TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG KHÓA 2016

**Trần Quang Minh - 1612374**

# ĐIỀU HƯỚNG KHÔNG BẢN ĐỒ CHO RÔ-BỐT DI ĐỘNG SỬ DỤNG HỌC TĂNG CƯỜNG SÂU

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

**Tp. Hồ Chí Minh, tháng 08/2020**

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN
LỚP CỬ NHÂN TÀI NĂNG KHÓA 2016

**Trần Quang Minh - 1612374**

# ĐIỀU HƯỚNG KHÔNG BẢN ĐỒ CHO RÔ-BỐT DI ĐỘNG SỬ DỤNG HỌC TĂNG CƯỜNG SÂU

## KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN CNTT

GIÁO VIÊN HƯỚNG DẪN
**PGS.TS. Lý Quốc Ngọc**

KHÓA 2016-2020

## NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

TpHCM, ngày .... tháng .... năm ....
Giáo viên hướng dẫn
[Ký tên và ghi rõ họ tên]

## NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Khóa luận đáp ứng yêu cầu của Khóa luận cử nhân CNTT

TpHCM, ngày .... tháng .... năm ....
Giáo viên phản biện
[Ký tên và ghi rõ họ tên]

# Lời cảm ơn

Tôi xin chân thành cảm ơn **PGS.TS. Lý Quốc Ngọc** đã tận tình hướng dẫn trong suốt quá trình hoàn thành khóa luận. Thầy luôn đưa ra những nhận xét, chỉnh sửa và truyền đạt những kiến thức cần thiết để khóa luận có thể thực hiện một cách chặt chẽ nhất.

Tôi xin gửi lời cảm ơn đến các anh trong công ty **AIOZ Việt Nam** đã giúp đỡ và tạo điều kiện trang thiết bị, máy móc hiện đại để tôi có thể hoàn thành tốt khóa luận.

Tôi cũng xin chân thành cảm ơn **Khoa Công Nghệ Thông Tin - Trường Đại Học Khoa Học Tự Nhiên**, đã tạo điều kiện để tôi có thể thực hiện luận văn này.

Cuối cùng, tôi xin dành lời cảm ơn cho gia đình tôi, đã đồng hành và hỗ trợ tôi suốt quãng thời gian đại học.

Khoa Công Nghệ Thông Tin
Lớp cử nhân tài năng khóa 2016

# ĐỀ CƯƠNG CHI TIẾT

| |
|---|
| **Tên Đề Tài:** Điều hướng không bản đồ cho rô-bốt di động sử dụng học tăng cường sâu |
| **Giáo viên hướng dẫn:** PGS. TS. Lý Quốc Ngọc |
| **Thời gian thực hiện:** Tháng 02/2020 - Tháng 08/2020 |
| **Sinh viên thực hiện:** Trần Quang Minh - 1612374 |
| **Loại đề tài:** Nghiên cứu |

**Nội dung đề tài:** Khóa luận thực hiện khảo sát các nghiên cứu giải quyết bài toán điều hướng rô-bốt, các phương pháp học tăng cường sâu, cũng như là các công trình sử dụng học tăng cường sâu giải quyết bài toán trên. Từ đó, xây dựng một hoạch định di chuyển không bản đồ cho rô-bốt di động có thể tự động điều hướng tránh vật cản và về đích an toàn, áp dụng phương pháp học tăng cường sâu với những đóng góp sau:

**Kế thừa:** Phương pháp cơ sở:
- Mô hình hóa bài toán dưới dạng một quá trình quyết định Markov (MDP), với trạng thái tại mỗi thời điểm là n giá trị khoảng cách từ rô-bốt đến các vật xung quanh theo n hướng khác nhau, hành động cần trả về là giá trị vận tốc gốc và vận tốc tuyến tính.
- Thực hiện giải bài toán với thuật toán học tăng cường sâu DDPG.

**Đề xuất:** Sử dụng camera đơn thay thế cảm biến khoảng cách với mục đích tiết kiệm chi phí cho thiết bị ngoại vi, khi mà giá thành cho các thiết bị có khả năng phát laser thu về giá trị khoảng cách thường là cao hơn nhiều so với camera đơn. Từ đó, khóa luận thực hiện thêm những tinh chỉnh để kết quả thu được có thể ngang ngửa hoặc thấp hơn đôi chút so với phương pháp cơ sở:
- Dự đoán ảnh độ sâu kế thừa từ một kiến trúc mạng tương đối gọn nhẹ với độ chính xác cao, đã được huấn luyện sẵn trên KITTI, tập dữ liệu cầu nối giữa thị giác máy tính và robotics.
- Khóa luận làm nhỏ vec-tơ trạng thái bằng cách chỉ lấy mẫu n giá trị độ sâu từ một dòng nhất định của ảnh độ sâu thu được.

**Phần thực nghiệm**
- Thiết kế và xây dựng các kịch bản cho việc học của rô-bốt và tiến hành huấn luyện.
- Thiết lập các môi trường thực tế đánh giá độ hiệu quả của phương pháp cơ sở và đề xuất dựa trên các tiêu chí: tổng thưởng, số bước thực hiện và quãng đường di chuyển.

**Kế hoạch thực hiện:**
- Tháng 02/2020: Thực hiện xây dựng, củng cố và ôn tập kiến thức nền tảng.
- Tháng 03/2020: Tìm kiếm khảo sát các bài toán với từ khóa liên quan, từ đó lựa chọn, phát biểu bài toán và khảo sát các tiến triển của các nghiên cứu trên bài toán đã nêu.
- Tháng 04/2020: Đề xuất phương pháp, cài đặt nhanh và thực hiện vài thực nghiệm nhỏ, củng cố phương pháp.
- Tháng 05-06/2020: Cài đặt phương pháp cơ sở với kết quả tương đương, cài đặt phương pháp đề xuất, lên ý tưởng và thực hiện các thực nghiệm chứng minh hiệu quả của phương pháp đề xuất so với phương pháp cơ sở.

| | |
|---|---|
| - Tháng 07-08/2020: Tổng kết báo cáo, xây dựng demo, tự đánh giá, phản biện. | |
| **Xác nhận của GVHD** | **Ngày ... tháng ... năm ...**<br>**SV Thực hiện** |

# Mục lục

# Danh sách hình

# Danh sách bảng

# Danh mục thuật ngữ - viết tắt

**CNN** Convolution Neural Network - Mạng nơ-ron tích chập. 8

**DDPG** Deep deterministic policy gradient - Thuật toán đạo hàm chính sách tất định sâu. 8

**DQN** Mạng học sâu hàm số Q. 12

**Dyna Q** Phương pháp học kết hợp lên kết hoạch dựa trên hàm số Q. 11

**Dyna Q+** Phương pháp học kết hợp lên kết hoạch dựa trên hàm số Q+. 11

**MDP** Markov Decision Process - Quá trình quyết định Markov. 8

**model-based** Quá trình quyết định có mô hình xác xuất chuyển dịch trạng thái. 11

**model-free** Quá trình quyết định không có mô hình xác xuất chuyển dịch trạng thái. 11

**Monte Carlo Gradient** Phương pháp học xấp xỉ hàm số suy giảm đạo hàm Monte Carlo. 11

**Monte Carlo Learning** Phương pháp học Monte Carlo. 11

**PPO** Proximal Policy Optimization - Thuật toán xấp xỉ tối ưu chính sách. 13

**Q-learning** Phương pháp học hàm số Q. 11

**Rainbow Q-learning** Mạng học sâu hàm số Q cầu vồng. 13

**ROS** Robot Operating System - Hệ điều hành rô-bốt mã nguồn mở. 36

**SAC** Soft Actor-Critic - Thuật toán tác nhân phê bình mềm. 8

**Sarsa** Phương pháp học trạng thái - hành động - phần thưởng - trạng thái - hành động. 11

**Sarsa Semi-Gradient** Phương pháp học xấp xỉ hàm số bán suy giảm đạo hàm trạng thái - hành động. 11

**SLAM** Simultaneous Localization and Mapping - Đồng thời định vị và tạo bản đồ. 5

**SOTA** State-of-the-art - Các thuật toán cho kết quả tốt nhất trên tập dữ liệu. 8

**T3D** Twin Delayed Deep Deterministic Policy Gradient - Thuật toán hai mô hình học trì hoãn đạo hàm chính sách tất định sâu. 13

**TRPO** Trust Region Policy Optimization - Thuật toán tin tưởng miền tối ưu chính sách. 13

**UNet** Mô hình mạng nơ-ron chữ U. 32

# Chương 1

# Giới thiệu

## 1.1 Động lực nghiên cứu

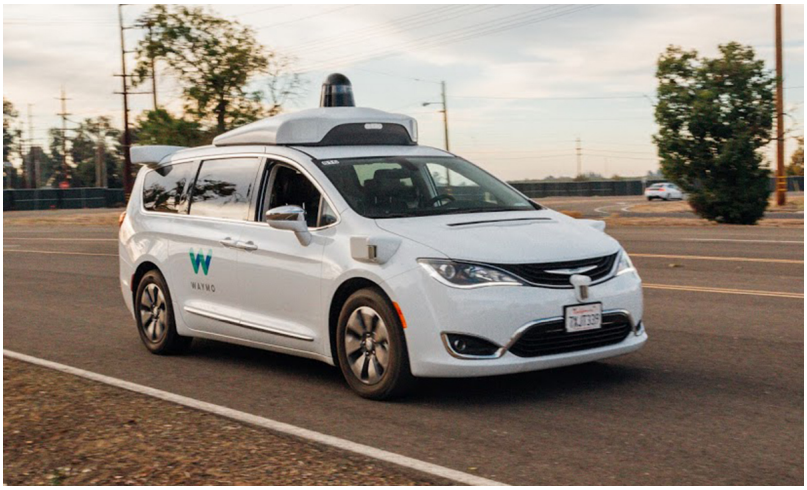
### 1.1.1 Động lực thực tiễn

Ngày nay, sự phát triển mạnh mẽ của các lĩnh vực cơ khí và rô-bốt đã mang lại rất nhiều lợi ích cho con người trong nhiều lĩnh vực khác nhau, đặc biệt là trong công nghiệp. Để có thể tối đa hóa lợi ích của chúng, tự động hoá các tác vụ cho rô-bốt là một trong những nhu cầu cần thiết và cũng đầy tiềm năng. Trong đó, tác vụ điều hướng di chuyển và tránh vật cản cho rô-bốt là một trong những tác vụ được khai thác nhiều, cũng như là tiền đề quan trọng cho các tác vụ khác, hay xa hơn nữa là một rô-bốt tự động hoàn chỉnh.

Thật vậy, dựa trên khả năng tự định hướng di chuyển trong môi trường và tránh được vật cản làm nền tảng, kết hợp với những tác vụ đặc trưng khác cho từng môi trường hoạt động, rô-bốt có thể ứng dụng rộng rãi. Có thể kể đến như việc áp dụng vào các bệnh viện như là một y tá thông minh trong bệnh viện, các nông trại cho các công việc tưới tiêu, thu hoạch, hay trong các nhà kho cho công tác vận chuyển hàng và sắp xếp hàng hóa, v.v. Những điều trên là động lực vô cùng lớn cho nghiên cứu của khóa luận.

Hình 1.1: Tự động hóa áp dụng vào các nông trại

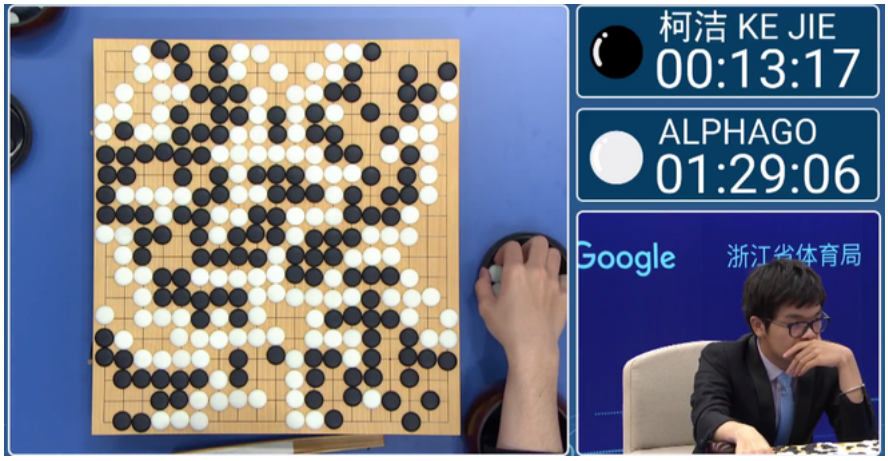
Các hệ tự hành, bán tự hành trên thế giới đang phát triển rất mạnh mẽ. Ví dụ như hệ thống hỗ trợ người lái Autopilot (Tesla), các dòng xe của họ cũng được trang bị một cách mạnh mẽ các hệ thống cảm biến tối tân nhất, sẵn sàng cho một tương lai xe tự hành phổ biến. Một ví dụ khác, xe tự hành Waymo (Google) cũng tạo nên những động lực to lớn cho phát triển hệ tự hành. Thật vậy, Waymo có khả năng hoàn thành một lộ trình di chuyển từ A đến B một cách an toàn, trong đó có việc tránh được trẻ em, người đi bộ, hay các phương tiện khác trên đoạn đường hẹp hay góc cua khó.



Hình 1.2: Xe tự hành Waymo của Google

### 1.1.2 Động lực khoa học

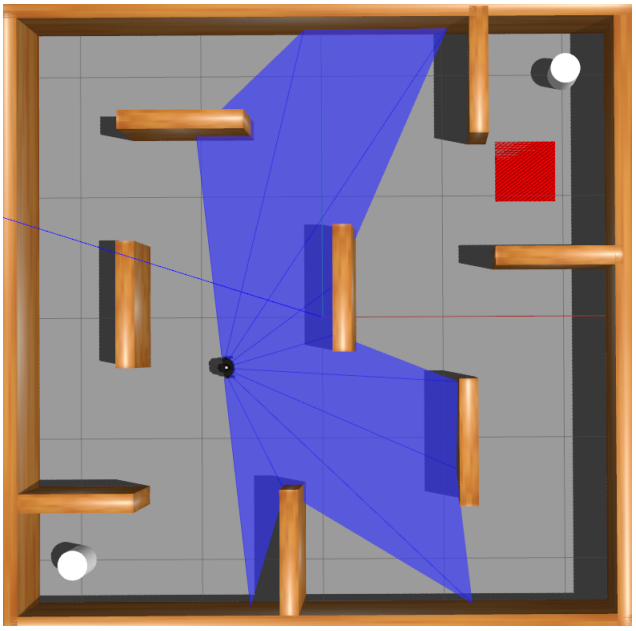
Sự phát triển mạnh mẽ của học sâu với khả năng tính toán từ các thiết bị phần cứng hỗ trợ tính toán song song, giải quyết được rất nhiều bài toán ở các lĩnh vực khác nhau như thị giác máy tính hay xử lý ngôn ngữ tự nhiên. Trong đó, phải kể đến những thành tựu của học tăng cường sâu khi tạo được những "bộ não" nhân tạo, đánh bại các kiện tướng ở các môn cờ. AlphaGo của DeepMind năm 2017 đánh bại cờ thủ số 1 thế giới ở bộ môn cờ vây, hay Alpha Zero cũng của chính DeepMind đạt được những kết quả ấn tượng trong bộ môn cờ vua. Kế thừa sự phát triển đó, học sâu nói chung và học tăng cường sâu nói riêng đang được nghiên cứu rộng rãi cho bài toán tự hành. Đây là một động lực mạnh mẽ cho khóa luận tìm hiểu, nghiên cứu và có những đóng góp.




Hình 1.3: Ván cờ của AlphaGo và kỳ thủ cờ vây số 1 thế giới năm 2017

## 1.2 Phát biểu bài toán

Khóa luận hướng đến việc thiết kế một hoạch định di chuyển không có bản đồ cho rô-bốt di động, có thể điều hướng và tránh vật cản tại mọi thời điểm cho đến khi về đích.

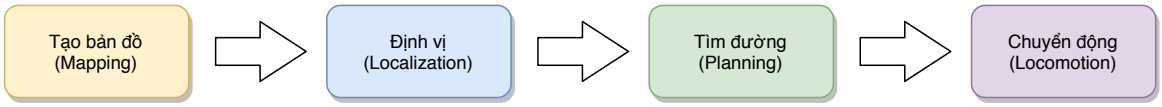


Hình 1.4: Rô-bốt (màu đen) điều hướng di chuyển tránh vật cản và tìm đường về đích (đỏ)

Bài toán được khóa luận giải quyết theo hướng tiếp cận không có bản đồ, tức là sẽ không có thông tin về hệ tọa độ của môi trường, cụ thể là vị trí tọa độ của rô-bốt, các vật thể và điểm đích. Thay vào đó, là việc chỉ dựa trên những gì rô-bốt quan sát được, từ đó đưa ra hành động tương ứng phù hợp. Cụ thể:

**Đầu vào**: Các giá trị thu được từ các cảm biến ngoại vi của rô-bốt. Ví dụ như, các giá trị cảm biến khoảng cách từ rô-bốt đến các vật thể xung quanh, hay các loại hình ảnh từ camera của rô-bốt,...
**Đầu ra**: Rô-bốt có thể tự động đưa ra những hành động cụ thể tại mỗi thời điểm để điều hướng và về đích an toàn.

Dạng bài toán này thường được chia nhỏ thành các mô-đun con bao gồm: tạo bản đồ, định vị, tìm đường đi và cuối cùng là thực hiện chuyển động. Trong đó, ở bước đầu tiên, rô-bốt sẽ cố gắng tái tạo bản đồ của môi trường xung quanh nó từ các cảm biến ngoại vi. Tiếp đến là công tác định vị vị trí của rô-bốt, vật cản, điểm đích..., Từ đó, tìm ra đường đi tối ưu và thực hiện những chuyển động phù hợp với đường đi đã đặt ra. Việc giải quyết các mô-đun này có thể diễn ra độc lập hoặc là kết hợp. Ví dụ, các thuật toán SLAM thực hiện mô-đun tạo bản đồ bằng việc thực hiện đồng thời ánh xạ bản đồ và thông tin vị trí của rô-bốt, hay các phương pháp dựa trên việc học kết hợp các mô-đun thành một "hộp đen" và tối ưu nó dựa trên quá trình thử và sai.




Hình 1.5: Các mô-đun cho bài toán điều hướng không bản đồ

## 1.3 Các thách thức của bài toán

Điều hướng di chuyển của rô-bốt và tránh vật cản mà không có bản đồ là chủ đề đã được phát triển một thời gian dài, đi kèm với đó là rất nhiều thách thức được đặt ra xuyên suốt quá trình này. Bên cạnh đó, với việc khóa luận tập trung nghiên cứu các công trình ứng dụng học tăng cường sâu cho bài toán này, những khó khăn mà phương pháp này gặp phải cũng sẽ được trình bày trong mục này.

### 1.3.1 Môi trường hoạt động của rô-bốt

Với hệ tự hành nói chung và rô-bốt nói riêng, môi trường hoạt động cho chúng là vô cùng lớn. Không những vậy, môi trường này là môi trường động, liên tục thay đổi, các tác nhân, trạng thái xảy đến một cách ngẫu nhiên, từ các môi trường trong nhà như căn hộ, văn phòng,... cho đến ngoài trời như sân bay, ga tàu, trung tâm thương mại, các nhà chứa hàng hay các loại hình giao thông như đường bộ, trên không, dưới nước, v.v

Đối với phương pháp dựa trên việc học, điều này càng gây ra nhiều khó khăn hơn. Tính tổng quát của mô hình học là một yếu tố như vậy. Làm thế nào để vừa đảm bảo được mô hình có thể phân biệt rõ ràng được các trạng thái để đưa ra quyết định phù hợp, nhưng cũng đồng thời phải tổng quát hóa được những quan sát, từ đó có thể thích ứng với những sự thay đổi trong môi trường động. Bên cạnh đó, vẫn đạt được hiệu quả khi được đưa vào một môi trường hoàn toàn khác so với môi trường mà rô-bốt đã được chuẩn bị và quen thuộc.

### 1.3.2 Thu thập dữ liệu đối với các phương pháp học

Với lượng môi trường cho rô-bốt hoạt động là rất lớn thì việc thu thập các dữ liệu này thực sự là một thách thức lớn. Công việc này thường được thực hiện trên môi trường giả lập, bởi lẽ việc làm này giúp tiết kiệm chi

phí trong việc thiết lập các môi trường và các kịch bản huấn luyện. Ví dụ như các trường hợp như rô-bốt ngã, lật, bị cán nát hoặc rớt xuống vực sâu, sẽ dễ dàng được thực hiện hơn.

Tuy nhiên, khó khăn ở đây chính là sự chuyển tiếp các mô hình học từ giả lập sang thực tế. Sự khác biệt về phần cứng cũng như là các cơ chế vật lý giữa thực tế và giả lập là hiện hữu. Các vật thể trên môi trường giả lập cũng thường khó đạt được độ chính xác như thực tế. Điều tương tự cũng đúng với độ chính xác của các thiết bị cảm biến ngoại vi. Dẫn đến, hai trạng thái có thể xem là như nhau giữa giả lập và thực tế nhưng lại được mô hình quan sát có chút khác nhau giữa 2 môi trường. Điều này càng làm tăng thêm khó khăn cho bài toán tổng quát hóa mô hình cho các trạng thái đã đề cập ở trên. Phải làm thế nào để xây dựng các kịch bản huấn luyện một cách có chọn lọc và phù hợp, song song với đó là xây dựng các nền tảng giả lập với cơ chế vật lý chính xác hơn, các vật thể gần gũi hơn với môi trường thực tế.

### 1.3.3 Các thiết bị cảm biến

Các thiết bị cảm biến là yếu tố quan trọng nhất cho rô-bốt có khả năng tự động hóa, đặc biệt là các cảm biến ngoại vi. Có thể kể đến các cảm biến khoảng cách quét toàn bộ không gian 360 độ và xa đến 20-30m với khoảng cách giữa các tia là rất nhỏ, hay những loại camera cung cấp rất nhiều thông tin khác nhau không chỉ là ảnh với độ phân giải cao mà còn là biểu đồ nhiệt, ảnh độ sâu. Các thiết bị này tạo điều kiện lớn cho các mô-đun tạo bản đồ và định vị hoạt động hiệu quả, từ đó làm tiền đề cho công tác tìm đường và chuyển động theo sau.

Mặt khác, để có thể đưa tự động hóa gần hơn với thực tiễn thì khả năng sản xuất hàng loạt cũng là một yếu tố then chốt. Các hướng nghiên cứu hiện nay cũng tập trung dựa vào khai thác tối đa thông tin từ các cảm

biến giá rẻ. Có thể kể đến các cảm biến khoảng cách thưa, ngắn, các ảnh phân giải thấp từ camera đơn. Các cảm biến giá rẻ cũng đi kèm với việc thông tin trạng thái bị mất mát đáng kể. Do vậy mà các thách thức là rất nhiều tuy nhiên tiềm năng là vô cùng lớn.

## 1.4 Đóng góp

Với bài toán phát biểu ở trên, khóa luận lựa chọn giải quyết bằng cách áp dụng học tăng cường sâu dựa trên phương pháp của nghiên cứu [46]. Trên cơ sở đó, thực hiện tìm hiểu một trong các thuật toán SOTA để giải bài toán tối ưu trong giai đoạn huấn luyện, trên các MDP có miền không gian hành động liên tục, là SAC [11]. Phương pháp này cải tiến trực tiếp từ thuật toán DDPG của phương pháp [46] và được [3] áp dụng cho bài toán tương tự.

Tiếp theo, cũng là đóng góp chính của khóa luận, đó là hướng đến việc sử dụng ảnh từ một camera đơn, thay vì sử dụng cảm biến khoảng cách. Mục đích chính của việc thay thế này là nhằm tiết kiệm chi phí cho thiết bị ngoại vi, khi mà giá thành cho các thiết bị có khả năng phát laser thu về giá trị khoảng cách thường là cao hơn nhiều so với camera đơn. Thật vậy, đã có những nghiên cứu [50], [23] cũng sử dụng mạng CNN để suy ra ảnh độ sâu từ ảnh RGB, và sử dụng nó như là trạng thái đầu vào của rô-bốt. Tuy nhiên thay vì sử dụng cả ảnh này, khóa luận làm nhỏ vec-tơ trạng thái bằng cách chỉ lấy mẫu $n$ giá trị độ sâu từ một dòng nhất định của ảnh. Công tác dự đoán độ sâu được kế thừa từ một kiến trúc mạng [10], đã được huấn luyện sẵn trên KITTI [9], tập dữ liệu cầu nối giữa thị giác máy tính và robotics. Từ đó, khóa luận thực hiện thêm những tinh chỉnh để kết quả thu được có thể ngang ngửa hoặc thấp hơn đôi chút so với phương pháp sử dụng giá trị khoảng cách từ cảm biến.

# Chương 2

# Các công trình liên quan

## 2.1 Bài toán điều hướng rô-bốt

Đã có rất nhiều công trình nghiên cứu về bài toán điều hướng rô-bốt, trải đều trên nhiều phương pháp khác nhau với những thành tựu nhất định. Đầu tiên phải kể đến các phương pháp dựa trên bản đồ [1], [2], cách làm này yêu cầu bản đồ cho trước của môi trường, từ đó đưa ra các quyết định nhằm điều hướng hành vi của rô-bốt. Tuy nhiên, không phải môi trường nào cũng có sẵn bản đồ mô tả, việc tái tạo bản đồ cũng có khá nhiều nghiên cứu tiếp cận [36], [49], [5], [31]. Mặc dù vậy, xây dựng bản đồ đòi hỏi rất nhiều độ chính xác và chi phí, để đảm bảo các thuật toán tìm đường, các chuyển động của rô-bốt có thể áp dụng được một cách hiệu quả nhất.

Theo đó, các thuật toán không dựa trên bản đồ cũng rất phổ biến với các mô-đun như hình 1.5. SLAM [7] là một phương pháp đồng thời kết hợp việc tái tạo bản đồ và định vị vị trí, từ đó thiết lập hoạch định di chuyển. Điểm mạnh của phương pháp là việc rô-bốt tự tái tạo bản đồ bằng sức mạnh của các thiết bị cảm biến ngoại vi như các loại LiDAR, Radar... Từ đó, việc rô-bốt chuyển động trong môi trường mà chính nó tạo ra, theo thuật toán tìm đường sẽ không gặp nhiều khó khăn. Có vài

thách thức đối với phương pháp này, có thể kể đến là sự tiêu tốn thời gian trong việc xây dựng và cập nhật bản đồ, cũng như là đòi hỏi về mặt phần cứng (các cảm biến ngoại vi) cần đạt được độ chính xác cao. Một vài công trình tương tự trong đó cố gắng tiết kiệm chi phí bằng việc sử dụng định vị wi-fi [38] hay phương pháp dựa trên sự giao tiếp ánh sáng nhìn thấy [29].

Để giảm vấn đề kinh phí yêu cầu phần cứng, những nghiên cứu về điều hướng rô-bốt chỉ dựa trên thông tin thị giác từ máy ảnh được quan tâm nghiên cứu nhiều hơn và cũng đem lại nhiều triển vọng, các phương pháp tập trung vào việc tránh vật cản với ảnh đầu vào cho trước có thể kể đến như [12], [21], [30], [33]. Song song với đó, để giảm các yêu cầu về tinh chỉnh và phụ thuộc vào các kiến thức chuyên ngành, các phương pháp kết hợp các mô-đun thành một "hộp đen", ánh xạ trực tiếp từ trạng thái quan sát sang chuyển động của rô-bốt dựa trên các thuật toán học. Thật vậy, với việc học sâu có những tiềm năng giải quyết tốt các bài toán xấp xỉ, [4] áp dụng để trích xuất thông tin ngữ nghĩa từ ảnh camera để quyết định hành động như rẽ trái, rẽ phải,... Hay [28] sử dụng mô hình học sâu để tìm ra hành động tương ứng từ cảm biến khoảng cách và vị trí điểm đích. Trong các phương pháp học cho bài toán điều hướng không bản đồ, học tăng cường sâu đang nhận được rất nhiều sự quan tâm khi các môi trường mà nó giải quyết và bài toán điều hướng rô-bốt có nhiều sự tương đồng. Theo đó, tiếp đến khóa luận sẽ lần lượt trình bày các khảo sát trên học tăng cường sâu, và học tăng cường sâu cho chính bài toán điều hướng không bản đồ.

## 2.2 Học tăng cường sâu

### 2.2.1 Học tăng cường

Các phương pháp giải bài toán học tăng cường cũng như các phân loại của nó là rất đa dạng. Tùy vào chu trình có trạng thái kết thúc hay không

mà các tác vụ được chia thành tác vụ có kết thúc và tác vụ liên tục. Bên cạnh đó, nếu xét về mục đích đánh giá hoặc tối ưu chính sách, ta có các tác vụ dự đoán và các tác vụ điều khiển. Đối với phương pháp, xét một MDP đã biết trước mô hình chuyển dịch trạng thái, dựa trên phương pháp quy hoạch động [39], ta có thuật toán đánh giá duyệt chính sách để giải quyết các bài toán dự đoán. Hơn thế nữa, với việc áp dụng xen kẽ thuật toán này cùng với lý thuyết cải thiện chính sách, mô hình duyệt chính sách tổng quát ra đời nhằm giải quyết các bài toán điều khiển. Tuy nhiên, trong thực tế không phải lúc nào mô hình chuyển dịch trạng thái cũng được biết trước, khi đó các thuật toán học dựa trên việc lấy mẫu từ môi trường và trải qua quá trình thử và sai, tiêu biểu là các thuật toán như Monte Carlo Learning [40], Sarsa, Q-learning [41]. Các thuật toán này còn được gọi là thuật toán model-free, khi nó giải quyết các bài toán dự đoán và điều khiển mà không quan tâm đến mô hình chuyển dịch trạng thái. Ngược lại, ta cũng có các thuật toán model-based, trong đó ưu tiên việc học mô hình này, có thể kể đến các thuật toán như Dyna Q, Dyna Q+ [42].

Các thuật toán trên lưu trữ truyền thống dưới dạng bảng các giá trị phần thưởng kỳ vọng của mỗi trạng thái và cập nhật chúng trong quá trình học. Tuy nhiên, số lượng trạng thái càng lớn thì càng tốn không gian lưu trữ. Khi đó các giá trị này sẽ được xấp xỉ bằng các hàm số tuyến tính hoặc phi tuyến với các bộ tham số được học theo thời gian và đầu vào là trạng thái tương ứng [43]. Không chỉ giải quyết vấn đề không gian lưu trữ, cách làm này còn giải quyết vấn đề tổng quát hóa. Thật vậy, cách dùng bảng phân biệt rất rõ ràng các trạng thái với nhau, thế nên các trạng thái dù chỉ khác nhau ở một hoặc vài đặc trưng cũng được cho là khác nhau hoàn toàn. Việc này làm mất thời gian tính toán khi có một trạng thái mới xuất hiện. Xấp xỉ hàm số cân bằng tốt hơn vấn đề tổng quát hóa - phân biệt hóa trong học máy nói chung và học tăng cường nói riêng. Các thuật toán được biến đổi phù hợp với cách dùng này như Monte Carlo Gradient, Sarsa Semi-Gradient,...

Không chỉ vậy, việc tham số hóa cũng áp dụng với chính sách và việc tối ưu chính sách, được mở đường bởi lý thuyết chính sách đạo hàm [44], cách làm này đem đến nhiều lợi thế so với các phương pháp trước như tự động hội tụ về chính sách tất định tối ưu, chính sách ngẫu nhiên và tăng tốc việc học và độ chính xác. Kết hợp việc tối ưu chính sách và giá trị phần thưởng kỳ vọng của trạng thái, ta có mô hình tác nhân - phê bình [19] giải quyết các bài toán toàn diện và hiệu quả hơn.

### 2.2.2 Học tăng cường sâu

Học sâu phát triển mạnh mẽ trong thập kỷ vừa qua với bước đột phá AlexNet [20] trên tập dữ liệu Image Net [6], kéo theo sự ra đời của hàng loạt các kiến trúc mạng học sâu mới [45], [37], [13]. Bên cạnh đó với sự ra đời của các kỹ thuật tối ưu mới [32] cùng với các phương pháp học chuyển đổi [47], và sự phát triển của phần cứng máy tính hỗ trợ việc tính toán song song đã mang đến sự thành công cho học sâu.

Kế thừa những thành tựu của học sâu, khi nó giải quyết tốt bài toán xấp xỉ hàm số, học tăng cường sâu cũng ra đời với những công trình nghiên cứu xuất sắc. Bắt đầu với DQN [26], thuật toán sử dụng mạng nơ-ron để xấp xỉ hàm số Q, kết hợp với bộ nhớ phát lại, lưu trữ các chuyển dịch trạng thái ở các thời gian khác nhau. Phương pháp này là phiên bản nâng cấp dựa trên cách học ngoại chính sách xuất phát từ Q-learning, trong đó chính sách điều khiển được tối ưu khác với chính sách điều khiển biểu diễn dùng để thu thập các chuyển dịch trạng thái trong quá trình học. Ưu điểm chính của phương pháp học ngoại chính sách hay Q-learning đến từ việc nó không cần toàn bộ tiến trình, từ trạng thái bắt đầu đến kết thúc, đồng thời có thể sử dụng lại các chuyển dịch trạng thái trong quá khứ dựa vào bộ nhớ phát lại, từ đó đem lại hiệu quả trong việc lấy mẫu. Một lợi thế khác đó là vì chính sách cần tối ưu khác với chính sách biểu diễn sẽ giúp

thuật toán duy trì khám phá, giải quyết tốt vấn đề đánh đổi khai thác - khám phá trong học tăng cường.

Tuy nhiên, điểm yếu của các phương pháp học ngoại chính sách là sự thiếu ổn định và khó khăn trong việc hội tụ, trong khi đó, các phương pháp tối ưu trên chính sách tối ưu chính xác những gì ta muốn, đó là giá trị phần thưởng trả về kỳ vọng và cập nhật trực tiếp lên chính sách điều khiển. Điều này khiến các thuật toán dạng này tin tưởng và ổn định hơn. Thuật toán Trust Region Policy Optimization (TRPO)[35] ra đời, kết hợp tối ưu chính sách truyền thống và mô hình tác nhân - phê bình, tuy nhiên thay đổi cách cập nhật tham số của chính sách so với thông thường bằng việc giải một bài toán tối ưu, với ràng buộc sự thay đổi các chính sách được đo bằng phân kỳ Kullback Leibler, cho phép các bước huấn luyện lớn hơn và tăng tốc quá trình hội tụ. Bằng việc cải tiến TRPO, Proximal Policy Optimization (PPO) [34] chỉ ra những khó khăn trong việc cài đặt TRPO bằng việc sử dụng một hàm lỗi thay thế, hỗ trợ tăng tốc hội tụ. Các thuật toán trên chính sách mạnh về tính ổn định nhưng lại thiếu đi hiệu quả trong việc lấy mẫu khi cần toàn bộ tiến trình từ trạng thái bắt đầu đến kết thúc khiến làm chậm quá trình hội tụ.

Để cân bằng điểm mạnh và điểm yếu của hai cách tiếp cận đã đề cập bên trên, các thuật toán ra đời đồng thời tối ưu trực tiếp chính sách di chuyển và xấp xỉ hàm số giá trị hành động dựa trên đẳng thức Bellman. Các thuật toán này thường giải quyết bài toán trên môi trường có miền hành động liên tục. Deep Deterministic Policy Gradient (DDPG) [22], Rainbow Q-learning [15], T3D [8], SAC [11] là các thuật toán như vậy.

## 2.3 Học tăng cường (sâu) cho bài toán điều hướng rô-bốt

Học tăng cường được áp dụng rộng rãi cho rô-bốt nói chung và bài toán điều hướng rô-bốt nói riêng. [18] sử dụng hướng tiếp cận tối ưu chính sách cho bài toán học tăng cường để điều khiển sự vận động của rô-bốt bốn chân. Cũng với phương pháp tương tự, [27] bàn về bài toán chuyển động nguyên thủy. [24] áp dụng thuật toán model-based cho bài toán tránh vật cản với ảnh camera đơn hay [16] ứng dụng học tăng cường để tự động hóa trực thăng.

Đối với bài toán điều hướng di chuyển không bản đồ, học tăng cường được sử dụng khá phổ biến. [46] sử dụng thuật toán DDPG giải quyết bài toán tìm đường về đích và tránh vật cản cho rô-bốt di động với cảm biến khoảng cách giá rẻ và có thể tích hợp vào môi trường thực tế. Cũng giải quyết bài toán trên rô-bốt di động với cảm biến ngoại vi giá rẻ (camera đơn), [50] dựa vào thông tin độ sâu của ảnh được học được từ một mô hình mạng tích chập, kết hợp với DQN được cải tiến, huấn luyện trên nhiều môi trường giả lập từ đơn giản đến phức tạp rồi đưa vào thực tế. Về phần các phương pháp tối ưu trên chính sách, [51] sử dụng mô hình tác nhân-phê bình kết hợp với thông tin ảnh đích như là một đầu vào ngầm định khiến cho phương pháp trở nên tổng quát hóa với những loại môi trường tương tự mà không cần phải huấn luyện lại từ đầu. [25] cũng sử dụng phương pháp tối ưu trên chính sách với việc ước tính ảnh độ sâu song song với học có giám sát huấn luyện trong môi trường giả lập phức tạp và toàn diện hơn. [23], [48] với thuật toán tối ưu trên chính sách PPO cũng có những cách tiếp cận tương tự.

# Chương 3

# Phương pháp

## 3.1 Phương pháp cơ sở

Khóa luận sử dụng phương pháp được đề cập trong nghiên cứu [46] để làm phương pháp cơ sở, từ đó bố trí các thực nghiệm để qua đó đánh giá được phương pháp đề xuất của khóa luận. Là một phương pháp dựa trên việc học, [46] kết hợp các mô-đun 1.5 thành một "hộp đen" duy nhất. Theo đó, ánh xạ trực tiếp trạng thái của môi trường sang chuyển động của rô-bốt tại mỗi thời điểm với mục tiêu về đích an toàn. Để làm được điều này, bài toán được chia làm 2 giai đoạn: giai đoạn huấn luyện, nơi mà rô-bốt được hướng dẫn rằng nên di chuyển như thế nào từ trạng thái quan sát được bởi những phần thưởng phù hợp với hành vi đó, và giai đoạn thực thế, để xem rằng liệu những gì rô-bốt được học có hoạt động hay không.

### 3.1.1 Các khái niệm cơ bản trong học tăng cường

Bài toán ở giai đoạn huấn luyện được mô tả và giải quyết dưới dạng một bài toán học tăng cường. Một cách bao quát, học tăng cường là sự nghiên cứu về các chủ thể (là những đối tượng có khả năng thực hiện những hành động cụ thể nào đó để đạt được một mục đích nhất định, trong trường hợp của khóa luận là rô-bốt di động) và cách chúng học bằng việc trải qua thử

và sai. Học tăng cường hình thức hóa ý tưởng về việc thưởng và phạt các chủ thể cho những hành vi mà chúng thực hiện, mà theo đó, chúng sẽ lặp lại hoặc không thực hiện những hành vi tương tự trong tương lai.

### Môi trường

Môi trường là nơi mà chủ thể thuộc về và có những tương tác với. Tại mỗi thời điểm mà chủ thể tương tác với môi trường, nó sẽ quan sát được một trạng thái của môi trường đó, và tiếp đến đưa ra quyết định để thực hiện hành động nào đó. Môi trường có thể thay đổi nếu chủ thể tác động lên nó hoặc chính nó cũng tự thay đổi theo thời gian.

Hình 1.4 là một dạng môi trường, bao gồm rô-bốt, các bức tường, vật cản và điểm đích. Khi rô-bốt, chạm vào vật cản, vị trí vật cản có thể bị thay đổi. Hay theo thời gian, các vật cản cũng có thể tự dịch chuyển vị trí, trở thành những vật cản động và gây ra những khó khăn cho rô-bốt.

### Hành động

Những kiểu môi trường khác nhau cho phép các hành động khác nhau. Tập hợp các giá trị của hành động thỏa điều kiện môi trường được gọi là không gian hành động. Trong một vài môi trường, ta có không gian hành động là hữu hạn, nơi mà chỉ có một số lượng hành động nhất định cho chủ thể. Ví dụ như việc rô-bốt chỉ được phép có 4 hành động bao gồm quay trái, quay phải, quay ra sau và tiến đến phía trước với vận tốc 0.1m/s chẳng hạn. Ngược lại, trong những môi trường khác, không gian hành động là liên tục, ví dụ như trong trường hợp của khóa luận.

Thật vậy, hành động của rô-bốt ở đây được kí là $a$, là một véc-tơ 2 chiều gồm 2 giá trị vận tốc tuyến tính $v$ (m/s), thể hiện tốc độ tịnh tiến về phía trước, và vận tốc xoay $\omega$ (rad/s), nhằm điều chỉnh hướng di chuyển của rô-bốt. Trong đó, $v$ thuộc tập các số thực trong $[0, 1]$ còn $\omega$ thuộc tập

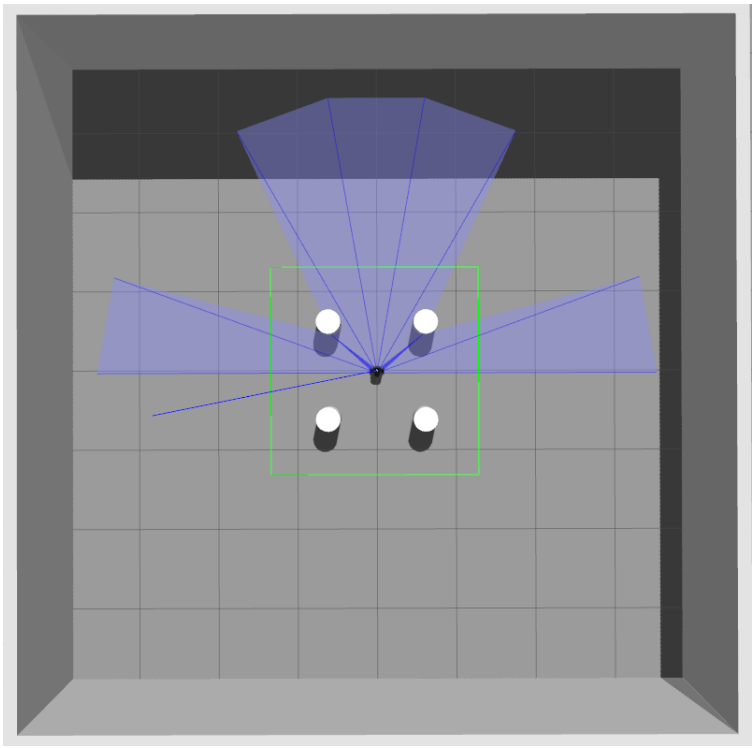
các số thực trong $[-0.5, 0.5]$.

### Trạng thái và quan sát

Một trạng thái mô tả môi trường tại một thời điểm nào đó. Nó có thể bao gồm các thông tin như vị trí các bức tường, các vật cản, điểm đích cũng như là màu sắc, hình dáng hay kích thước của chúng. Trong khi đó, một quan sát, thường chỉ là 1 phần nhỏ của trạng thái được quan sát bởi chủ thể, ví dụ như là đó chỉ là hình ảnh từ camera của rô-bốt chẳng hạn, với rất nhiều thông tin bị bỏ sót.

Trong trường hợp của phương pháp cơ sở [46], quan sát có được từ rô-bốt $o_t$ là $n$ giá trị cảm biến khoảng cách từ $n$ hướng cách đều nhau trong khoảng từ -90 đến 90 độ.

$$o_t = [x_1, x_2, ..., x_n] \tag{3.1}$$



Hình 3.1: 10 tia laser được bắn ra từ rô-bốt cách đều nhau trong khoảng từ -90 đến 90 độ trả về các giá trị khoảng cách của rô-bốt so với các vật thể xung quanh

Trong khi đó, trạng thái trong trường hợp này $s_t$ bao gồm $o_t$, thêm vào đó là véc-tơ $p_t$ gồm 2 giá trị thể hiện vị trí tương đối của rô-bốt so với điểm đích (bao gồm khoảng cách (m) và góc (rad)) và hành động của rô-bốt tại thời điểm trước đó $a_{t-1}$

$$s_t = [o_t, p_t, a_{t-1}] \tag{3.2}$$

### Chính sách

Chính sách là những luật, với đầu vào là trạng thái $s$, được sử dụng bởi chủ thể để quyết định xem hành động nào nên được thực hiện. Nói cách khác, chính sách có thể coi là "bộ não" của chủ thể. Các chính sách thường được chia làm 2 loại là tất định và ngẫu nhiên. Chính sách ngẫu nhiên thường được định kí hiệu là $\pi$

$$a \sim \pi(.|s) \tag{3.3}$$

, công thức trên chỉ việc hành động sẽ được lấy mẫu từ phân phối xác trên tập không gian hành động. Trong khi đó, chính sách tất định được kí hiệu là $\mu$

$$a = \mu(s) \tag{3.4}$$

, thể hiện việc hành động được suy ra trực tiếp từ chính sách.

Các thuật toán khác nhau sử dụng các loại chính sách khác nhau từ 1 trong 2 loại trên. Ngoài ra, trong học tăng cường, tham số hóa chính sách khá phổ biến: các giá trị hành động đầu ra phụ thuộc vào 1 tập các tham số. Khi đó, ta có thể điều chỉnh các hành động này thông qua các thuật toán tối ưu. Các tham số hóa chính sách với bộ tham số $\theta$ thường được kí hiệu là $\pi_\theta$ hoặc $\mu_\theta$.

### Chu trình, phần thưởng và tổng thưởng

Chu trình là chuỗi các trạng thái và hành động từ bắt đầu đến kết thúc

$$\tau = (s_0, a_0, s_1, ..., s_{T-1}, a_{T-1}, s_T), \tag{3.5}$$

với $s_0$ là trạng thái của môi trường khi bắt đầu. Đây là thời điểm rô-bốt sẽ được khởi tạo tại một vị trí bất kì trong môi trường với xung quanh là các vật thể cố định và điểm đích. Trong khi đó, $s_T$ là trạng thái tại thời điểm kết thúc, bao gồm việc rô-bốt di chuyển chạm vật cản hoặc về đến đích.

Việc môi trường có sự thay đổi trong trạng thái giữa 2 thời điểm $t$ và $t+1$ được gọi là một chuyển dịch trạng thái. Chuyển dịch trạng thái này chỉ phụ thuộc vào trạng thái và hành động gần nhất, mà không quan tâm gì đến các trạng thái và hành động trong quá khứ. Tính chất này còn được gọi là tính chất Markov và cả chu trình được gọi là quá trình quyết định

Markov (Markov Decision Process - MDP)

$$s_{t+1} = P(.|s_t, a_t) \tag{3.6}$$

Phần thưởng, kí hiệu là $r$ là một khái niệm quan trọng trong học tăng cường. Nó là một giá trị vô hướng thường phụ thuộc vào trạng thái hiện tại $s_t$ của môi trường, hành động vừa thực hiện $a_t$ và trạng thái kế tiếp $s_{t+1}$, có thể biểu diễn một cách tổng quát bằng một hàm số $R$ như sau.

$$r = R(s_t, a_t, s_{t+1}) \tag{3.7}$$

Trong trường hợp của khóa luận, phần thưởng được định nghĩa cụ thể hơn như sau:

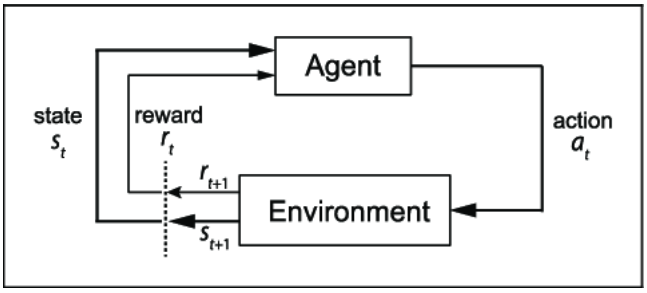
$$r = R(s_t, a_t, s_{t+1}) = \begin{cases} r_{arrive}, & \text{if } d_{t+1} \leq c_d \\ r_{collision}, & \text{if } min_{x_{t+1}} < c_o \\ c_r(d_t - d_{t+1}), & \text{cho các trường hợp còn lại} \end{cases} \tag{3.8}$$

Trong đó, $r_{arrive}$ là phần thưởng đạt được khi khoảng cách từ rô-bốt đến điểm đích $d_{t+1}$ nhỏ hơn ngưỡng cho trước $c_d$, $r_{collision}$ là phần thưởng âm phải nhận khi giá trị nhỏ nhất trong các biến phạm vi lấy được từ cảm biến $min_{x_{t+1}}$ nhỏ hơn ngưỡng va chạm cho trước $c_o$. Nếu không là hai trường hợp trên, phần thưởng đạt được sẽ là hiệu của khoảng cách đến điểm đích ở thời điểm trước và khoảng cách đến điểm đích ở thời điểm sau nhân với một siêu tham số cần tinh chỉnh $c_r$, phần thưởng này sẽ là âm nếu khoảng cách từ rô-bốt đến điểm đích không được thu ngắn theo thời gian. Các giá trị ngưỡng cụ thể sẽ được trình bày trong phần thực nghiệm.

Tổng thưởng, ký hiệu là $G$, là tổng các phần thưởng $r_t$ trong chu trình

$$\tau = (s_0, a_0, r_1, s_1, a_1, ..., s_{T-1}, a_{T-1}, r_T, s_T), \tag{3.9}$$

$$G = \sum_{t=0}^{T} r_t \tag{3.10}$$




Hình 3.2: Sơ đồ mô tả một quá trình quyết ngẫu nhiên Markov của rô-bốt với môi trường

### Hàm số giá trị trạng thái - hành động

Mục đích của hàm số này là để đánh giá một chính sách $\mu_\theta$ có tốt hay không. Thật vậy, cho các chu trình bắt đầu bằng trạng thái $s$ và hành động $a$, và phần còn lại của chu trình được sinh ra theo chính sách $\mu_\theta$.

$$\tau = (s, a, r_1, s_1, \mu_\theta(s_1), ..., s_{T-1}, \mu_\theta(s_{T-1}), r_T, s_T), \tag{3.11}$$

ta có $Q_{\mu_\theta}(s, a)$ đánh giá chính sách $\mu_\theta$ bằng việc tính tổng thưởng kỳ vọng của các chu trình được sinh ra bởi chính sách này.

$$Q_{\mu_\theta}(s, a) = \mathop{\mathbb{E}}_{\tau \sim \mu_\theta}[G(\tau)|s_0 = s, a_0 = a] \tag{3.12}$$

Với giá trị trạng thái bắt đầu là $s$ và hành động $a$, chính sách $\mu_\theta$ nào cho giá trị $Q$ lớn nhất sẽ là chính sách tối ưu nhất, ký hiệu là $\mu_\theta^*$. Khi đó, hàm số $Q$ lớn nhất này được gọi là hàm số giá trị trạng thái - hành động tối ưu, ký hiệu ngắn gọn là $Q^*(s, a)$

$$Q^*(s, a) = \max_{\mu_\theta} Q(s, a) \tag{3.13}$$

Hơn thế nữa, hàm số $Q^*$ này luôn thỏa mãn một đẳng thức, gọi là đẳng thức Bellman, được trình bày như sau:

$$Q^*(s,a) \approx r + \gamma \max_{a'} Q^*(s',a') \tag{3.14}$$

Thật vậy, giá trị tổng thưởng kì vọng lớn nhất với trạng thái bắt đầu $s$ và hành động $a$ chính bằng phần thưởng đạt được $r$ cộng cho tổng thưởng kì vọng tối ưu với trạng thái bắt đầu $s'$ (đạt được từ việc thực hiện hành động $a$) và hành động $a'$ sao cho kết quả $Q^*(s',a')$ lớn nhất. Thêm vào đó, giá trị $\gamma \in (0,1)$ thể hiện sự tin tưởng vào các hàm số giá trị trạng thái - hành động giảm dần theo thời gian, các trạng thái càng xa trạng thái $s$ càng đóng góp ít hơn cho $Q^*(s,a)$.

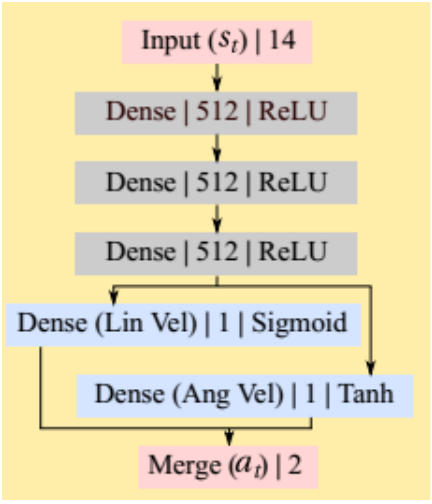
### 3.1.2 Giai đoạn huấn luyện

Ở giai đoạn này, mục tiêu chính là tìm được chính sách di chuyển tối ưu $\mu_\theta^*$, từ đó đưa vào triển khai ở giai đoạn thực tế.

#### Tham số hóa chính sách di chuyển của rô-bốt

Chính sách di chuyển của rô-bốt được định nghĩa bằng một hàm tất định, được xấp xỉ bằng một mạng nơ-ron với bộ trọng số $\theta$, $\mu_\theta(s_t)$, bao gồm 3 lớp ẩn với số chiều là 512 cùng với hàm kích hoạt ReLU tại mỗi lớp.

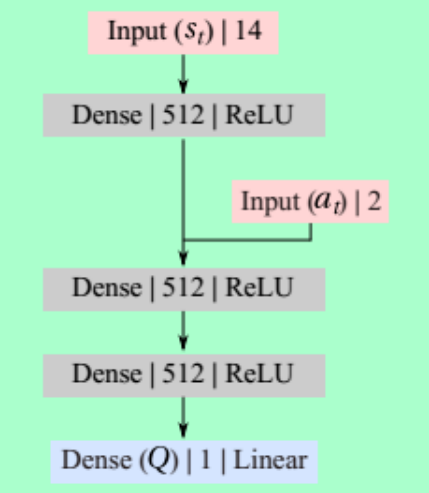
Với lớp cuối của mạng nơ-ron là véc-tơ 2 chiều, ứng với mỗi chiều, đầu ra của hàm số $a_t = \mu_\theta(s_t)$ bao gồm 2 giá trị vận tốc tuyến tính $v_t$ (được đưa về khoảng $[0,1]$ (m/s) với hàm kích hoạt sigmoid) và $\omega_t$ (được đưa về khoảng [-1,1] rad/s với hàm kích hoạt tanh).




Hình 3.3: Mạng nơ-ron xấp xỉ chính sách di chuyển

### Thuật toán Deep Deterministic Policy Gradient

Việc một hàm số $Q^*$ luôn thỏa mãn đẳng thức Bellman 3.14, đã đưa đến ý tưởng tham số hóa $Q^*$ và từ đó giải bài toán tối ưu cực tiểu với hàm mục tiêu là bình phương hiệu 2 về của đẳng thức. Thật vậy, [46] xấp xỉ $Q^*$ bởi một mạng nơ-ron (Hình 3.4) gồm 3 lớp ẩn (512 chiều và hàm kích hoạt ReLU) với bộ tham số $\phi$, gọi là $Q_\phi$. Trạng thái đầu vào $s_t$ là véc-tơ 14 chiều tương tự như đầu vào của chính sách di chuyển $\mu_\theta$, tham số hành động $a$ được đưa vào ở lớp ẩn thứ 2 của mạng nơ ron, lớp cuối mạng nơ-ron là 1 giá trị tuyến tính, dự đoán xấp xỉ hàm số giá trị trạng thái - hành động tối ưu.




Hình 3.4: Mạng nơ-ron xấp xỉ hàm số giá trị trạng thái - hành động $Q(s, a)$

Tiếp theo, thuật toán đề xuất việc thu thập các chuyển dịch trạng thái. Gọi $\mathcal{D}$ là tập hợp bao gồm các chuyển dịch trạng thái $\mathcal{D}_i = (s, a, r, s', d)$, với $d$ thể hiện việc $s'$ có là trạng thái kết thúc hoặc không tương ứng với hai giá trị (1, 0).

Đến đây, thay vì chỉ thiết lập và tìm giá trị $\phi^*$ tối ưu thỏa mãn đẳng thức Bellman như [26] đã làm, thuật toán Deep Deterministic Policy Gradient (DDPG) có sự bổ sung, với chính sách di chuyển $\mu_\theta$ đã được định nghĩa ở trên, thuật toán đi tìm $\theta^*$, cực đại hóa giá trị $Q_\phi$ cho các trạng thái thu thập được, bằng việc giải bài toán tối ưu sau:

$$\theta^* = \arg\max_\theta \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\big[Q_\phi(s, \mu_\theta(s))\big] \tag{3.15}$$

khi đó, ta có thể xấp xỉ

$$\max_a Q^*(s, a) \approx Q_\phi(s, \mu_\theta(s)) \tag{3.16}$$

Kết hợp 3.16 và đẳng thức Bellman 3.14, thiết lập hàm mục tiêu **Sai**

**số toàn phương trung bình Bellman**

$$\mathcal{L}(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s',d)\sim\mathcal{D}}\left[\Big(Q_\phi(s,a) - \big(r + \gamma(1-d)\,Q_\phi(s', \mu_\theta(s'))\big)\Big)^2\right] \quad (3.17)$$

,và cực tiểu nó:

$$\phi^* = \arg\min_\phi \mathcal{L}(\phi, \mathcal{D}) \quad (3.18)$$

Giá trị **mục tiêu** mà $Q_\phi(s,a)$ hướng đến là $r + \gamma(1-d)\,Q_\phi(s', \mu_\theta(s'))$, vấn đề là, giá trị mục tiêu này cũng phụ thuộc vào $\phi$, tham số cần học, điều này khiến cho việc cực tiểu $\mathcal{L}(\phi, \mathcal{D})$ không ổn định. Giải pháp đưa ra là sử dụng một bộ tham số khác cho hàm mục tiêu, $\phi_{targ}$. Tham số này sẽ được cập nhật một phần từ $\phi$ theo một tham số $\rho \in (0,1)$ (thường là rất gần 1). Tương tự, để tách biệt chính sách di chuyển hành vi và mục tiêu, tham số $\theta_{targ}$ được sử dụng, cách làm này được gọi là thuật toán ngoại chính sách.

$$\phi_{targ} \leftarrow \rho\phi_{targ} + (1-\rho)\phi$$
$$\theta_{targ} \leftarrow \rho\theta_{targ} + (1-\rho)\theta$$

Kết hợp lại, hàm mục tiêu cực tiểu được viết lại như sau:

$$\mathcal{L}(\phi, \mathcal{D}) = \mathop{\mathbb{E}}_{(s,a,r,s',d)\sim\mathcal{D}}\left[\Big(Q_\phi(s,a) - \big(r + \gamma(1-d)\,Q_{\phi_{targ}}(s', \mu_{\theta_{targ}}(s'))\big)\Big)^2\right] \quad (3.19)$$

$$\phi^* = \arg\min_\phi \mathcal{L}(\phi, \mathcal{D}) \quad (3.20)$$

Một điểm nữa là vấn đề duy trì khám phá, mà cụ thể ở đây, là việc cố gắng tránh khỏi khai thác các phần thưởng cục bộ để tiếp tục khám phá để tìm ra phần thưởng toàn cục. Ví dụ, một khi rô-bốt đã tìm được đường đi mà nó cho là tốt nhất, nó sẽ không ngừng di chuyển (khai thác) trên con đường này. Dù vậy, đường đi trên có thể chưa phải là tối ưu hoặc tại

một thời điểm nào đó, môi trường thay đổi và một con đường tối ưu hơn được mở ra. Làm thế nào để rô-bốt có thể duy trì việc khám phá để tìm ra chúng, tránh mắc kẹt tại tối ưu cục bộ là một bài toán rất khó giải quyết trong học tăng cường sâu hiện tại.

Đối với DDPG, thuật toán thực hiện thêm một giá trị nhiễu $\epsilon$ lấy mẫu từ phân phối chuẩn hóa, khi đó hành động được chọn ở mỗi bước sẽ là

$$a = clip(\mu_\theta(s) + \epsilon, a_{low}, a_{high}) \tag{3.21}$$

Hành động của rô-bốt sẽ tránh việc phụ thuộc hoàn toàn vào chính sách đang được học, từ đó di chuyển ngẫu nhiên hơn và tăng khả năng tìm được cực tiểu toàn bộ. Rô-bốt sẽ thực hiện khám phá nhiều ở những giai đoạn đầu của quá trình huấn luyện và sẽ ngưng việc khám phá này theo thời gian để tập trung khai thác phần thưởng tối ưu. Thật vậy, giá trị $\epsilon$ sẽ được giảm dần theo thời gian cho đến khi về 0.

Mã giả thuật toán DDPG:

**Algorithm 1** Thuật toán DDPG

**procedure** DDPG($\theta, \phi, \mathcal{D}$) ▷ $\theta, \phi$ được khởi tạo ngẫu nhiên, $\mathcal{D}$ rỗng
  $\phi_{targ} \leftarrow \phi$, $\theta_{targ} \leftarrow \theta$
  **repeat**
    Quan sát trạng thái $s$
    Lựa chọn hành động $a = clip(\mu_\theta(s) + \epsilon, a_{low}, a_{high})$, với $\epsilon \sim \mathcal{N}$
    Thực hiện a $\rightarrow s', r, d$
    Lưu $(s, a, s', r, d)$ vào $\mathcal{D}$
    **if** $\mathcal{D}$ đạt đến số lượng mẫu nhất định **then**
      **for** số lần cập nhật **do**
        Lấy mẫu ngẫu nhiên $B = (s, a, s', r, d)$ từ $\mathcal{D}$
        Tính toán giá trị mục tiêu
          $$y(s', r, d) = r + \gamma(1 - d)\, Q_{\phi_{targ}}(s', \mu_{\theta_{targ}}(s'))$$
        Cập nhật $\phi$ với Gradient Descent
          $$\nabla_\phi \frac{1}{|B|} \sum_{(s,a,s',r,d) \in B} (Q_\phi(s, a) - y(s', r, d))^2$$
        Cập nhật $\theta$ với Gradient Descent
          $$\nabla_\theta \frac{1}{|B|} \sum_{s \in B} Q_\phi(s, \mu_\theta(s))$$
        Cập nhật các tham số mục tiêu
          $$\phi_{targ} \leftarrow \rho\phi_{targ} + (1 - \rho)\phi$$
          $$\theta_{targ} \leftarrow \rho\theta_{targ} + (1 - \rho)\theta$$
  **until** hội tụ

### Thuật toán SAC

Cũng trên bài toán đã phát biểu, [3] thay thế thuật toán của giai đoạn huấn luyện SAC, một cải tiến trực tiếp từ DDPG. SAC sử dụng một chính sách ngẫu nhiên được tham số hóa $\pi_\theta(.|s_t)$. Bên cạnh đó, có những cải tiến trực tiếp từ những vấn đề mà DDPG giải quyết chưa tốt, bao gồm:

**• Kiểm soát tốt hơn vấn đề duy trì khám phá hay khai thác phần thưởng hiện tại**

Thật vậy, phần thưởng tại mỗi thời điểm được cộng thêm một giá trị tỉ lệ với giá trị entropy $H$ của $\pi_\theta$ theo hệ số $\alpha$.

$$R(s_t, a_t, s_{t+1}) + \alpha H(\pi_\theta(.|s_t)) \tag{3.22}$$

Theo đó, việc tối ưu phần thưởng này (hay tối đa giá trị entropy) khiến xác suất lựa chọn các hành động của chính sách được phân tán đều hơn. Đồng nghĩa với việc rô-bốt sẽ duy trì khám phá thay vì chỉ tập trung thực hiện hành động tối đa phần thưởng tại thời điểm hiện tại. Giá trị $\alpha$ có thể được điều chỉnh trong quá trình huấn luyện để tăng giảm mức độ khám phá theo mong muốn.

Đẳng thức Bellman cũng có những thay đổi tương ứng với phần thưởng mới, cụ thể như sau

$$Q(s,a) \approx r + \gamma(Q(s', \tilde{a}') + \alpha H \pi(.|s')) \approx r + \gamma(Q(s', \tilde{a}') - \alpha \log \pi(\tilde{a}'|s')) \tag{3.23}$$

trong đó, hành động tại mỗi thời điểm được lấy mẫu từ chính sách ngẫu nhiên $\pi_\theta$, là một phân phối Gauss với giá trị kì vọng $\mu$ và phương sai $\sigma$ được tham số hóa dưới dạng các hàm số tất định với tham số $\theta$:

$$\tilde{a} = \tanh(\mu_\theta(s) + \sigma_\theta(s) \odot \xi), \quad \xi \sim \mathcal{N}(0, I) \tag{3.24}$$

- **Ổn định hàm giá trị trạng thái - hành động**

Trong DDPG, hàm số $Q$ học được thường có xu hướng ước tính giá trị tổng thưởng kì vọng lớn hơn thực tế. Để khắc phục điều này, SAC thực hiện tham số 2 hàm giá trị trạng thái - hành động $Q_{\theta_1}, Q_{\theta_2}$, và sử dụng giá trị nhỏ hơn để tính toán mục tiêu cho sai số toàn phương trung bình

Bellman, theo đó

$$y(r, s', d) = r + \gamma(1 - d)\left(\min_{j=1,2} Q_{target,j}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s')\right) \quad (3.25)$$

Trong khi đó, việc tối ưu chính sách sẽ được thực hiện bằng việc giải bài toán tối ưu

$$\max_\theta = \mathop{\mathbb{E}}_{s\sim\mathcal{D}}\Big[\min_{j=1,2} Q_{\phi_j}(s, \tilde{a}_\theta(s, \xi)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s, \xi)|s)\Big] \quad (3.26)$$

Mã giả thuật toán SAC:

**Algorithm 2** Thuật toán SAC

**procedure** SAC($\theta, \phi_1, \phi_2, \mathcal{D}$) ▷ $\theta, \phi_1, \phi_2$ được khởi tạo ngẫu nhiên, $\mathcal{D}$ rỗng
  $\phi_{targ_1} \leftarrow \phi_1$, $\phi_{targ_2} \leftarrow \phi_2$
  **repeat**
    Quan sát trạng thái $s$ và lựa chọn $a \sim \pi_\theta(.|s)$
    Thực hiện a $\rightarrow s', r, d$
    Lưu $(s, a, s', r, d)$ vào $\mathcal{D}$
    **if** $\mathcal{D}$ đạt đến số lượng mẫu nhất định **then**
      **for** số lần cập nhật **do**
        Lấy mẫu ngẫu nhiên $B = (s, a, s', r, d)$ từ $\mathcal{D}$
        Tính toán giá trị mục tiêu
          $y(r, s', d) = r + \gamma(1 - d)*$
          $\left( \min_{j=1,2} Q_{target,j}(s', \tilde{a}') - \alpha \log \pi_\theta(\tilde{a}'|s') \right)$
        Cập nhật $\phi$ với Gradient Descent
          $\nabla_{\phi_j} \frac{1}{|B|} \sum_{(s,a,s',r,d) \in B} (Q_{\phi_j}(s, a) - y(s', r, d))^2$ ▷ for $j = 1, 2$
        Cập nhật $\theta$ với Gradient Descent
          $\nabla_\theta \frac{1}{|B|} \sum_{s \in B} \left( \min_{j=1,2} Q_{\phi_j}(s, \tilde{a}_\theta(s, \xi)) - \alpha \log \pi_\theta(\tilde{a}_\theta(s, \xi)|s) \right)$
        Cập nhật các tham số mục tiêu
          $\phi_{targ_1} \leftarrow \rho \phi_{targ_1} + (1 - \rho)\phi_1$
          $\phi_{targ_2} \leftarrow \rho \phi_{targ_2} + (1 - \rho)\phi_2$
  **until** hội tụ

### 3.1.3 Giai đoạn thực tế

Kết thúc quá trình huấn luyện, ta thu được giá trị $\theta^*$ và $\phi^*$ tối ưu. Tuy nhiên, giá trị $\phi$ chỉ được sử dụng cho quá trình học nhằm hỗ trợ tìm ra chính sách tối ưu. Theo đó, trong giai đoạn này, hành động của mỗi bước sẽ được suy thẳng từ chính sách tối ưu

$$a_t = \mu_{\theta^*}(s_t) \tag{3.27}$$

Khi đó, chu trình được chính sách tối ưu sinh ra sẽ có dạng

$$\tau = (s, a, r_1, s_1, \mu_{\theta^*}(s_1), ..., s_{T-1}, \mu_{\theta^*}(s_{T-1}), r_T, s_T) \tag{3.28}$$

Để đánh giá chính sách tối ưu trong giai đoạn thực tế, các độ đo khóa luận sử dụng bao gồm tổng thưởng của chu trình, số bước thực hiện, và quãng đường rô-bốt di chuyển. Trong đó, tổng thưởng $G$ được tính toán bằng việc lấy tổng các phần thưởng của chu trình trong công thức 3.28

$$G = \sum_{t=1}^{T} r_t \tag{3.29}$$

Giá trị $T$ cũng được đưa vào để đánh giá khi nó thể hiện được tốc độ thực hiện một chu trình hoàn chỉnh của rô-bốt. Bên cạnh đó, gọi tọa độ của rô-bốt tại mỗi thời điểm là $p_t = (p_x, p_y)$, quãng đường di chuyển $\mathcal{S}$ của rô-bốt sẽ là tổng khoảng cách Euclid của tọa độ của nó tại thời điểm $t$ và $t + 1$

$$\mathcal{S} = \sum_{t=1}^{T} \parallel p_t - p_{t-1} \parallel_2 \tag{3.30}$$

## 3.2 Đóng góp cải tiến

Mục đích chính đề xuất cải tiến này là nhằm tiết kiệm chi phí cho thiết bị ngoại vi, khi mà giá thành cho các thiết bị có khả năng phát laser thu về

giá trị khoảng cách thường là cao hơn nhiều so với camera đơn. Thật vậy, đã có những nghiên cứu cũng sử dụng mạng CNN để suy ra ảnh độ sâu từ ảnh RGB, và sử dụng nó như là trạng thái đầu vào của rô-bốt. Tuy nhiên những vấn đề gặp phải của cách làm này bao gồm: ảnh độ sâu đạt được chưa tận dụng được các mô hình dự đoán ảnh độ sâu đã được huấn luyện sẵn và trạng thái đầu vào lớn khiến rô-bốt chậm trong quá trình quyết định.

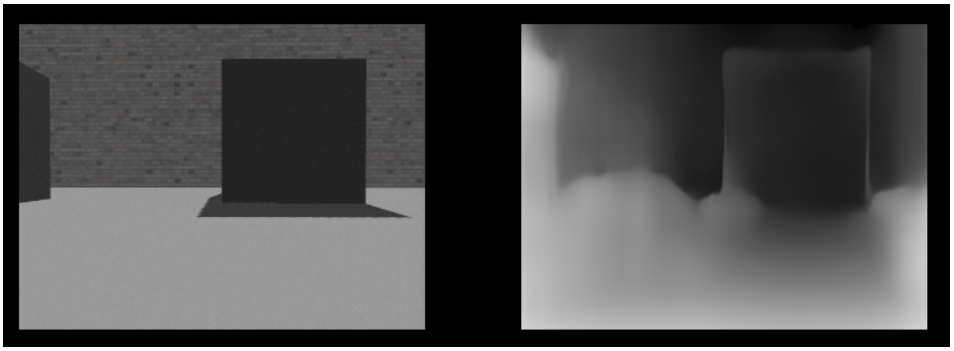
Thật vậy, để giải quyết các vấn đề này, khóa luận làm nhỏ véc-tơ trạng thái bằng cách chỉ lấy mẫu n giá trị độ sâu từ một dòng nhất định của ảnh. Công tác dự đoán độ sâu được kế thừa từ một kiến trúc mạng đã được huấn luyện sẵn với ưu điểm là có mô hình gọn nhẹ và độ chính xác cao. Từ đó, khóa luận thực hiện thêm những tinh chỉnh để kết quả thu được có thể ngang ngửa hoặc thấp hơn đôi chút so với phương pháp sử dụng giá trị khoảng cách từ cảm biến.

Khóa luận kế thừa mô hình được học sẵn để dự đoán thông tin độ sâu từ ảnh camera đơn 3 kênh màu. Mô hình này được trình bày ở công trình [10], được thiết kế là một mạng nơ-ron UNet, gồm 2 phần: mã hóa và giải mã. Công trình này huấn luyện mô hình trên tập dữ liệu KITTI [9], tập dữ liệu được thiết kế như là cầu nối giữa thị giác máy tính và các hệ tự hành.


Hình 3.5: Mô hình UNet dự đoán ảnh độ sâu

Mô hình UNet này cho phép đạt được các đặc trưng sâu cũng như là thông tin cục bộ của ảnh. Thật vậy, phần mã hóa được sử dụng mô hình ResNet-18 [14], với 11 triệu tham số, và được huấn luyện sẵn trên tập ImageNet. Trong khi đó, phần giải mã bao gồm các Upsampling Convolution

với hàm kích hoạt ELU tại mỗi lớp và hàm sigmoid tại lớp cuối cùng để thu về ma trận điểm giá trị khác biệt $D$ (disparity map).



Hình 3.6: Hình ảnh được camera của rô-bốt quan sát trong môi trường giả lập Gazebo (bên trái), và ma trận điểm giá trị khác biệt (disparity map) (bên phải).

Ưu điểm của mô hình này là tuy chỉ sử dụng phần mã hóa là ResNet-18 với đặc trưng sâu thu được từ ảnh là kém hiệu quả hơn so với các mô hình đồ sộ khác, nhưng lại cho lợi thế về tốc độ trong quá trình thực thi. Hơn thế nữa kết quả của mô hình cũng là cạnh tranh so với các phương pháp khác, khi nó được hỗ trợ mạnh mẽ dựa trên phương pháp học tự giảm sát, hay tự sinh ra nhãn từ dữ liệu sẵn có. Thật vậy, công trình này sử dụng yếu tố thời gian giữa các khung hình để thiết lập các hàm lỗi dựa trên sự tương quan của chúng. Phương pháp này suy diễn véc-tơ hướng liên quan $T_{t\to t'}$ giữa các 2 ảnh theo thời gian $I_t, I_{t'}$ bằng một mạng nơ-ron trích xuất đặc trưng (Hình 3.7). Gọi $D_t$ là ma trận điểm khác biệt được dự đoán từ ảnh $I_t$, và $I_{t\to t'}$ là ảnh $I_t$ được tái tạo bằng cách nội suy giá trị của phép chiếu $D_t$ lên $I_{t'}$ thông qua thông tin hướng $T_{t\to t'}$

$$I_{t\to t'} = I_{t'}\langle proj(D_t, T_{t\to t'})\rangle \tag{3.31}$$

, khi đó việc tối ưu hàm lỗi giữa độ tương đồng của ảnh đích $I_t$ và ảnh được tái tạo lại $I_{t\to t'}$ giúp tăng hiệu quả của mô hình dự đoán giải quyết phần nào vấn đề bị che khuất trong dự đoán ảnh độ sâu khi nó ghép cặp

được các điểm ảnh cùng thể hiện đối tượng thật theo thời gian dựa vào thông tin hướng.

$$L_p = \sum_t pe(I_t, I_{t\to t'}) \tag{3.32}$$

$$pe(I_a, I_b) = \frac{\alpha}{2}(1 - SSIM(I_a, I_b)) + (1-\alpha)||I_a - I_b||_1 \tag{3.33}$$

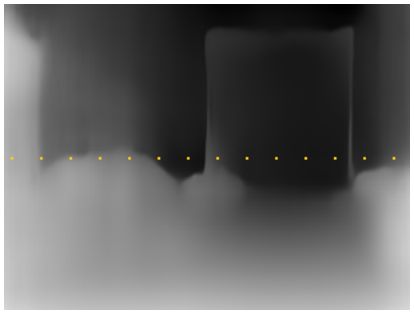
Trong đó, SSIM là độ đo tương đồng cấu trúc giữa hai ảnh, giá trị $\alpha = 0.85$


Hình 3.7: Mô hình dự đoán thông tin hướng bổ trợ

Với ma trận điểm khác biệt (disparity map) $D$ kích thước $(w, h)$ thu được, khóa luận thực hiện lấy mẫu $n$ giá trị $D_{i,j}$ nghịch đảo cách đều nhau, $i \in (0, w), j = h/2$.

$$o = \left[\frac{1}{D_{i_1,j}}, \frac{1}{D_{i_2,j}}, ..., \frac{1}{D_{i_n,j}}\right] \tag{3.34}$$

Thật vậy, giá trị $\frac{1}{D_{i,j}}$ tỉ lệ thuận với giá trị độ sâu của các điểm ảnh thuộc ảnh gốc. Điều này giống như việc xây dựng một hệ $n$ tia laser giả thu về khoảng cách chỉ dùng camera đơn với giá thành thấp hơn nhiều so với các thiết bị như LiDAR hay Radar.

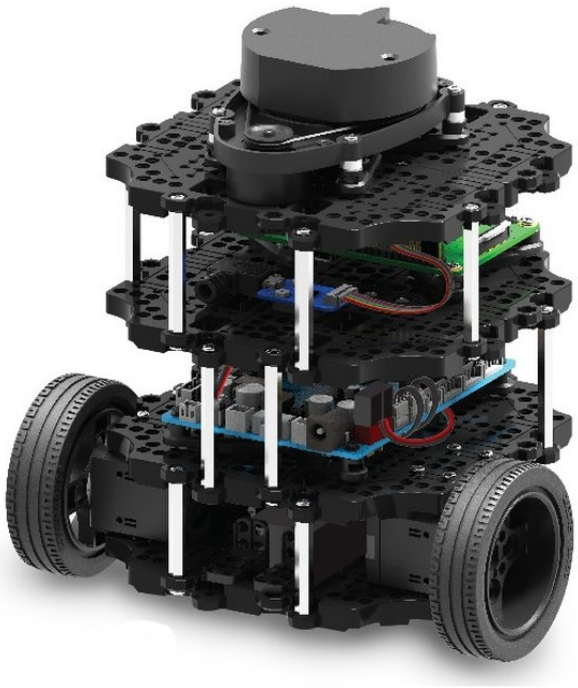


Hình 3.8: n giá trị độ sâu (màu vàng) lấy mẫu từ một dòng của ảnh độ sâu được dự đoán

# Chương 4

# Thực nghiệm

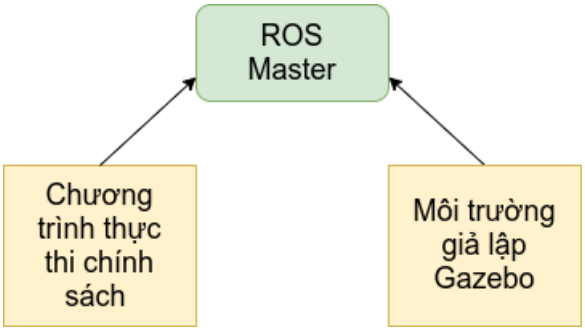
Ở chương này, khóa luận thực hiện quá trình huấn luyện và đánh giá trên môi trường ảo, giả lập bởi phần mềm Gazebo. Rô-bốt được chọn là TurtleBot Burger phiên bản 3.0. Khóa luận thiết kế các kịch bản huấn luyện cũng như là kịch bản thực tế từ đơn giản đến phức tạp qua đó đánh giá độ hiệu quả của việc sử dụng ảnh 3 kênh màu từ camera đơn giá rẻ so với sử dụng cảm biến với laser khoảng cách.



Hình 4.1: Hình dạng của TurtleBot Burger

Để hỗ trợ thực nghiệm, khóa luận cài đặt thêm một camera đơn và một cảm biến chạm cho rô-bốt. Cảm biến chạm giúp rô-bốt xác định được là đã chạm vật cản hay chưa với giá trị trả về là đúng hoặc sai. Trong khi đó, camera trả về một ảnh RGB có kích thước (228,304,3).

Khóa luận sử dụng nền tảng ROS là cầu nối giữa môi trường giả lập và mô hình học tăng cường sâu. Thật vậy, môi trường giải lập sẽ cho phép chương trình thực thi truy cập các thông tin về vị trí rô-bốt, các vật cản và điểm đích, cũng như là hình ảnh camera, giá trị khoảng cách thu được từ các tia laser hay việc rô-bốt đã va chạm phải vật cản hay chưa. Ở hướng ngược lại, chương trình sẽ gửi về môi trường ảo các giá trị hành động, cụ thể là vận tốc tuyến tính và vận tốc xoay cho rô-bốt hoạt động trong môi trường.



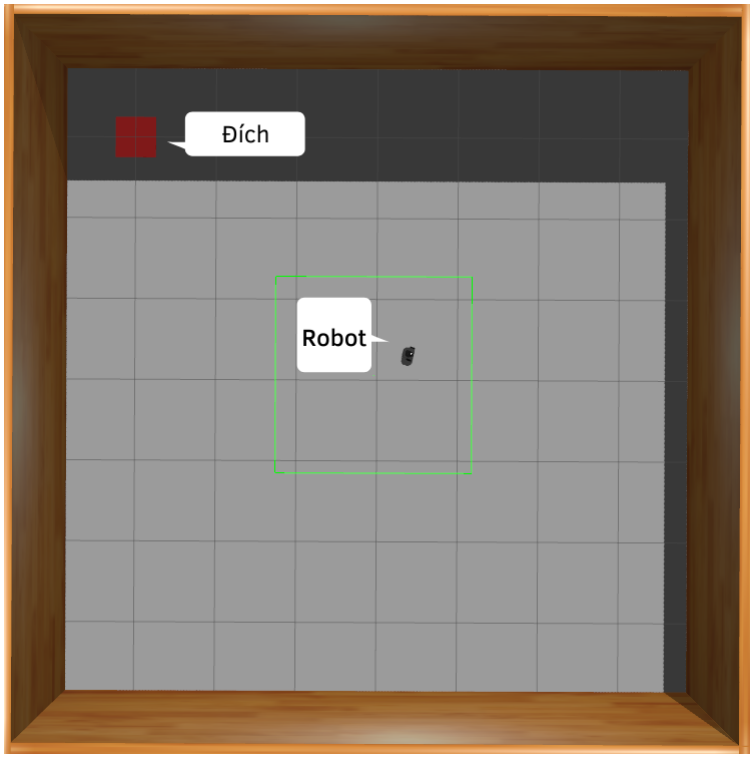

Hình 4.2: ROS kết nối chương trình thực thi chính sách học tăng cường sâu và môi trường giải lập Gazbo

## 4.1 Huấn luyện trên môi trường giả lập

### 4.1.1 Kịch bản huấn luyện thứ nhất

Ở kịch bản thứ nhất, rô-bốt được đặt trong không gian trống có diện tích khoảng $64m^2$, với xung quanh là bốn bức tường. Các phần thưởng sẽ xuất hiện ngẫu nhiên trong không gian này. Mỗi lần rô-bốt kết thúc một chu trình, bằng việc va vào tường, hay vượt quá thời hạn quy định cho một chu trình, nó sẽ được đặt lại vào vị trí ban đầu. Trong khi đó, nếu rô-bốt về đến đích, một phần thưởng mới sẽ được sinh ra ở một vị trí khác.

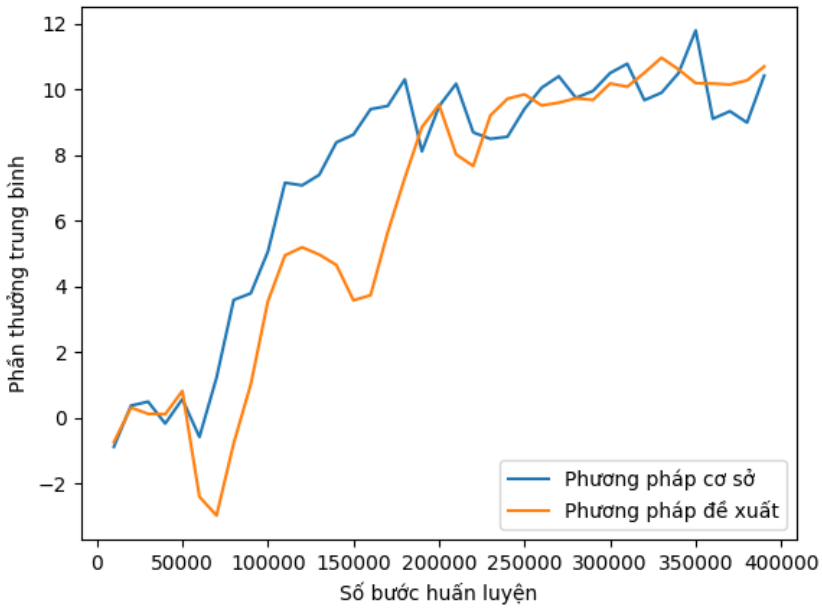
Kịch bản này không có vật cản nào được đặt ra, nên đây là một môi trường dễ, tạo điều kiện cho các mô hình dễ hội tụ, từ đó đánh giá bước đầu về sự thay đổi quan sát của rô-bốt được đề xuất.




Hình 4.3: Kịch bản huấn luyện số 1

Trong thực nghiệm này, các phần thưởng có giá trị lần lượt là: $r_{collision} = -200, r_{arrive} = 150, c_r = 500$. Giá trị $n$ số các giá trị khoảng cách từ cảm biến hay ảnh độ sâu được dự đoán là 10. Giới hạn của tập $\mathcal{D}$ bao gồm 100000 chuyển dịch trạng thái $\mathcal{D}_i$, số mẫu cho mỗi lần lấy từ tập này là 128. Để cập nhật các tham số tối ưu hàm mục tiêu, khóa luận thực hiện sử dụng thuật toán Adam [17] với tốc độ học có giá trị 0.0001. Bên cạnh đó, $\rho = 0.999$ cho việc cập nhật tham số mục tiêu. Thời gian huấn luyện tốn khoảng 45 phút mỗi 10000 chuyển dịch trạng thái (bước) được thực hiện.

Dưới đây là kết quả phần thưởng trung bình đạt được tại mỗi thời điểm $t$, trên mỗi 10000 bước huấn luyện. Trong thực nghiệm này số bước huấn luyện cho đến khi mô hình hội tụ 160000:



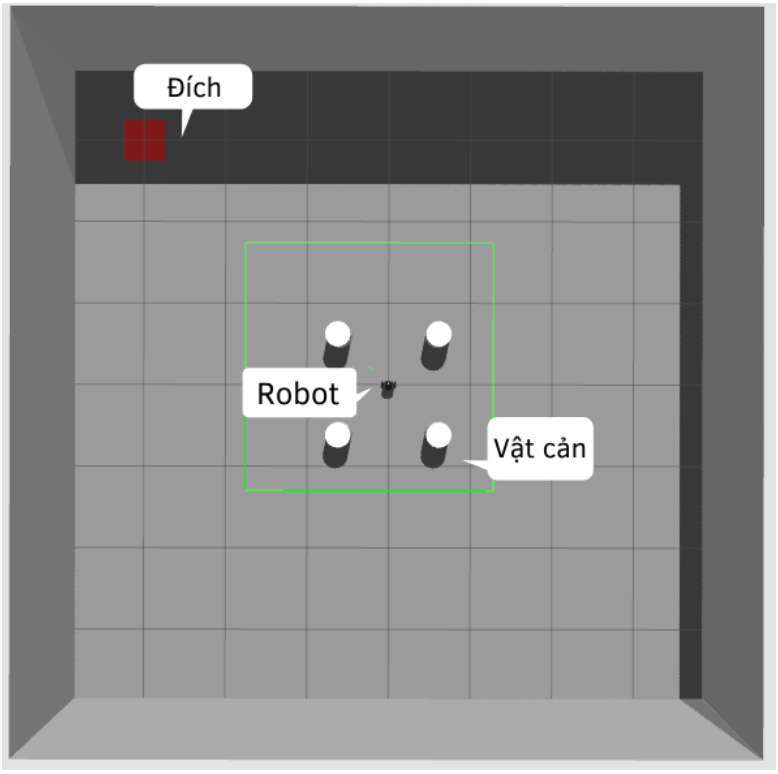

Hình 4.4: Phần thưởng trung bình đạt được tại một bước trên mỗi 10000 bước huấn luyện số 1

Nhận xét: có thể thấy rằng phương pháp đề xuất cho kết quả khả quan khi sử dụng mô hình học độ sâu được huấn luyện sẵn. Thật vậy, tuy hội tụ chậm hơn so với phương pháp cơ sở (200000 so với 150000 bước huấn luyện), thì phương pháp đề xuất vẫn cho kết quả xấp xỉ và có phần ổn định hơn đôi chút. Từ đó khóa luận tự tin áp dụng cho việc học trong môi trường phức tạp hơn, cụ thể là kịch bản huấn luyện thứ 2 dưới đây.

### 4.1.2 Kịch bản huấn luyện thứ 2

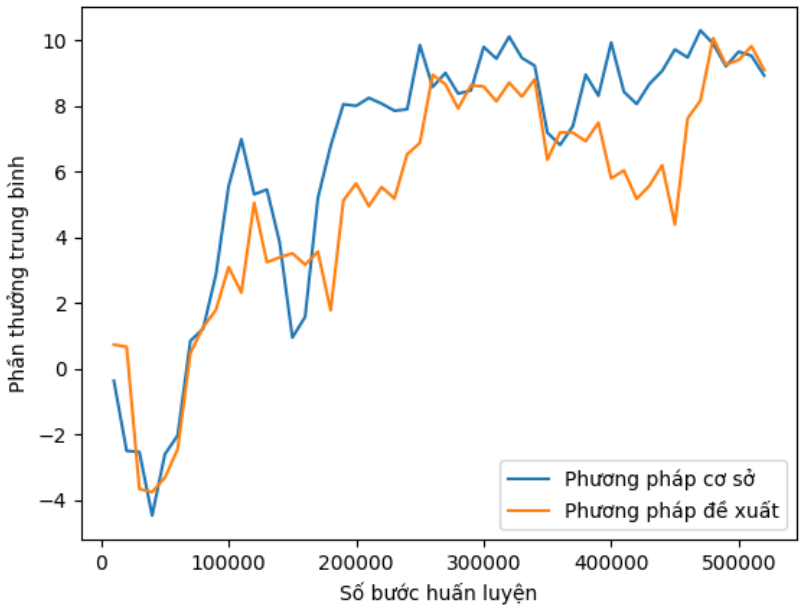
Ở kịch bản thứ hai, rô-bốt được cũng được đặt trong không gian trống có diện tích khoảng $64m^2$, với xung quanh là bốn bức tường. Tuy nhiên khác biệt ở đây, là sự xuất hiện của 4 vật cản hình trụ xung quanh vị trí xuất phát của rô-bốt, nhằm làm tăng tần suất của các trạng thái có vật cản cho việc thu thập các chu trình huấn luyện. Các phần thưởng cũng sẽ xuất hiện ngẫu nhiên trong không gian này, không trùng với các vị trí của vật cản. Mỗi lần rô-bốt kết thúc một chu trình, bằng việc va vào tường,

hay vượt quá thời hạn quy định cho một chu trình, nó sẽ được đặt lại vào vị trí ban đầu. Trong khi đó, nếu rô-bốt về đến đích, một phần thưởng mới sẽ được sinh ra ở một vị trí khác.




Hình 4.5: Kịch bản huấn luyện số 2

Trong thực nghiệm này, các giá trị phần thưởng, các tham số có thể tinh chỉnh trong thuật toán tương tự như khi huấn luyện trên kịch bản số 1 4.3. Dưới đây là kết quả phần thưởng trung bình đạt được tại mỗi thời điểm $t$, trên mỗi 10000 bước huấn luyện. Trong thực nghiệm này số bước huấn luyện khoảng hơn 500000 bước:



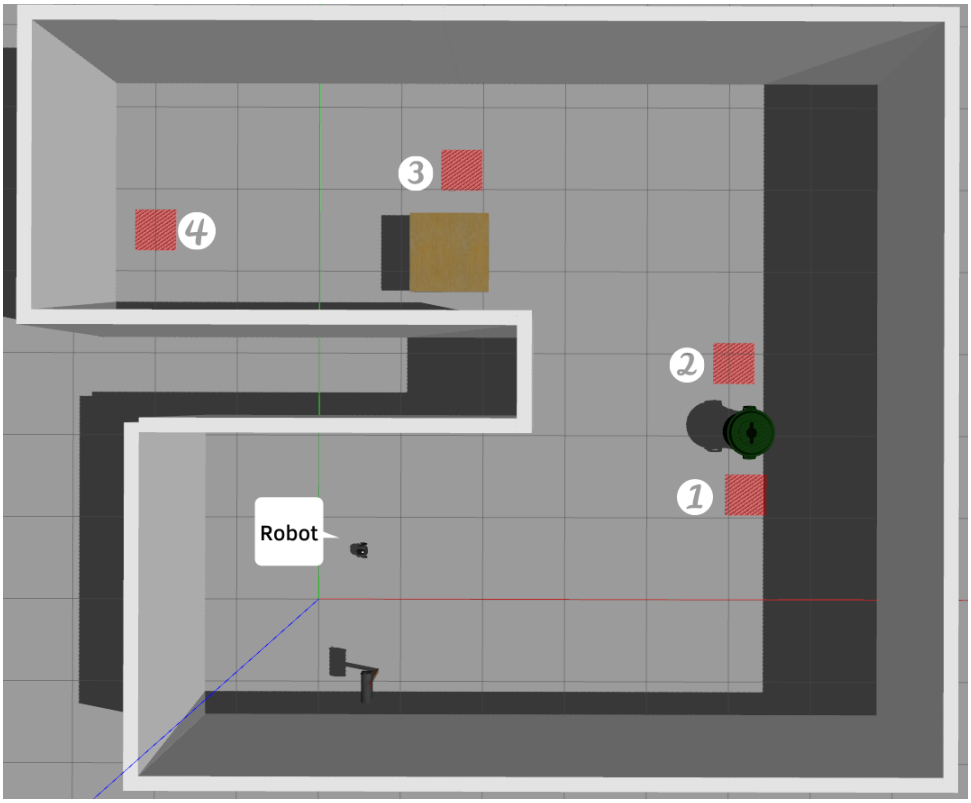

Hình 4.6: Phần thưởng trung bình đạt được tại một bước trên mỗi 10000 bước huấn luyện trên kịch bản huấn luyện số 2

Từ đồ thị trên có thể thấy rằng, thời gian hội tụ của 2 mô hình là như nhau (khoảng ở bước thứ 300000). Hơn thế nữa, kể từ đó, trong phần lớn các thời điểm, phương pháp đề xuất cũng cho phần thưởng trung bình xấp xỉ với phương pháp cơ sở.

## 4.2 Giai đoạn thực tế

Trong phần này, vì không có kinh phí để triển khai lên rô-bốt trên môi trường thực tế, khóa luận xây dựng các môi trường trên giả lập để mô phỏng quá trình này. Đây là những môi trường mà rô-bốt chưa từng nhìn thấy, từ đó đánh giá phương pháp đề xuất của khóa luận.

### 4.2.1 Môi trường thực tế 1




Hình 4.7: Môi trường thực tế số 1

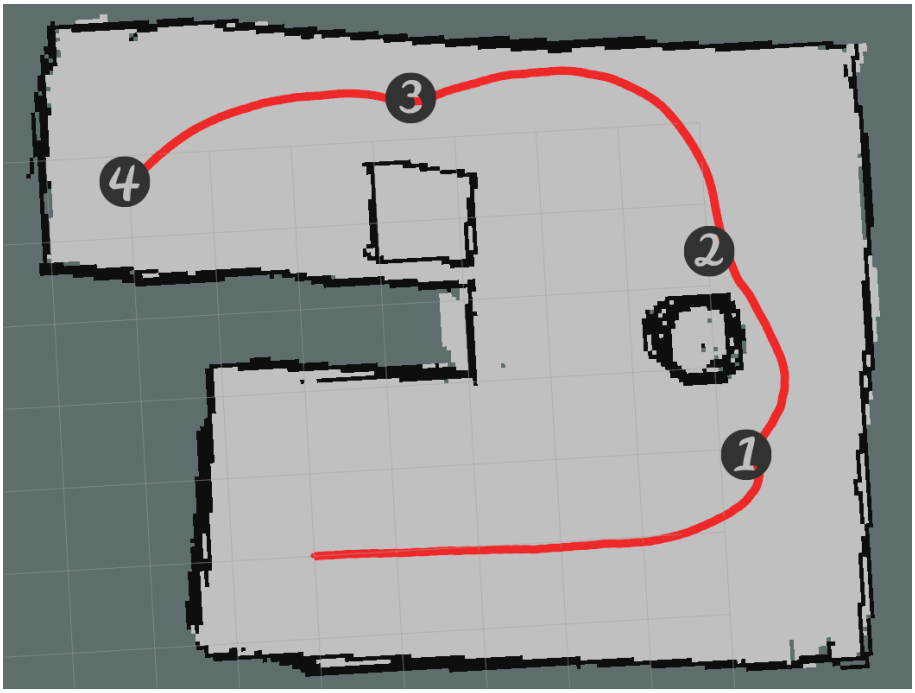
Ở môi trường thực tế số 1 này, mục tiêu của rô-bốt sẽ là di chuyển lần lượt đến các điểm đích màu đỏ như trên hình. Các tiêu chí đánh giá cho môi trường này bao gồm, số phần thưởng hoàn thành, nếu rô-bốt hoàn thành tất cả các mục tiêu, các tiêu chí tiếp theo sẽ bao gồm tổng thưởng sau cùng, số bước thực hiện và quãng đường di chuyển. Khóa luận thực hiện mỗi phương pháp 5 lần trên môi trường thực tế này và lấy kết quả đánh giá được chia trung bình. Số liệu được trình bày trong bảng 4.1 dưới đây.

Có thể thấy rằng, phương pháp đề xuất cho tổng phần thưởng khá xấp xỉ với phương pháp cơ sở, trong khi đó số bước thực hiện và quãng đường phải di chuyển là thấp hơn. Tuy nhiên, điều này cũng không chứng minh phương pháp đề xuất là tốt hơn, bởi vì đây chỉ là kết quả đánh giá trên

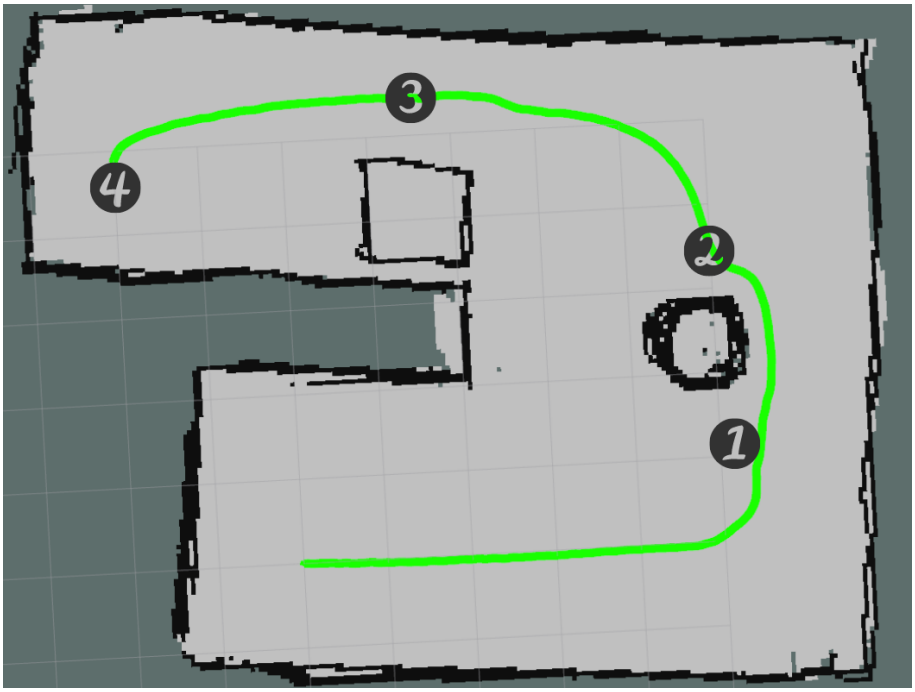
trung bình các lần thực hiện. Khóa luận muốn chỉ ra ở đây là phương pháp đề xuất có thể xấp xỉ tốt được phương pháp cơ sở mà chi phí bỏ ra cho thiết bị ngoại vi là thấp hơn. Quãng đường di chuyển của rô-bốt của từng phương pháp cũng tương đối giống nhau, được thể hiện quan Hình 4.8.

Bảng 4.1: Các kết quả đánh giá trên môi trường thực tế thứ 1

| **Phương pháp** | **Số điểm đích hoàn thành** | **Tổng thưởng** | **Số bước thực hiện** | **Quãng đường di chuyển** |
|---|---|---|---|---|
| Cơ sở + Kịch bản 1 | 2 | - | - | - |
| **Đề xuất + Kịch bản 1** | 1 | - | - | - |
| Cơ sở + Kịch bản 2 | 4 | **7621.67** | 631 | 17.50 |
| **Đề xuất + kịch bản 2** | **4** | 7538.65 | **513** | **17.08** |



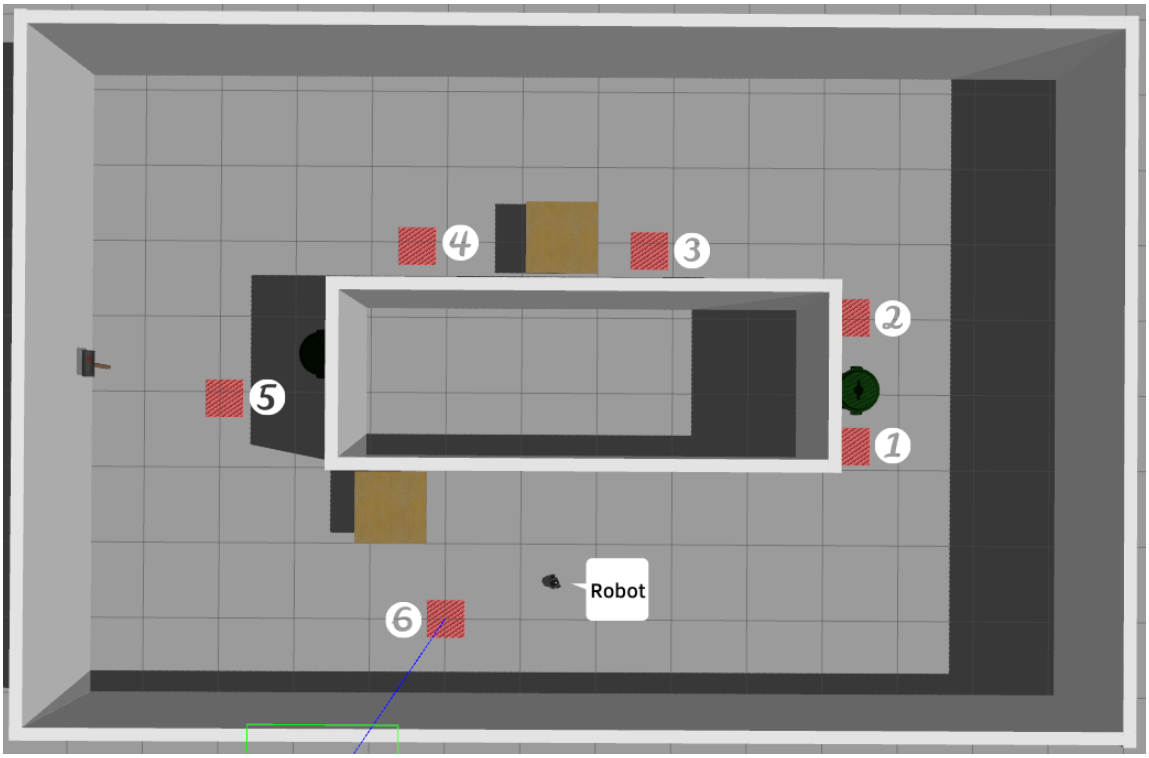

(a) Phương pháp cơ sở




(b) Phương pháp đề xuất

Hình 4.8: Đoạn đường di chuyển của rô-bốt qua các điểm đích của môi trường thực tế 1

### 4.2.2 Môi trường thực tế 2



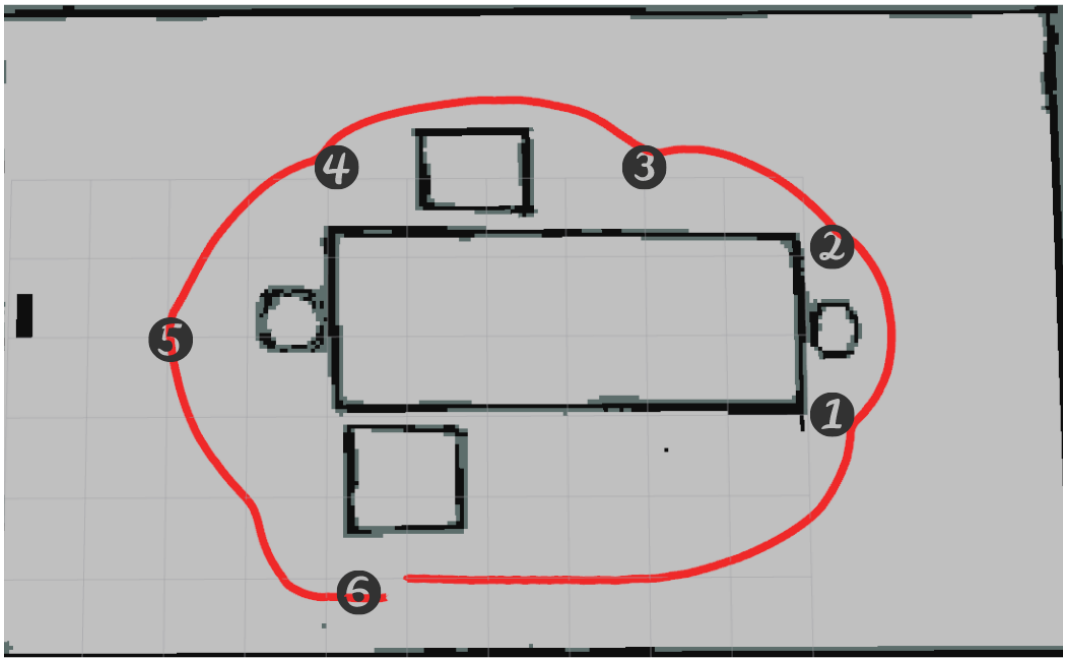

Hình 4.9: Môi trường thực tế số 2

Môi trường này được khóa luận xây dựng có phần phức tạp hơn so với cái trước. Cụ thể, môi trường được xây dựng giống như khuôn viên của một văn phòng với quãng đường phải di chuyển của rô-bốt là dài hơn. Các điểm đích cũng được đặt sát với các vật cản hơn và bị che khuất so với điểm đích kế tiếp. Khóa luận cũng thực hiện các thực nghiệm tương tự với những tiêu chí đánh giá giống như môi trường thực tế 1. Dưới đây là bảng kết quả 4.3 và quãng đường mà rô-bốt di chuyển (Hình 4.10).

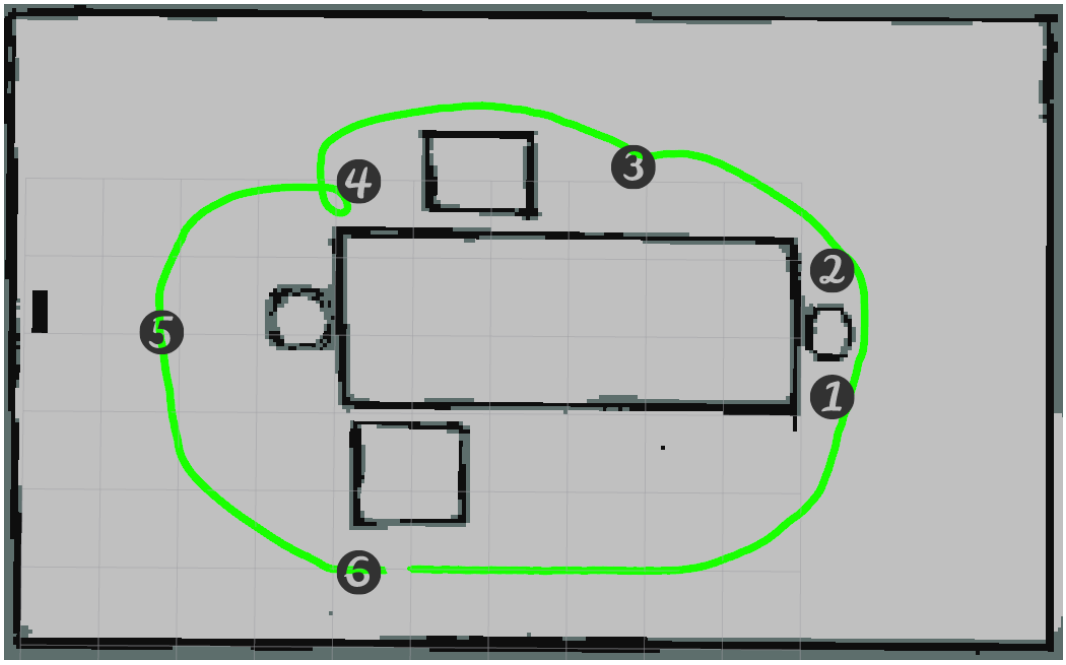
Kết quả thu được cũng là cạnh tranh đến từ phương pháp đề xuất. Mặc dù quãng đường di chuyển là dài hơn, số bước phải thực hiện lại ít hơn và tổng thưởng đạt được là nhiều hơn so với phương pháp cơ sở.

Bảng 4.2: Các kết quả trên môi trường thực tế thứ 2

| **Phương pháp** | **Số điểm đích hoàn thành** | **Tổng thưởng** | **Số bước thực hiện** | **Quãng đường di chuyển** |
|---|---|---|---|---|
| Cơ sở + Kịch bản 1 | 1 | - | - | - |
| **Đề xuất + Kịch bản 1** | 1 | - | - | - |
| Cơ sở + Kịch bản 2 | 6 | 10548.76 | 924 | **24.05** |
| **Đề xuất + kịch bản 2** | **6** | **10591.78** | **827** | 25.99 |




(a) Phương pháp cơ sở

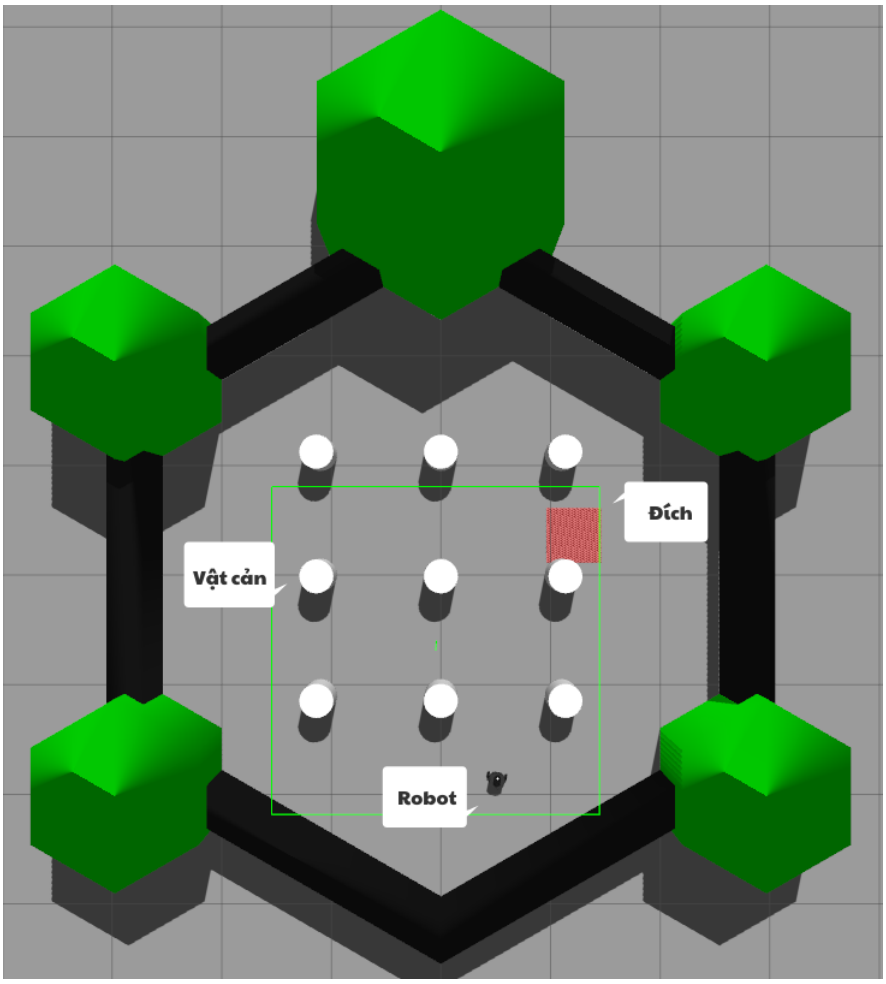



(b) Phương pháp đề xuất

Hình 4.10: Đoạn đường di chuyển của rô-bốt qua các điểm đích của môi trường thực tế 1

### 4.2.3 Môi trường thực tế 3

Đây là môi trường phức tạp nhất khi nó mô phỏng việc rô-bốt điều hướng tránh vật cản trong đám đông. Thật vậy, diện tích của khu vực là khá nhỏ với nhiều vật cản khá sát nhau (Hình 4.11). Điểm đích xuất hiện ngẫu nhiên trong môi trường, mỗi lần rô-bốt hoàn thành một chu trình (về đích hoặc chạm vật cản), một phần thưởng mới sẽ được sinh ra. Khóa luận thực hiện thiết lập 50 điểm đích và tiêu chí đánh giá được sử dụng là tỉ lệ thành công của chu trình mà mỗi phương pháp đạt được. Các phương pháp bao gồm phương pháp cơ sở và phương pháp đề xuất được huấn luyện trên môi trường huấn luyện 1 và 2.




Hình 4.11: Môi trường thực tế số 3

Bảng 4.3: Các kết quả trên môi trường thực tế thứ 3

| **Phương pháp** | **Tỉ lệ thành công (% )** |
|---|---|
| Cơ sở + Kịch bản 1 | 6 |
| **Đề xuất + Kịch bản 1** | 8 |
| Cơ sở + Kịch bản 2 | 40 |
| **Đề xuất + kịch bản 2** | 36 |

Với việc đây là một môi trường phức tạp với số lượng vật cản dày đặc, tỉ lệ thành công của các phương pháp là không quá 0.5. Tuy nhiên, xét trên mục đích của thực nghiệm, có thể thấy việc sử dụng camera đơn cũng cho kết quả xấp xỉ tốt các cảm biến ngoại vi giá thành cao.

# Chương 5

# Kết luận và hướng phát triển

## 5.1 Kết luận

Khóa luận đã thực hiện khảo sát các nghiên cứu giải quyết bài toán điều hướng rô-bốt, các phương pháp học tăng cường sâu và các nghiên cứu sử dụng học tăng cường sâu giải quyết bài toán điều hướng rô-bốt. Từ đó, xây dựng một hoạch định di chuyển không bản đồ cho rô-bốt di động có thể tự động điều hướng tránh vật cản và về đích an toàn, áp dụng phương pháp học tăng cường sâu.

Khóa luận kế thừa, trình bày cũng như cài đặt phương pháp cơ sở [46], với việc mô hình hóa bài toán dưới dạng một mô hình quyết định Markov (MDP) với trạng thái tại mỗi thời điểm là n giá trị khoảng cách từ rô-bốt đến các vật xung quanh theo n hướng khác nhau. Hành động cần trả về là giá trị vận tốc gốc và vận tốc tuyến tính. Thuật toán học tăng cường sâu được sử dụng là DDPG. Khóa luận cũng đề xuất sử dụng ảnh từ một camera đơn, thay thế cảm biến khoảng cách với mục đích chính là nhằm tiết kiệm chi phí cho thiết bị ngoại vi, đồng thời với những điều chỉnh phù hợp để giải quyết một số vấn đề trong các công trình sử dụng camera khác.

Kết quả thu được từ thực nghiệm cho thấy phương pháp đề xuất có tính cạnh tranh so với phương pháp sử dụng giá trị khoảng cách thu được từ các laser cảm biến.

## 5.2 Hướng phát triển

### 5.2.1 Về mặt lý thuyết

Trên cơ sở lý thuyết và thực nghiệm đã thực hiện, khóa luận cũng nhận thấy những hướng phát triển mới có thể tiếp tục nghiên cứu mở rộng hơn trong tương lai.

#### Cải tiến thuật toán ở giai đoạn huấn luyện

Hiện tại, học tăng cường sâu đang phát triển mạnh mẽ khi mà những thuật toán huấn luyện mới xuất hiện, với những ý tưởng cũng như chiến lược độc đáo để giải quyết các vấn đề then chốt trong học tăng cường. Có thể kể đến như làm thế nào để đạt được hiệu quả trong việc lấy mẫu, tức thời gian huấn luyện là ngắn nhưng kết quả đạt được vẫn như kỳ vọng. Hay đó là bài toán đánh đổi giữa việc duy trì khai thác phần thưởng hiện tại hay khám phá để có thể tìm được tối ưu toàn cục trong tương lai. Những thuật toán này thường được cài đặt trên môi trường trò chơi điện tử cũng như là các giả lập khác mà chưa được áp dụng nhiều vào điều hướng rô-bốt và tránh vật cản. Những hướng nghiên cứu mới mở ra đó là có thể điều chỉnh, cải tiến thêm các thuật toán này sao cho phù hợp, từ đó hy vọng đạt được những kết quả ấn tượng hơn.

#### Tăng tốc quá trình ra quyết định của rô-bốt

Việc làm gọn nhẹ những kiến trúc mạng đồ sộ để tạo ra những mô hình đơn giản với tốc độ nhanh trong quá trình thực thi và độ hiệu quả cạnh tranh cũng là một hướng nghiên cứu rất được quan tâm trong học sâu nói

chung. Thật vậy, ánh xạ từ trạng thái sang hành động một cách nhanh chóng là tiền đề quan trọng trong bài toán điều hướng rô-bốt, đặc biệt là với vật cản động. Thật vậy, để hoạt động hiệu quả, mô hình cần nhận xử lý được trạng thái đầu vào lớn với tốc độ nhanh. Đây cũng là một song đề khá phổ biến trong việc tiếp cận bài toán điều hướng tránh vật cản với học tăng cường sâu.

**Kết hợp song song yếu tố con người và học tăng cường sâu**

Học tăng cường sâu không phải lúc nào cũng hiệu quả cũng như là rất khó lường trước được. Thế nên, các cách tiếp cận có đồng thời sự trợ giúp của con người với một vài heuristic nhất định tại một vài thời điểm cần thiết có thể đem đến hiệu quả cao.

### 5.2.2 Về mặt ứng dụng

Khóa luận hướng đến các hướng nghiên cứu xây dựng những môi trường giả lập với quy mô lớn, gần gũi với thực tế hơn. Có thể kể đến điều hướng di chuyển rô-bốt trong các hội trường của những hội nghị lớn, nơi mà số lượng người (hay có thể xem là vật cản động) là con số hàng trăm. Từ đó, đặt ra các độ đo cũng như cài đặt những thuật toán cơ sở, góp phần tạo ra sân chơi cho những nghiên cứu lý thuyết mạnh mẽ khác có thể áp dụng.

# Tài liệu tham khảo

## Tiếng Anh


[1] J. Borenstein and Y. Koren. "Real-time obstacle avoidance for fast mobile robots". In: *IEEE Transactions on Systems, Man, and Cybernetics* 19.5 (1989), pp. 1179–1187.

[2] J. Borenstein and Y. Koren. "The vector field histogram-fast obstacle avoidance for mobile robots". In: *IEEE Transactions on Robotics and Automation* 7.3 (1991), pp. 278–288.

[3] Thomas Chaffre et al. "Sim-to-Real Transfer with Incremental Environment Complexity for Reinforcement Learning of Depth-Based Robot Navigation". In: *arXiv preprint arXiv:2004.14684* (2020).

[4] Chenyi Chen et al. "DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving". In: *The IEEE International Conference on Computer Vision (ICCV)*. 2015.

[5] Andrew Davison. "Real-time simultaneous localisation and mapping with a single camera". In: vol. 2. Jan. 2003, pp. 1403–1410. DOI: 10.1109/ICCV.2003.1238654.

[6] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: *CVPR09*. 2009.

[7] H. Durrant-Whyte and T. Bailey. "Simultaneous localization and mapping: part I". In: *IEEE Robotics Automation Magazine* 13.2 (2006), pp. 99–110.

[8] Scott Fujimoto, Herke van Hoof, and David Meger. *Addressing Function Approximation Error in Actor-Critic Methods*. 2018. arXiv: 1802.09477 [cs.AI].

[9] Andreas Geiger et al. "Vision meets robotics: The kitti dataset". In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[10] Clément Godard et al. "Digging into self-supervised monocular depth estimation". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2019, pp. 3828–3838.

[11] Tuomas Haarnoja et al. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. 2018. arXiv: 1801.01290 [cs.LG].

[12] H. Haddad et al. "Reactive navigation in outdoor environments using potential fields". In: *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*. Vol. 2. 1998, 1232–1237 vol.2.

[13] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[14] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[15] Matteo Hessel et al. *Rainbow: Combining Improvements in Deep Reinforcement Learning*. 2017. arXiv: 1710.02298 [cs.AI].

[16] H Jin Kim et al. "Autonomous helicopter flight via reinforcement learning". In: *Advances in neural information processing systems*. 2004, pp. 799–806.

[17] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[18] Nate Kohl and Peter Stone. "Policy gradient reinforcement learning for fast quadrupedal locomotion". In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*. Vol. 3. IEEE. 2004, pp. 2619–2624.

[19] Vijay R. Konda and John N. Tsitsiklis. "Actor-Critic Algorithms". In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 1008–1014. URL: `http://papers.nips.cc/paper/1786-actor-critic-algorithms.pdf`.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., 2012, pp. 1097–1105. URL: `http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf`.

[21] S. Lenser and M. Veloso. "Visual sonar: fast obstacle avoidance using monocular vision". In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453)*. Vol. 1. 2003, 886–891 vol.1.

[22] Timothy P. Lillicrap et al. *Continuous control with deep reinforcement learning*. 2015. arXiv: `1509.02971 [cs.LG]`.

[23] Liulong Ma, Jiao Chen, and Yanjie Liu. "Using RGB Image as Visual Input for Mapless Robot Navigation". In: *arXiv preprint arXiv:1903.09927* (2019).

[24] Jeff Michels, Ashutosh Saxena, and Andrew Y Ng. "High speed obstacle avoidance using monocular vision and reinforcement learning". In: *Proceedings of the 22nd international conference on Machine learning*. 2005, pp. 593–600.

[25] Piotr Mirowski et al. “Learning to navigate in complex environments”. In: *arXiv preprint arXiv:1611.03673* (2016).

[26] Volodymyr Mnih et al. *Playing Atari with Deep Reinforcement Learning*. 2013. arXiv: `1312.5602 [cs.LG]`.

[27] Jan Peters and Stefan Schaal. “Reinforcement learning of motor skills with policy gradients”. In: *Neural networks* 21.4 (2008), pp. 682–697.

[28] M. Pfeiffer et al. “From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots”. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 1527–1533.

[29] K. Qiu, F. Zhang, and M. Liu. “Let the Light Guide Us: VLC-Based Localization”. In: *IEEE Robotics Automation Magazine* 23.4 (2016), pp. 174–183.

[30] A. Remazeilles, F. Chaumette, and P. Gros. “Robot motion control from a visual memory”. In: *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA ’04. 2004*. Vol. 5. 2004, 4695–4700 Vol.5.

[31] E. Royer et al. “Outdoor autonomous navigation using monocular vision”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2005, pp. 1253–1258.

[32] Sebastian Ruder. *An overview of gradient descent optimization algorithms*. 2016. arXiv: `1609.04747 [cs.LG]`.

[33] P. Saeedi, P. D. Lawrence, and D. G. Lowe. “Vision-based 3-D trajectory tracking for unknown environments”. In: *IEEE Transactions on Robotics* 22.1 (2006), pp. 119–136.

[34] John Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: `1707.06347 [cs.LG]`.

[35] John Schulman et al. *Trust Region Policy Optimization*. 2015. arXiv: 1502.05477 [cs.LG].

[36] Robert Sim and J.J. Little. "Autonomous vision-based exploration and mapping using hybrid maps and Rao-Blackwellised particle filters". In: Nov. 2006, pp. 2082 –2089. DOI: 10.1109/IROS.2006.282485.

[37] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2014. arXiv: 1409.1556 [cs.CV].

[38] Y. Sun, M. Liu, and M. Q. . Meng. "WiFi signal strength-based robot indoor localization". In: *2014 IEEE International Conference on Information and Automation (ICIA)*. 2014, pp. 250–256.

[39] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*. Second Edition, Chapter 4, Dynamic Programming.

[40] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*. Second Edition, Chapter 5, Monte Carlo Methods.

[41] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*. Second Edition, Chapter 6, Temporal-Difference Learning.

[42] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*. Second Edition, Chapter 8, Planning and Learning with Tabular Methods.

[43] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning*. Second Edition, Part II, Approximate Solution Methods.

[44] Richard S Sutton et al. "Policy Gradient Methods for Reinforcement Learning with Function Approximation". In: *Advances in Neural Information Processing Systems 12*. Ed. by S. A. Solla, T. K. Leen, and K. Müller. MIT Press, 2000, pp. 1057–1063. URL: http://papers.nips.cc/paper/1713-policy-gradient-methods-for-reinforcement-learning-with-function-approximation.pdf.

[45] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[46] Lei Tai, Giuseppe Paolo, and Ming Liu. "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 31–36.

[47] Chuanqi Tan et al. *A Survey on Deep Transfer Learning*. 2018. arXiv: 1808.01974 [cs.LG].

[48] Zeng TP et al. "Learning Continuous Control through Proximal Policy Optimization for Mobile Robot Navigation". In: (2018).

[49] D. Wooden. "A guide to vision-based map building". In: *IEEE Robotics Automation Magazine* 13.2 (2006), pp. 94–98.

[50] Linhai Xie et al. "Towards monocular vision based obstacle avoidance through deep reinforcement learning". In: *arXiv preprint arXiv:1706.09829* (2017).

[51] Yuke Zhu et al. "Target-driven visual navigation in indoor scenes using deep reinforcement learning". In: *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE. 2017, pp. 3357–3364.