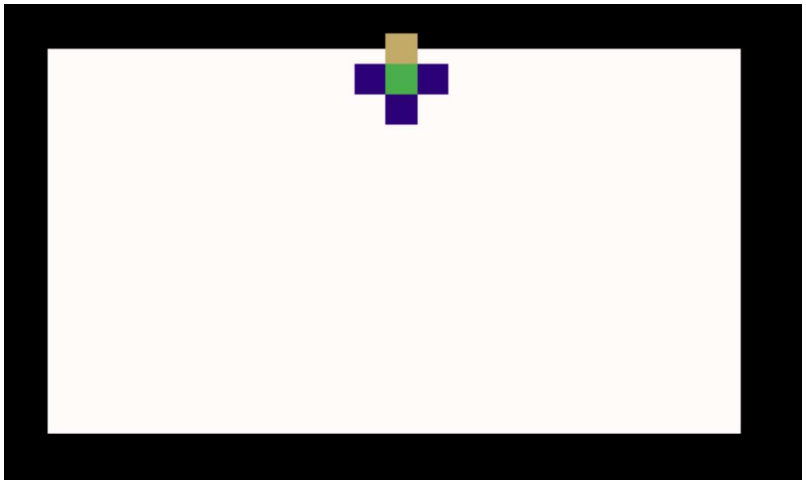
Figura 4.4: Mapas de entrada da CNN.



(a) Mapa de confiança que apresenta por onde o robô já fez leitura.



(b) Mapa de concentração média.



(c) Mapa de ocupação.

Fonte: O Autor.

$$r(s,a) = \begin{cases} -1 & \text{por iteração.} \\ +1 & \text{estar dentro da pluma.} \\ +1 & \text{por exploração (se a média dos valores do mapa de confiança aumentar).} \\ +100 & \text{encontro da fonte (Condição de parada).} \\ -100 & \text{colisão (Condição de parada).} \\ +0 & \text{mais de 80 iterações (Condição de parada).} \end{cases} \tag{4.1}$$

4.4. Treinamento, Ajustes e Resultados

Os treinamentos foram executados em um computador 32GB de memória RAM, processador Intel AMD Ryzen 5, placa gráfica NVIDIA RTX 3060 12GB VRAM, com sistema operacional Ubuntu 20.04 LTS

Como apresentado no Capítulo 2 o o GSL tem três objetivos, busca da pluma, rastreio da pluma e declaração da fonte, para esse treinamento não haverá a declaração da fonte por meio do agente, o encontro da fonte é definido no simulador que retorna o encontro da fonte.

4.4.1. Ambiente 1

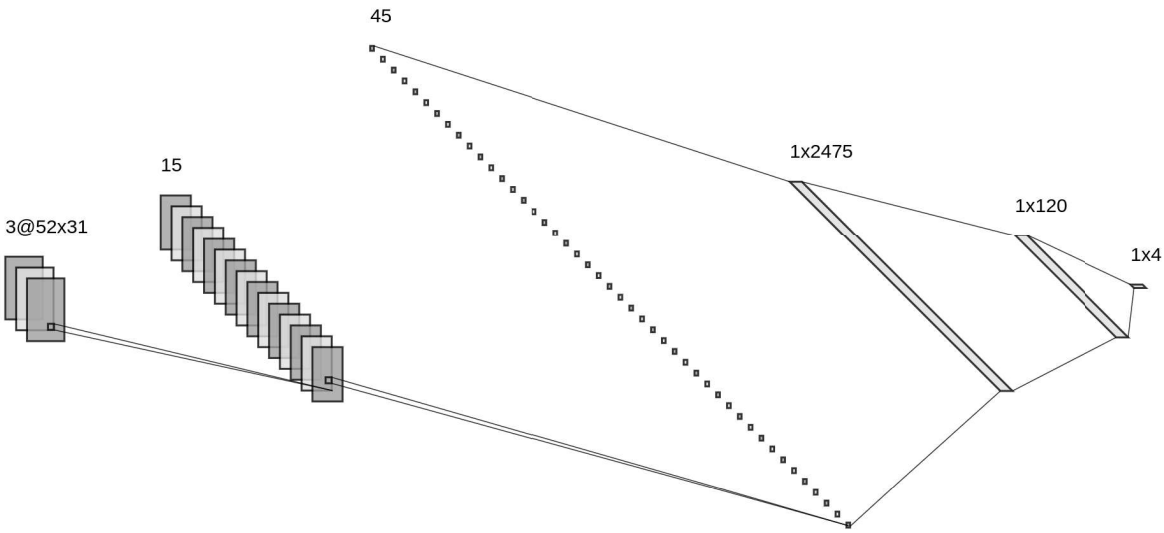
Os parâmetros utilizados serão os mesmos e o que irá variar são as arquiteturas. As arquiteturas utilizadas são: (i) 2 camadas de 64 perceptrons cada e relu de ativação; (ii) Convolução proposta (Figura 4.5); (iii) DenseNet; (iv) EfficientNet (Figura 2.8). A ideia da DenseNet (HUANG *et al.*, 2018) e EfficientNet é utilizar CNNs mais complexas e validadas, e aproveitar os pesos pré-treinados que possuem bons resultados na ImageNet (IMAGENET, 2021). Essas CNNs são redes treinadas para classificar imagens, uma abordagem diferente da utilizada no aprendizado por reforço, porém, a utilização dos pesos pré-carregados é válida. Sendo possível aproveitar partes do aprendizado, isso pode acelerar o aprendizado do rastreamento de fontes de gás.

O primeiro treinamento é realizado em condições mais simples para validar os parâmetros utilizados. Para esse treinamento a fonte é posicionada em uma posição fixa e o robô é inciado em posições aleatórias dentro do ambiente e fora da fonte. O treinamento ocorreu no ambiente apresentado na Figura 4.2c. Todas as arquiteturas foram capazes de convergir com esses parâmetros nesta situação.

O segundo treinamento foi configurado para a fonte e o robô variarem de posições (Figura 4.8). As posições simuladas de fonte de gás são apresentadas na Figura 4.6.

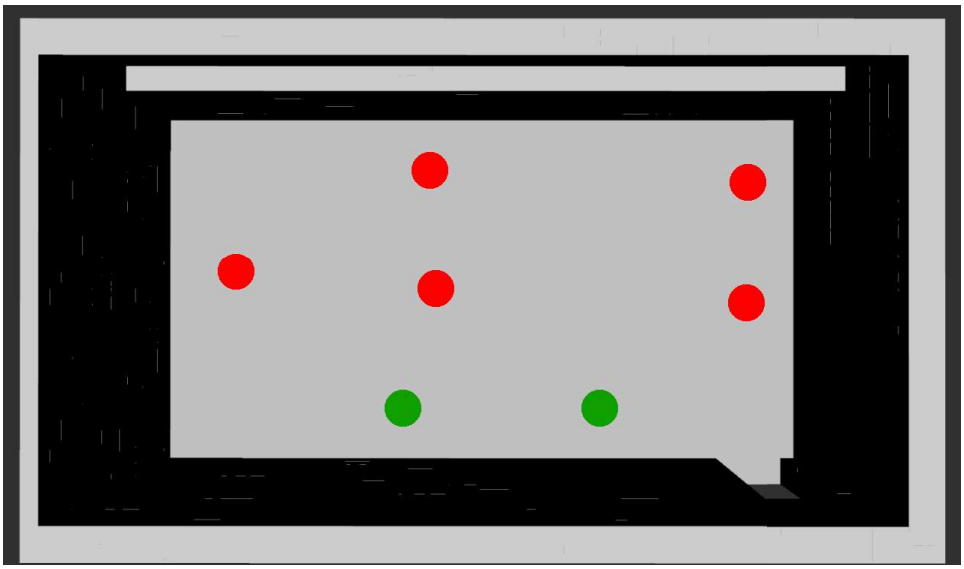
No segundo treinamento não foi carregado os parâmetros do treinamento anterior. A Tabela 4.2 apresenta a quantidade de episódios necessários para convergir o treinamento referente

Figura 4.5: Convolução proposta.



Fonte: O Autor.

Figura 4.6: Posições de fontes de gás simuladas. Em vermelho posições utilizadas para treinamento e em verde posições utilizadas para testes (Teste1 = esquerda, Teste2 = direita).



Fonte: O Autor.

a cada arquitetura, onde nota-se que a arquitetura de MLPs demora 10x mais episódios que as outras arquiteturas para convergir.

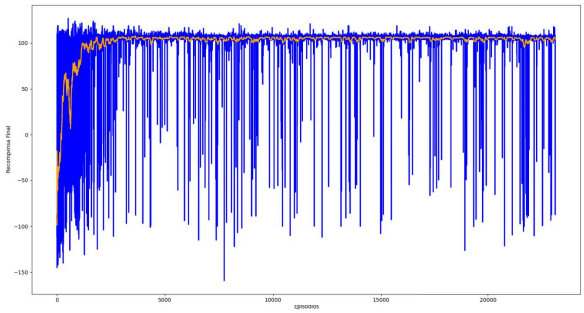
Tabela 4.2: Comparação tempo para convergir por arquitetura.

Ambientes	(i)	(ii)	(iii)	(iv)
Número de episódios para convergir	2600	220	240	280

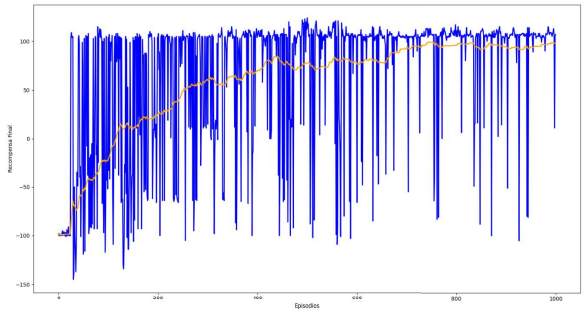
Fonte: O Autor.

Os resultados são apresentados nas Figuras 4.7a, 4.7b, 4.7c, 4.7d. Vale ressaltar que a rede (i) teve um tempo de análise maior pois não convergiu totalmente.

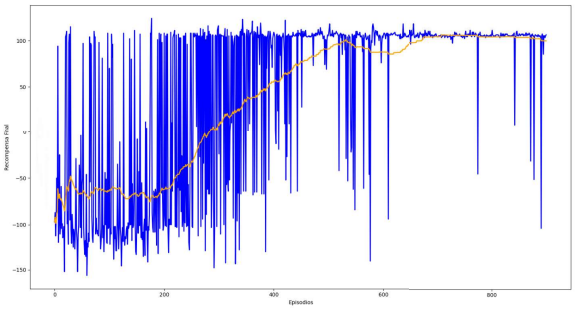
Figura 4.7: Resultados gerados para o treinamento com o robô e a fonte em posições aleatórias.



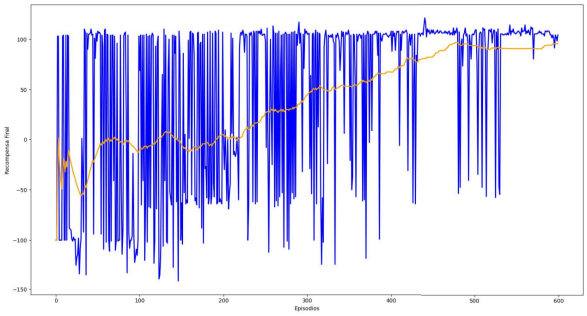
(a) Gráfico recompensa por episódios arquitetura DL (i).



(b) Gráfico recompensa por episódios arquitetura CNN proposta (ii).



(c) Gráfico recompensa por episódios arquitetura EfficientNet (iii).



(d) Gráfico recompensa por episódios arquitetura DenseNet (iv).

Fonte: O Autor.

Para verificar se o agente realmente aprendeu o comportamento esperado é necessário realizar testes onde o robô deve encontrar a fonte de gás em posições o qual o agente não foi treinado. A Figura 4.6 sinaliza em verde as posições utilizadas para validar os agentes. Os testes realizados seguem de forma que o robô é colocado aleatoriamente no ambiente 100 vezes. A Tabela 4.3 apresenta os resultados por arquitetura dos testes realizados e a Figura 4.9b apresenta a trajetória e o comportamento do agente rastreando a fonte de gás[5].

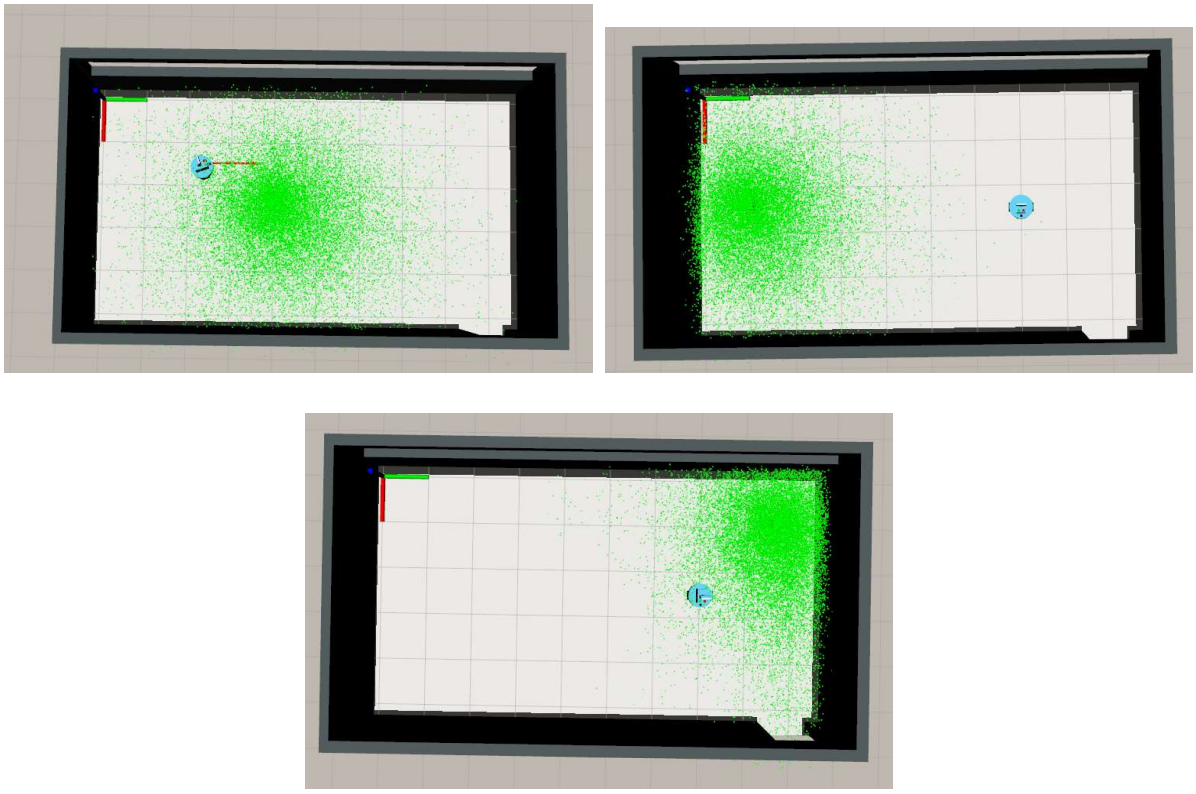
[5] https://www.youtube.com/watch?v=ac_dNKRZGuE

Tabela 4.3: Resultados dos testes realizados no ambiente 1.

Testes/Arquiteturas e número de episódios treinados	(i) 23.000	(ii) 1.000	(iii) 900	(iv) 600
Teste 1	64%	65%	52%	95%
Teste 2	61%	22%	62%	86%

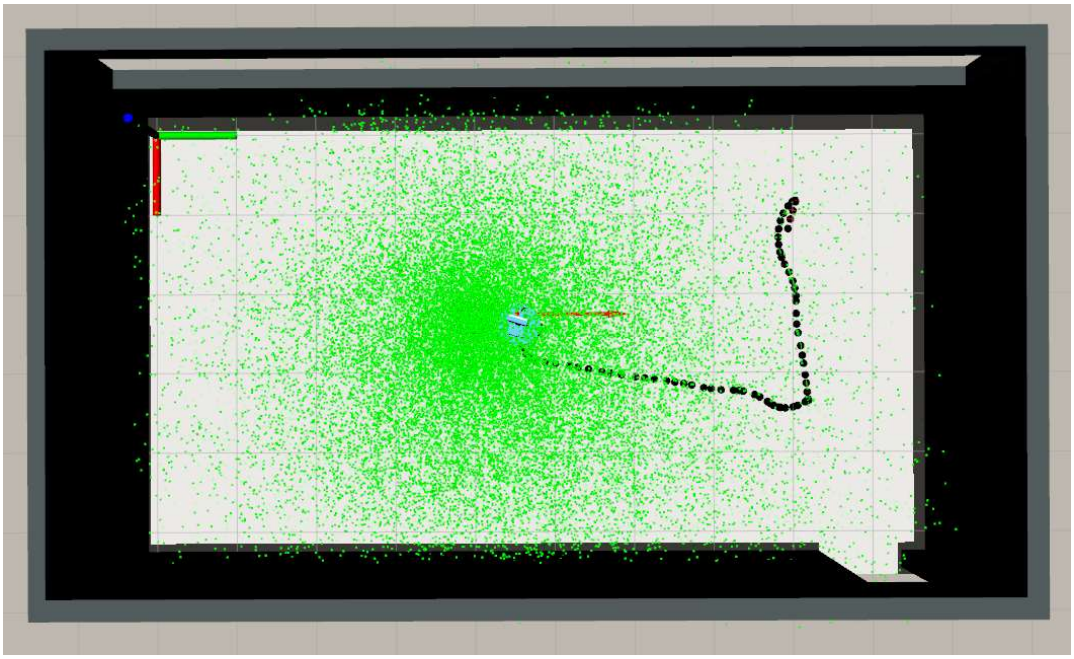
Fonte: O Autor.

Figura 4.8: Fonte gerada em diferentes posições para o treinamento.

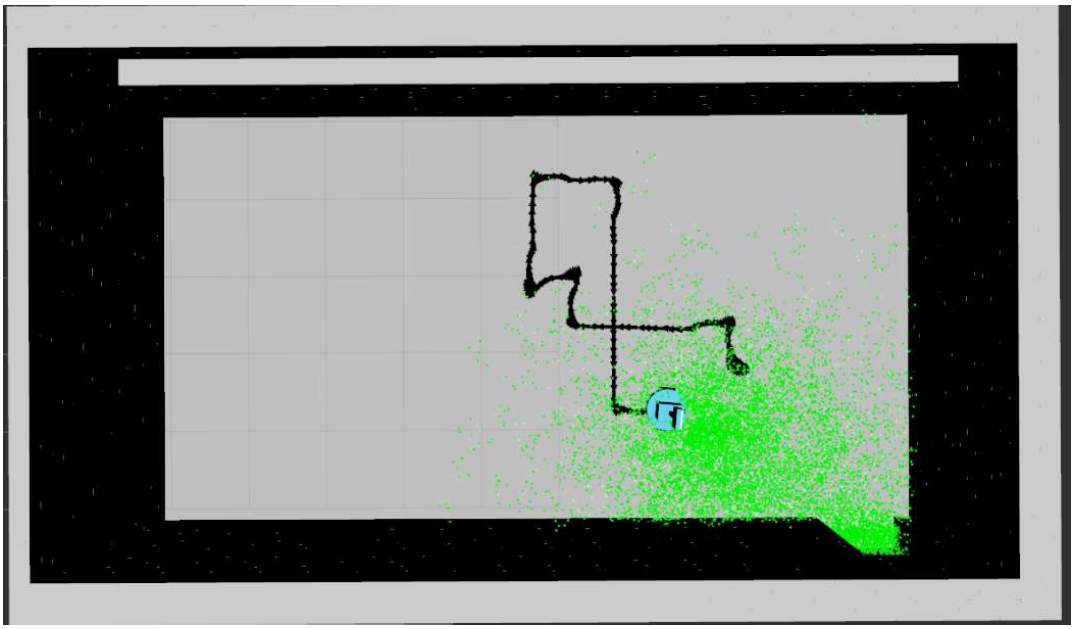


Fonte: O Autor.

Figura 4.9: Trajetórias feitas pelo agente treinado no ambiente 1.



(a) Trajetória realizada por agente treinado (fonte em posição de treinamento).



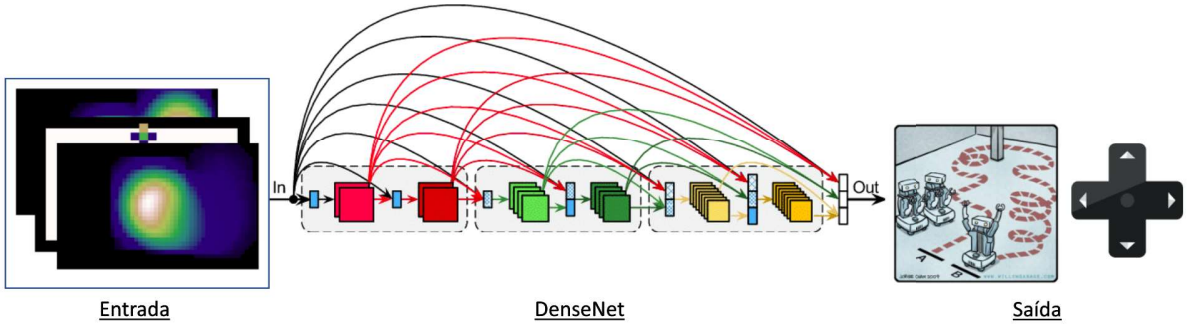
(b) Trajetória realizada por agente treinado (fonte em posição de teste).

Fonte: O Autor.

4.4.2. Ambiente 2

O ambiente 2 é uma adaptação do primeiro, onde foi adicionado um obstáculo ao centro da sala, uma parede e também é adicionado o fluxo de ar (Tabela 4.1) que entra pela janela e sai na porta. Para essa etapa será utilizada a arquitetura que melhor performou no teste passado, a DenseNet (Figura 4.10).

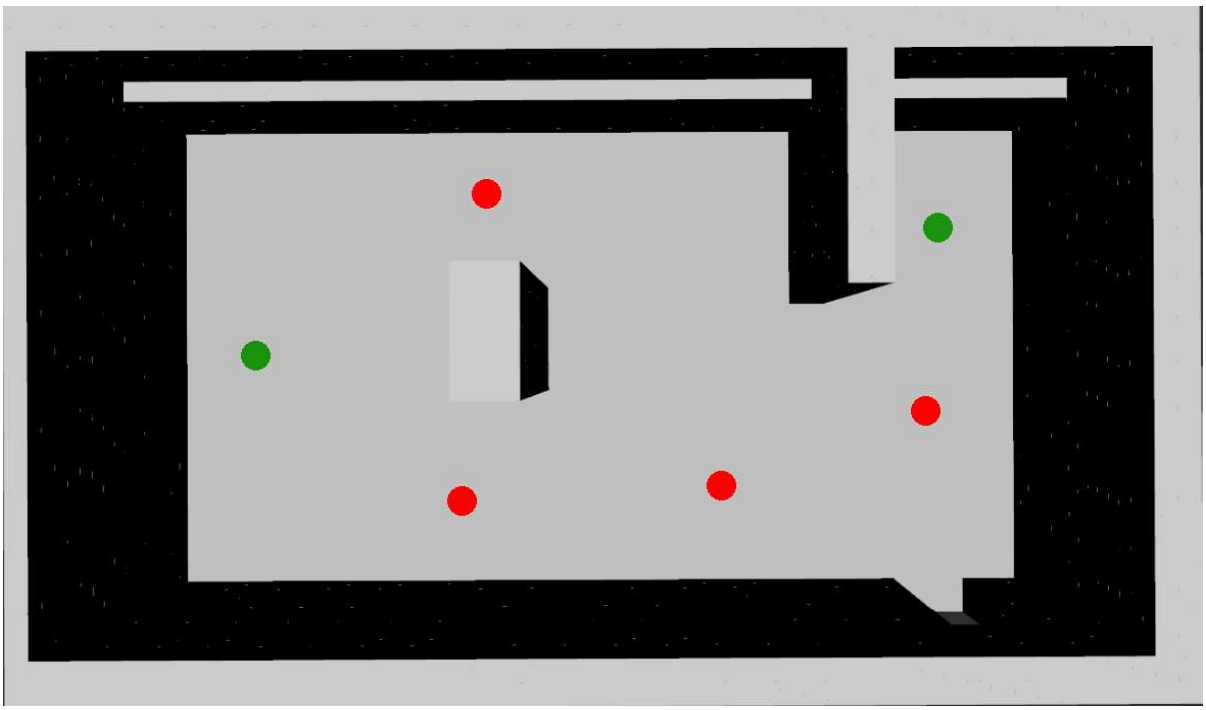
Figura 4.10: Estrutura com a DenseNet.



Fonte: O Autor.

O treinamento é realizado com a fonte (Figura 4.11) e o robô alternando de posições iniciais. Por haver um fluxo de ar considerável, a taxa de dispersão de gás da fonte é aumentada para esse treinamento de 10 para 100 por segundo, pois, caso contrário todo o gás seria rapidamente evacuado do ambiente.

Figura 4.11: Posições de fontes de gás simuladas. Em vermelho posições utilizadas para treinamento e em verde posições utilizadas para testes. (Teste1 = esquerda, Teste2= direita)



Fonte: O Autor.

Por dois motivos foi necessário adaptar a condição de parada: (i) dispersão de gás com fluxo de ar; (ii) aumento da evacuação da fonte. A dispersão de gás com fluxo de ar sempre gera bolsões de ar, então a condição de parada considerará caso o robô em um raio de 1m da fonte de gás com uma leitura superior a 200.

Após rodar o treinamento com os mesmos parâmetros da DQN apresentados na Seção 4.3 foram obtidos os resultados apresentados na Figura 4.12a. No entanto, apesar de ter convergido no treinamento a condição de parada de 1m da fonte não apresentou bons resultados nos testes, 100 tentativas por posição de teste, com baixa taxa de assertividade ($\approx 5\%$) em ambos testes. Esse resultado mostra que o agente não aprendeu o comportamento esperado de rastrear fonte de gás, e sim decorou as posições de fonte do treinamento.

A partir do resultado anterior, foi realizado o ajuste na distância da condição de parada de 1m para 0.5m, onde a Figura 4.12b apresenta o resultado do treinamento, que aparentemente oscilou mais. No entanto, obteve melhores resultados nos testes (Tabela 4.4). Essa assertividade não é tão alta, porém mostra que o agente conseguiu começar a compreender como rastrear fontes de gás turbulentas, que pode ser analisado nas Figuras 4.13a e 4.13b[6].

Para obter resultados melhores é necessário que o agente realize mais treinamentos e que a quantidade de situações aumente, como treinar nos outros ambientes e também variar mais as possibilidades de posicionamento da dispersão de gás.

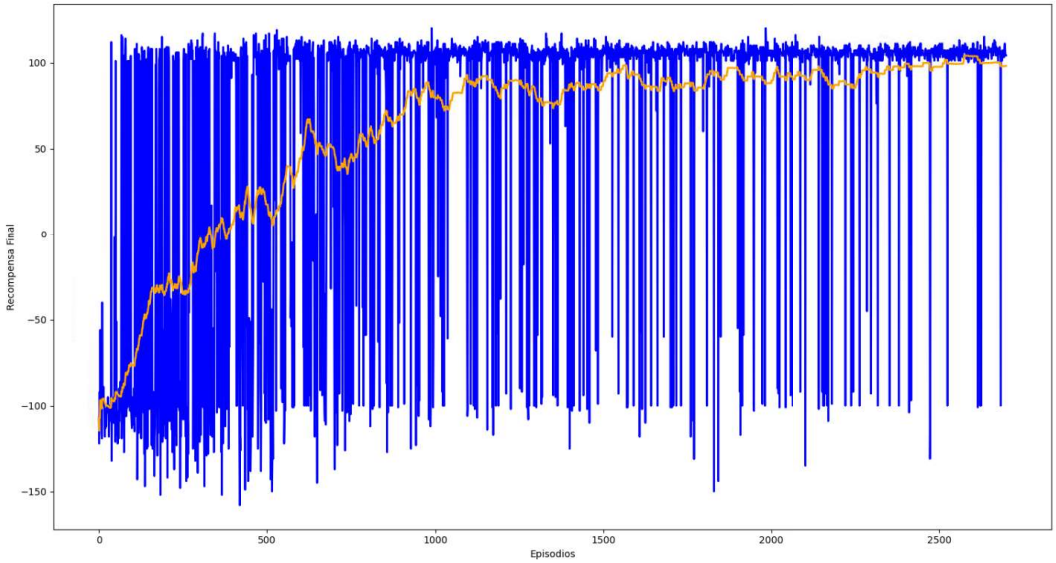
Tabela 4.4: Resultados dos testes realizados no ambiente 2.

Testes/Assertividade	
Teste 1	75%
Teste 2	45%

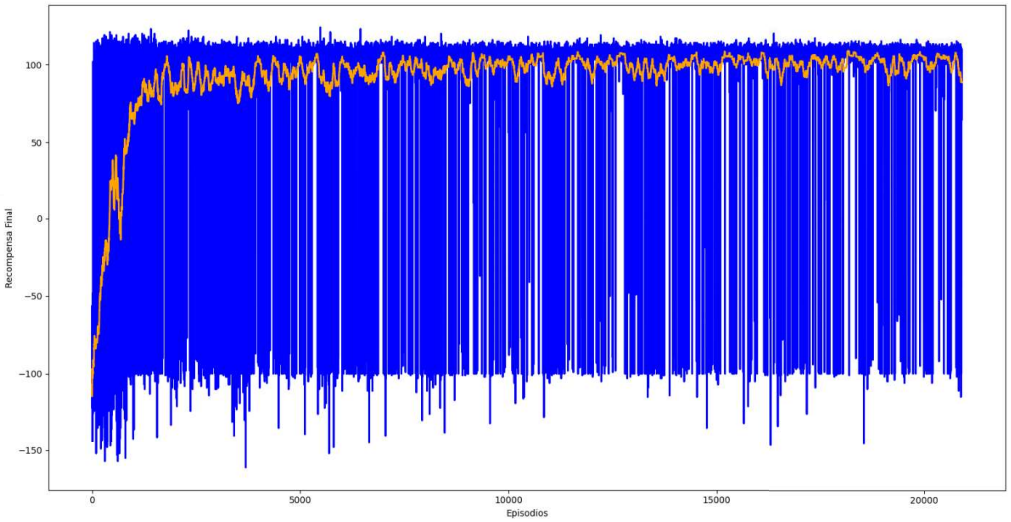
Fonte: O Autor

[6] https://www.youtube.com/watch?v=BBwuoWpoKuY

Figura 4.12: Resultado do treinamento no ambiente 2.



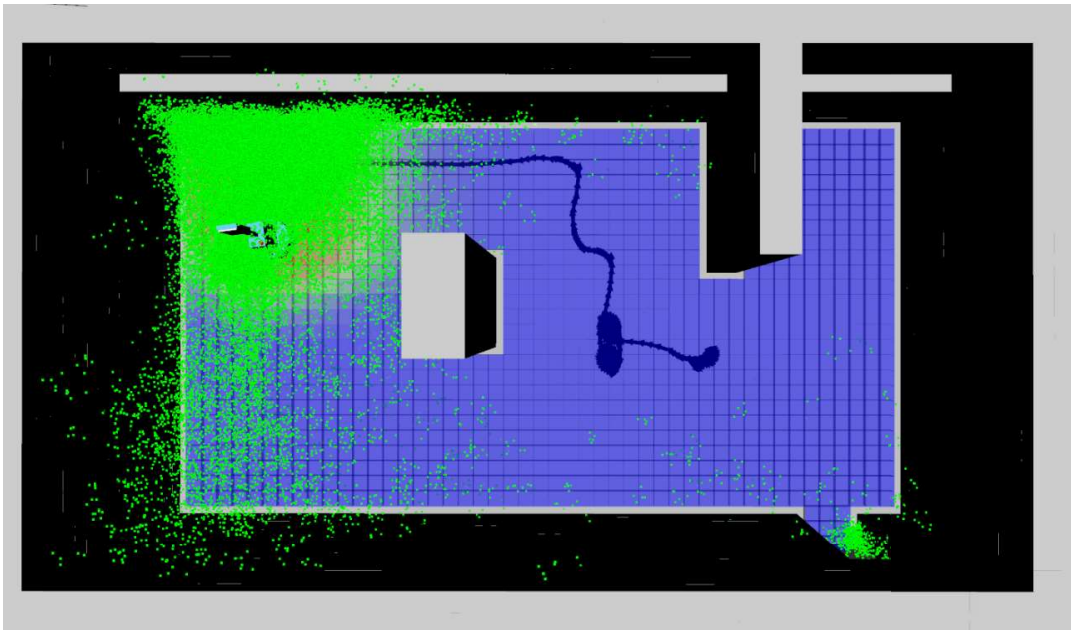
(a) Condição de parada a 1 metro da fonte.



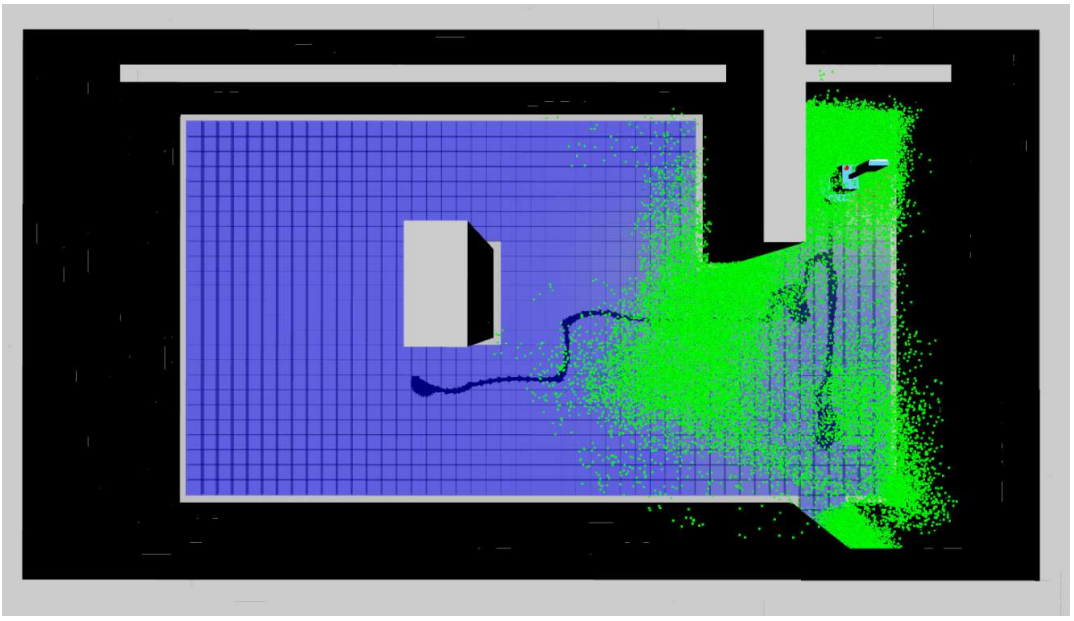
(b) Condição de parada a 0.5 metros da fonte.

Fonte: O Autor.

Figura 4.13: Movimentação do treinamento no ambiente 2.



(a) Teste 1.



(b) Teste 2.

Fonte: O Autor.

5. Considerações finais e trabalhos futuros

Foi apresentado uma primeira metodologia em que foi possível resolver o problema de rastreamento de fonte de gás laminar com desvio de obstáculos. Esta primeira abordagem apresentou limitações sobre o modelo de dispersão, pois não apresenta interação das partículas de gás com o ambiente e não é capaz de gerar outros modelos de plumas mais realistas.

A segunda metodologia propôs solucionar o problema de rastreamento de plumas de gases turbulentas no simulador GADEN, para isso foi apresentada uma nova abordagem de utilização da DQN. Esta abordagem se baseia em uma maior quantidade de informação de entrada, que é a utilização das técnicas de GDM em conjunto com mapa de ocupação para alimentar a DQN. Com este novo formato de entrada, foram comparadas diversas arquiteturas e foi comprovado que as redes convolucionais obtiveram melhor desempenho em relação as redes MLPs. No entanto, também foram identificados desafios e limitações, sendo necessário maior poder computacional para o treinamento dessas redes.

5.1. Trabalho Futuros

Na segunda etapa este trabalho mostrou resultados para dois dos quatro cenários propostos no simulador GADEN. A performance apresentada pelo agente pode ser melhorada, então como trabalho futuro é importante realizar mais treinamentos variando as possibilidades de ambientes e de posições de plumas, também é interessante ter episódios onde o ambiente não tem pluma de gás, nesse caso adaptando o sistema de recompensas para beneficiar mais a exploração.

Outro trabalho futuro é a validação desses resultados em cenário real, para isto, é necessário utilizar a rede pré-treinada e fazer uma pequena adaptação. A adaptação necessária é converter o mapa gerado pelo *gmapping* em um imagem com a adição da posição do robô. Seguindo a ideia dos ambientes reais que utilizam algoritmos de mapeamento que necessitam da conversão desses mapas para imagem, surge uma necessidade para ambientes muito grandes, que seria o recorte dos mesmos, uma solução seria um mapa local (janela flutuante) que assim mantém um tamanho de entrada fixo da rede, o que evita problemas de manipulação da entrada das CNNs.

Também podem ser adicionadas mais saídas ao modelo, como a saída referente à declaração da fonte, pode ser uma informação binária ou até mesmo coordenadas numéricas.

Esse estudo mostrou que as redes convolucionais podem performar bem em conjunto com o aprendizado por reforço, contudo existem diversos tipos de arquiteturas de CNNs que não foram testadas. Um próximo estudo pode ser a análise de qual arquitetura de CNNs melhor performa nesse cenário, alguns exemplos seriam outras CNNs de classificação como a VGG, LeNet, MobileNet, ou até mesmo as de Segmentação como as UNets.

Referências Bibliográficas

AGENCY, D. A. R. P. "**Unearthing the Subterranean Environment - Subterranean Challenge**". 2021. Disponível em: <https://www.subtchallenge.com/index.html#about>. Acesso em: 14/02/2022.

AGHA, A., OTSU, K., MORRELL, B., et al.. "**NeBula: Quest for Robotic Autonomy in Challenging Environments; TEAM CoSTAR at the DARPA Subterranean Challenge**". 2021. Acesso em: 01/03/2022.

ALGORITHMIA. "**Introduction to optimizers**". 2018. Disponível em: <https://algorithmia.com/blog/introduction-to-optimizers>. Acesso em: 06/04/2021.

AZPÚRUA, H., REZENDE, A. M. C., POTJE, G. A., et al.. "**Towards Semi-autonomous Robotic Inspection and Mapping in Confined Spaces with the EspeleoRobô**", *Journal of Intelligent & Robotic Systems*, v. 101, 2021. Disponível em: <https://api.semanticscholar.org/CorpusID:233313044>.

BARLOW, H. B. "**Unsupervised learning**", *Neural computation*, v. 1, n. 3, pp. 295–311, 1989.

BASELINES, S. "**RL Algorithms**". 2020. Disponível em: <https://stable-baselines.readthedocs.io/en/master/guide/algos.html>. Acesso em: 10/04/2022.

BROCKMAN, G., CHEUNG, V., PETTERSSON, L., et al.. "**OpenAI Gym**", *ArXiv*, v. abs/1606.01540, 2016.

CHEN, X., HUANG, J. "**Odor source localization algorithms on mobile robots: A review and future outlook**", *Robotics and Autonomous Systems*, v. 112, 11 2018.

CHEN, X., FU, C., HUANG, J. "**A Deep Q-Network for robotic odor/gas source localization: Modeling, measurement and comparative study**", *Measurement*, v. 183, pp. 109725, 2021. ISSN: 0263-2241. doi: https://doi.org/10.1016/j.measurement.2021.109725. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0263224121006862>.

CID, A., NAZÁRIO, M., SATHLER, M., et al.. "**A Simulated Environment for the Development and Validation of an Inspection Robot for Confined Spaces**". Em: *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pp. 1–6, 2020.

COPPELIAROBOTICS. "**CoppeliaSim**". 2019. Disponível em: <https://www.coppeliarobotics.com/>. Acesso em: 23/03/2021.

CYBERBOTICS. "**Getting Started with Webots.**" 2020. Disponível em: <https://cyberbotics.com/doc/guide/getting-started-with-webots>. Acesso em: 25/05/2020.

DAVIES, A., LOUIS, M., WEBB, B. "**A Model of Drosophila Larva Chemotaxis**", *PLOS Computational Biology*, v. 11, pp. e1004606, 11 2015.

DHARIWAL, A., SUKHATME, G., REQUICHA, A. "**Bacterium-inspired robots for environmental monitoring**". Em: *IEEE ICRA 2004*, v. 2, pp. 1436–1443 Vol.2, 2004.

DO AÇO, D. "**Explosão na usina da Gerdau deixa três mortos.**" 2016. Disponível em: <https://www.diariodoaco.com.br/noticia/0002283>. Acesso em: 19/03/2021.

DO VALE, D. "**Trabalhador morre em acidente no AF-3 da CSN**". 2023. Disponível em: <https://diariodovale.com.br/tempo-real/trabalhador-morre-em-acidente-no-af-3-da-csn/>. Acesso em: 18/06/2023.

DUISTERHOF, B. P., LI, S., BURGUÉS, J., et al.. "**Sniffy Bug: A Fully Autonomous Swarm of Gas-Seeking Nano Quadcopters in Cluttered Environments**". 2021. Acesso em: 26/07/2022.

DYNAMICS, B. "**Mobile Robot Guide**". 2020. Disponível em: <https://mobilerobotguide.com/2020/08/27/boston-dynamics/>. Acesso em: 13/10/2022.

GAZEBO. "**Home**". 2019. Disponível em: <http://gazebosim.org/>. Acesso em: 25/05/2020.

GERKEY, B., VAUGHAN, R. T., HOWARD, A., et al.. "**The player/stage project: Tools for multi-robot and distributed sensor systems**". Em: *Proceedings of the 11th international conference on advanced robotics*, v. 1, pp. 317–323. Citeseer, 2003.

GLOBO. "**Funcionário de siderúrgica morre após vazamento de gás em Bom Despacho**". 2022. Disponível em: <http://glo.bo/3TyGjBE>. Acesso em: 18/06/2023.

GLOBO. “**Explosão de gasômetro na Usiminas foi causada por falha técnica de equipamento**”. 2018. Disponível em: <https://g1.globo.com/mg/vales-mg/noticia/2018/08/17/explosao-de-gasometro-na-usiminas-foi-causada-por-falha\-tecnica-de-equipamento.ghtml>. Acesso em: 19/03/2021.

GLOBO, O. “**Funcionários de siderúrgica passam mal após inalar gás em Itaúna**”. 2020. Disponível em: <http://glo.bo/3tlExJE>. Acesso em: 18/06/2023.

HAARNOJA, T., HA, S., ZHOU, A., et al.. “**Learning to Walk via Deep Reinforcement Learning**”. 2019. Acesso em: 21/03/2022.

HE, K., ZHANG, X., REN, S., et al.. “**Deep Residual Learning for Image Recognition**”. Em: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

HU, H., SONG, S., CHEN, C. L. P. “**Plume Tracing via Model-Free Reinforcement Learning Method**”, *IEEE Transactions on Neural Networks and Learning Systems*, v. 30, n. 8, pp. 2515–2527, 2019.

HUANG, G., LIU, Z., VAN DER MAATEN, L., et al.. “**Densely Connected Convolutional Networks**”. 2018. Acesso em: 12/06/2021.

IMAGENET. “**ImageNet**”. 2021. Disponível em: <https://www.image-net.org/>. Acesso em: 13/10/2022.

JAMES, S., FREESE, M., DAVISON, A. J. “**PyRep: Bringing V-REP to Deep Robot Learning**”, *arXiv preprint arXiv:1906.11176*, 2019.

JPL, N. “**Nebula Initial page**”. 2023. Disponível em: <https://costar.jpl.nasa.gov/>. Acesso em: 04/08/2023.

KIMURA, H., YAMAMURA, M., KOBAYASHI, S. “**Reinforcement learning by stochastic hill climbing on discounted reward**”. Em: *Machine Learning Proceedings 1995*, Elsevier, pp. 295–303, 1995.

KOTSIANTIS, S., ZAHARAKIS, I., PINTELAS, P. “**Machine learning: A review of classification and combining techniques**”, *Artificial Intelligence Review*, v. 26, pp. 159–190, 11 2006. doi: 10.1007/s10462-007-9052-3.

KOWADLO, G., RUSSELL, R. A. “**Robot Odor Localization: A Taxonomy and Survey**”, *The International Journal of Robotics Research*, v. 27, n. 8, pp. 869–894, 2008. doi: 10.1177/0278364908095118. Disponível em: <https://doi.org/10.1177/0278364908095118>.

KRIZHEVSKY, A., SUTSKEVER, I., HINTON, G. E. "**ImageNet Classification with Deep Convolutional Neural Networks**". Em: Pereira, F., Burges, C., Bottou, L., et al. (Eds.), *Advances in Neural Information Processing Systems*, v. 25. Curran Associates, Inc., 2012. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.

LEATHERS, K. W., MICHAELIS, B. T., REIDENBACH, M. A. "**Interpreting the Spatial-Temporal Structure of Turbulent Chemical Plumes Utilized in Odor Tracking by Lobsters**", *Fluids*, v. 5, n. 2, 2020. ISSN: 2311-5521. doi: 10.3390/fluids5020082. Disponível em: <https://www.mdpi.com/2311-5521/5/2/82>.

LECUN, Y., JACKEL, L. D., BOTTOU, L., et al.. "**Comparison of learning algorithms for handwritten digit recognition**". 1995.

LILIENTHAL, A. J., LOUTFI, A., BLANCO, J. L., et al.. "**A rao-blackwellisation approach to GDM-SLAM: Integrating SLAM and gas distribution mapping (GDM)**". Em: *3rd European conference on mobile robots, ECMR'07, Freiburg, Germany, September 19-21, 2017*, pp. 126–131, 2007.

LIN, M., CHEN, Q., YAN, S. "**Network In Network**". Em: *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 91–99, 2013.

MATHWORKS. "**What Is Reinforcement Learning?**" 2021. Disponível em: <https://la.mathworks.com/help/reinforcement-learning/ug/what-is-reinforcement-learning.html>. Acesso em: 29/03/2021.

MITCHELL, T., BUCHANAN, B., DEJONG, G., et al.. "**Machine learning**", *Annual review of computer science*, v. 4, n. 1, pp. 417–433, 1990.

MONROY, J., BLANCO, J. L., GONZÁLEZ-JIMÉNEZ, J. "**Time-Variant Gas Distribution Mapping with Obstacle Information**", *Autonomous Robots*, 04 2015. doi: 10.1007/s10514-015-9437-0.

MONROY, J., HERNANDEZ-BENNETTS, V., FAN, H., et al.. "**GADEN: A 3D Gas Dispersion Simulator for Mobile Robot Olfaction in Realistic Environments**", *Sensors*, v. 17, n. 7, 2017a. ISSN: 1424-8220. doi: 10.3390/s17071479. Disponível em: <https://www.mdpi.com/1424-8220/17/7/1479>.

MONROY, J., HERNANDEZ-BENNETTS, V., FAN, H., et al.. "**GADEN: A 3D Gas Dispersion Simulator for Mobile Robot Olfaction in Realistic Environments**", *Sensors*, v. 17, n. 7, 2017b. ISSN: 1424-8220. doi: 10.3390/s17071479. Disponível em: <https://www.mdpi.com/1424-8220/17/7/1479>.

MURPHY, K. P. *Machine learning: a probabilistic perspective*. MIT press, 2012.

NIU, L., SONG, S., YOU, K. "**A plume-tracing strategy via continuous state-action reinforcement learning**". Em: *2017 Chinese Automation Congress (CAC)*, 2017.

OJEDA, P., MONROY, J., GONZALEZ-JIMENEZ, J. "**An Evaluation of Gas Source Localization Algorithms for Mobile Robots**". Em: *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, APPIS 2020, New York, NY, USA, 2020. Association for Computing Machinery. ISBN: 9781450376303. doi: 10.1145/3378184.3378220. Disponível em: <https://doi.org/10.1145/3378184.3378220>.

PHANITEJA, S., DEWANGAN, P., GUHAN, P., et al.. "**A deep reinforcement learning approach for dynamically stable inverse kinematics of humanoid robots**". Em: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1818–1823, 2017. doi: 10.1109/ROBIO.2017.8324682.

PINCIROLI, C., TRIANNI, V., O'GRADY, R., et al.. "**ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems**", *Swarm Intelligence*, v. 6, n. 4, pp. 271–295, 2012.

POMPY. "**Puff-based odour plume model in Python**". 2018. Disponível em: <https://pompy-docs.readthedocs.io/en/latest/>. Acesso em: 01/04/2021.

PRABOWO, Y. A., RANASINGHE, R., DISSANAYAKE, G., et al.. "**A Bayesian approach for gas source localization in large indoor environments**". Em: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4432–4437, 2020. doi: 10.1109/IROS45743.2020.9341747.

PROGRAMARIA. "**Machine Learning**". 2020. Disponível em: <http://bit.ly/473AejM>. Acesso em: 10/03/2022.

REGGENTE, M., LILIENTHAL, A. "**The 3D-Kernel DM+V/W Algorithm: Using Wind Information in Three Dimensional Gas Distribution Modelling with a Mobile Robot**". 11 2010. doi: 10.1109/ICSENS.2010.5690924.

ROCHA, F., AZPÚRUA, H., GARCIA, G., et al.. "**Análise e Comparação de Mobilidade de um Robô com Locomoção Reconfigurável**". 10 2019. doi: 10.17648/sbai-2019-111280.

ROS. "**Navigation**". 2020. Disponível em: <http://wiki.ros.org/navigation>. Acesso em: 10/03/2022.

SATHLER, M. S. "**Análise e localização de fontes de gás usando aprendizagem por reforço com robôs móveis na indústria da mineração**", 2021. Monografia.

SATHLER, M. S., CID, A. L. M., OLIVEIRA, F. G., et al.. "**Localização de Fontes de Gás Usando Aprendizagem por Reforço com Robôs Móveis na Indústria da Mineração**", *Simpósio Brasileiro de Automação Inteligente*, 2021.

SCHULMAN, J., WOLSKI, F., DHARIWAL, P., et al.. "**Proximal Policy Optimization Algorithms**". 2017. Disponível em: <https://arxiv.org/abs/1707.06347>. Acesso em: 12/06/2021.

SILVER, D., SCHRITTWIESER, J., SIMONYAN, K., et al.. "**Mastering the game of Go without human knowledge**", *Nature*, v. 550, pp. 354–, out. 2017. Disponível em: <http://dx.doi.org/10.1038/nature24270>.

SILVER, D., HUBERT, T., SCHRITTWIESER, J., et al.. "**A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play**", *Science*, v. 362, n. 6419, pp. 1140–1144, 2018. ISSN: 0036-8075. doi: 10.1126/science.aar6404. Disponível em: <https://science.sciencemag.org/content/362/6419/1140>.

SIMONYAN, K., ZISSERMAN, A. "**Very Deep Convolutional Networks for Large-Scale Image Recognition**", *arXiv preprint arXiv:1409.1556*, 2014.

SINGH, S., VAN BREUGEL, F., RAO, R., et al.. "**Understanding biological plume tracking behavior using deep reinforcement-learning**". pp. 750–752, 01 2020.

SPACE. "**Perseverance rover's 1st year on Mars**". 2022. Disponível em: <https://www.space.com/mars-perseverance-rover-ingenuity-helicopter-first-year-photos>. Acesso em: 13/10/2022.

SZEGEDY, C., LIU, W., JIA, Y., et al.. "**Going Deeper with Convolutions**". Em: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.

TEMPO, O. "**Ipatinga: trabalhadores são hospitalizados após vazamento de gás em siderúrgica**". 2022. Disponível em: <https://bit.ly/3GLXusc>. Acesso em: 18/06/2023.

THRUN, S., SCHWARTZ, A. "**Issues in Using Function Approximation for Reinforcement Learning**". Em: *Connectionist Models Summer School*, 1993.

TRANZATTO, M., DHARMADHIKARI, M., BERNREITER, L., et al.. "**Team CERBERUS Wins the DARPA Subterranean Challenge: Technical Overview and Lessons Learned**". 2022. Disponível em: <https://arxiv.org/abs/2207.04914>. Acesso em: 24/01/2023.

VAUGHAN, R. "**Massively multi-robot simulation in stage**", *Swarm intelligence*, v. 2, n. 2-4, pp. 189–208, 2008.

WANG, Z., SCHAUL, T., HESSEL, M., et al.. “**Dueling Network Architectures for Deep Reinforcement Learning**”. 2016. Acesso em: 12/02/2021.

WATKINS, C. J. C. H., DAYAN, P. “**Q-Learning**”, *Machine Learning*, v. 8, n. 3-4, pp. 279–292, 1992. Disponível em: <http://jmvidal.cse.sc.edu/library/watkins92a.pdf>.

WAYMO. “**Waymo Via**”. 2022. Disponível em: <https://waymo.com/waymo-via/>. Acesso em: 13/10/2022.

WONG, C., YANG, E., YAN, X.-T., et al.. “**An overview of robotics and autonomous systems for harsh environments**”. Em: *2017 23rd International Conference on Automation and Computing (ICAC)*, pp. 1–6, 2017. doi: 10.23919/IConAC.2017.8082020.

ZHANG, A., LIPTON, Z. C., LI, M., et al.. “**Dive into Deep Learning**”. 2023. Acesso em: 02/06/2021.