NGUYỄN ĐÌNH THÀNH

HỆ THỐNG THÔNG TIN

2025

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**


**NGUYỄN ĐÌNH THÀNH**

# NGHIÊN CỨU GIẢI THUẬT HỌC TĂNG CƯỜNG ỨNG DỤNG ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO XE TỰ HÀNH

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội – 2025

BỘ CÔNG THƯƠNG

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**


**NGUYỄN ĐÌNH THÀNH**

# NGHIÊN CỨU GIẢI THUẬT HỌC TĂNG CƯỜNG ỨNG DỤNG ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO XE TỰ HÀNH

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

Hà Nội – 2025

BỘ CÔNG THƯƠNG
**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**


**NGUYỄN ĐÌNH THÀNH**

# NGHIÊN CỨU GIẢI THUẬT HỌC TĂNG CƯỜNG ỨNG DỤNG ĐIỀU KHIỂN BÁM QUỸ ĐẠO CHO XE TỰ HÀNH

Ngành: Hệ thống thông tin

Mã số: 8480104

ĐỀ ÁN TỐT NGHIỆP THẠC SĨ HỆ THỐNG THÔNG TIN

NGƯỜI HƯỚNG DẪN:

1. TS. Nguyễn Văn Thiện

Hà Nội – 2025

# LỜI CAM ĐOAN

Tôi xin cam đoan đề án này là công trình nghiên cứu của riêng tôi và những nội dung được trình bày trong đề án này là hoàn toàn trung thực.

Những nội dung trình bày trong đề án này do tôi tìm hiểu, nghiên cứu và trình bày dưới sự hướng dẫn của TS. Nguyễn Văn Thiện.

Những số liệu, bảng biểu phục vụ cho việc phân tích và dẫn dắt được thu thập từ các nguồn tài liệu khác nhau được ghi chú trong mục tài liệu tham khảo hoặc chú thích ngay bên dưới các bảng biểu.

Ngoài ra, đối với các tài liệu diễn giải để làm rõ thêm các luận điểm đã phân tích và trích dẫn trong phần phụ lục cũng được chú thích nguồn gốc dữ liệu.

*Hà Nội, ngày  tháng  năm 2025*

Học viên thực hiện

# LỜI CẢM ƠN

Tôi xin chân thành cảm ơn TS. Nguyễn Văn Thiện đã tin tưởng và cho phép tôi chọn đề tài "Nghiên cứu giải thuật học tăng cường ứng dụng điều khiển bám quỹ đạo cho xe tự hành". Đề tài này đã mang lại cho tôi nhiều trải nghiệm quý báu cũng như kiến thức vô cùng bổ ích trong lĩnh vực trí tuệ nhân tạo.

Trong quá trình thực hiện đề án, tôi đã được hỗ trợ nhiệt tình từ các thầy. Những kiến thức, kinh nghiệm cùng những lời khuyên của các thầy đã giúp tôi hoàn thành đề tài một cách hiệu quả nhất.

Tôi cũng xin bày tỏ lòng biết ơn sâu sắc đến tập thể giáo viên và những học viên đã giúp đỡ, động viên và cổ vũ tôi trong suốt quá trình nghiên cứu, thực hiện đề án.

Đề án này không chỉ giúp tôi nâng cao hiểu biết và kỹ năng nghiên cứu mà còn giúp tôi có cơ hội thực hành và áp dụng các kiến thức đã học vào thực tế. Tôi tin rằng những kết quả và kinh nghiệm thu được từ đề án sẽ có giá trị thực tiễn cao và có thể áp dụng được trong công việc của tôi trong tương lai.

Một lần nữa, tôi xin chân thành cảm ơn các thầy đã giúp đỡ tôi trong quá trình nghiên cứu và thực hiện đề án này.

Trân trọng!

Học viên thực hiện

# MỤC LỤC

# DANH MỤC CÁC KÝ HIỆU, CÁC TỪ VIẾT TẮT

| Viết tắt | Tiếng Anh | Tiếng Việt |
|---|---|---|
| ADP | Adaptive Dynamic Programming | Phương pháp lập trình động thích nghi |
| AI | Artificial Intelligence | Trí tuệ nhân tạo |
| DQN | Deep Q-Network | Giải thuật Deep Q-Network |
| DT | Discrete-Time | Hệ thống động lực phi tuyến rời rạc |
| FLS | Fuzzy Logic System | Hệ thống logic mờ |
| FNN | Fuzzy Neural Network | Mạng nơ-ron mờ |
| FVI | Fitted Value Iteration | Giải thuật lặp giá trị xấp xỉ Fitted Value Iteration, là biến thể của thuật toán Value Iteration |
| LQR | Linear Quadratic Regulator | Bộ điều khiển toàn phương tuyến tính |
| MDP | Markov Decision Process | Mô hình quyết định Markov |
| MIMO | Multiple-Input Multiple-Output | Nhiều đầu vào - nhiều đầu ra |
| NN | Neural Network | Mạng nơ-ron nhân tạo |
| OADP | Online Adaptive Dynamic Programming | Phương pháp lập trình động thích ứng |
| PDZ | Pseudo Dead Zone | Vùng chết giả là vùng chết được mô phỏng hoặc giả lập trong hệ điều khiển, để mô phỏng các đặc tính phi tuyến |
| PE | Persistent Excitation | Điều kiện kích thích bền vững trong các hệ thống thích nghi |

| | | |
|---|---|---|
| PI | Policy Iteration | Giải thuật Policy Iteration |
| RL | Reinforcement Learning | Học tăng cường |
| RLNN | Reinforcement Learning Neural Network | Mạng Nơ-ron trong Học Tăng cường |
| UUB | Uniformly Ultimately Bounded | Tính bị chặn cuối cùng đồng nhất |
| VI | Value Iteration | Giải thuật chính sách tối ưu Value Iteration |
| WMR | Wheeled Mobile Robot | Robot di động có bánh xe |

# DANH MỤC HÌNH ẢNH

# LỜI MỞ ĐẦU

## 1. Lý do chọn đề tài

Đề tài "Nghiên cứu giải thuật học tăng cường ứng dụng điều khiển bám quỹ đạo cho xe tự hành" được chọn với những lý do và tính cấp thiết quan trọng trong bối cảnh phát triển mạnh mẽ của phong trào xe tự hành. Xe tự hành, đang trở thành xu hướng quan trọng trong lĩnh vực ô tô và công nghệ, đặt ra những thách thức lớn đối với các nhà nghiên cứu và kỹ sư. Việc nghiên cứu giải thuật học tăng cường là cực kỳ cần thiết để cải thiện khả năng tự hành của xe trong môi trường đường phố động và biến đổi liên tục.

Tính cấp thiết của đề tài thể hiện qua nhu cầu ngày càng tăng về an toàn giao thông, hiệu suất và linh hoạt của xe tự hành. Điều này không chỉ đảm bảo sự an toàn cho hành khách và người dùng, mà còn tạo ra những ứng dụng thực tế và có ý nghĩa lớn trong cuộc sống hàng ngày. Khả năng giải quyết các thách thức thời gian thực cũng là một khía cạnh quan trọng, giúp giảm độ trễ trong quá trình đưa ra quyết định, nâng cao khả năng xử lý thời gian thực của xe tự hành.

Nghiên cứu giải thuật học tăng cường cũng có tính chất đặc biệt trong việc chịu ảnh hưởng của các yếu tố đa dạng như thời tiết biến đổi, điều kiện đường khác nhau, và mức độ giao thông đa dạng. Đề tài không chỉ hướng tới việc giải quyết các thách thức hiện tại mà còn đóng góp cho sự tiến bộ toàn cầu trong lĩnh vực học máy và xe tự hành. Trong bối cảnh này, việc nghiên cứu giải thuật học tăng cường cho xe tự hành không chỉ là một đề tài hứa hẹn mà còn là đóng góp to lớn cho sự phát triển của công nghiệp và xã hội.

## 2. Mục tiêu nghiên cứu

Đề tài đặt ra một loạt mục tiêu quan trọng để đáp ứng những thách thức ngày càng phức tạp của công nghệ xe tự hành. Trước hết, nghiên cứu sẽ tập trung phân tích chi tiết về các thách thức mà xe tự hành phải đối mặt, bao gồm cả tình huống giao thông phức tạp và biến đổi thời tiết. Mục tiêu chính của đề tài là xây dựng và phát triển một giải thuật học tăng cường đặc biệt, được tối ưu hóa cho xe tự hành, có khả năng học từ dữ liệu thực tế và điều chỉnh để tối ưu hóa quyết định trong môi trường biến động.

Để đảm bảo tính ứng dụng thực tế, nghiên cứu đặt ra mục tiêu tích hợp giải thuật vào hệ thống xe tự hành thực tế và tiến hành các bài kiểm thử chi tiết để đánh giá hiệu suất và khả năng thích ứng của nó trong nhiều tình huống giao thông và điều kiện đường khác nhau. Qua đó, mục tiêu tối ưu hóa hiệu suất và linh hoạt của giải thuật, giúp xe tự hành thích ứng linh hoạt với môi trường biến động.

Đặc biệt, đề tài hướng tới việc ứng dụng rộng rãi giải thuật trong ngữ cảnh công nghiệp và cuộc sống hàng ngày, với mong muốn đóng góp vào sự phát triển của công nghệ xe tự hành. Cuối cùng, mục tiêu cuối cùng là đánh giá ý nghĩa xã hội của nghiên cứu, đặc biệt là trong việc cải thiện an toàn giao thông và mang lại tiện ích cho người dùng. Những mục tiêu này đồng lòng hướng tới một nghiên cứu toàn diện và ứng dụng có ảnh hưởng tích cực trong lĩnh vực ngày càng quan trọng của xe tự hành.

## 3. Đối tượng và phạm vi nghiên cứu

### ❖ Đối tượng nghiên cứu

Đối tượng nghiên cứu chủ yếu là các hệ thống xe tự hành và các thành phần liên quan trong lĩnh vực công nghiệp ô tô. Nghiên cứu sẽ tập trung vào việc phát triển giải thuật học tăng cường để điều khiển bám quỹ đạo cho xe tự

hành khi tính đến yếu tố bất định, xe di chuyển với nhiều loại quỹ đạo khác nhau.

❖ **Phạm vi nghiên cứu**

- Nghiên cứu và so sánh các giải thuật học tăng cường hiện đại, nghiên cứu xây dựng mô hình toán học, chú trọng vào khả năng ứng dụng chúng trong điều khiển bám quỹ đạo của xe tự hành.

- Nghiên cứu đề xuất bộ điều khiển bám trong trường hợp nhiễu tác động có biên độ nhỏ.

- Nghiên cứu đề xuất bộ điều khiển thích nghi cho xe tự hành bám quỹ đạo có xem xét bù trượt bánh xe để nâng cao chất lượng điều khiển xe.

**4. Phương pháp nghiên cứu**

Để đạt được mục tiêu nghiên cứu của đề tài quá trình nghiên cứu được thiết kế theo một chuỗi bước hợp lý và toàn diện. Đầu tiên, tôi sẽ tiến hành một phân tích chi tiết về các thách thức mà xe tự hành phải đối mặt, bao gồm cả tình huống giao thông phức tạp và biến đổi thời tiết.

Sau đó, quá trình thu thập dữ liệu sẽ bắt đầu từ nhiều nguồn khác nhau, nhằm đảm bảo độ đa dạng và đầy đủ cho quá trình huấn luyện giải thuật. Dữ liệu này bao gồm cả môi trường thực tế và môi trường mô phỏng, cung cấp nền tảng cho việc xây dựng mô hình học tăng cường chuyên biệt cho xe tự hành.

Mô hình này sẽ được phát triển để có khả năng học từ dữ liệu thực tế và linh hoạt điều chỉnh quyết định dựa trên những tình huống mới và biến động. Quá trình tích hợp giải thuật vào hệ thống xe tự hành thực tế sẽ được thực hiện để đảm bảo sự tương tác mượt mà và hiệu quả.

Tiếp theo, tôi sẽ thực hiện các bài kiểm thử chi tiết trong môi trường mô phỏng, đánh giá khả năng phản ứng và quyết định của giải thuật trong các tình

huống đặc biệt. Sau đó, quá trình kiểm thử thực tế sẽ được thực hiện trên đường phố để xác nhận hiệu suất của giải thuật trong điều kiện thực tế.

Cuối cùng, kết quả của giải thuật sẽ được so sánh với các phương pháp khác trong lĩnh vực tương tự, và đánh giá độ chính xác và hiệu suất của giải thuật trong nhiều điều kiện đường và tình huống giao thông. Đánh giá sẽ không chỉ tập trung vào khía cạnh lý thuyết mà còn vào ứng dụng thực tế và ý nghĩa xã hội, đặc biệt là trong việc cải thiện an toàn giao thông và tiện ích cho người dùng. Như vậy, phương pháp nghiên cứu này hứa hẹn mang lại những đóng góp tích cực cho lĩnh vực ngày càng quan trọng của xe tự hành.

## 5. Cấu trúc của đề án

Đề án được cấu trúc gồm 3 chương:

Chương 1: Cơ sở lý thuyết

Chương 2: Mô hình bài toán và giải pháp đề xuất

Chương 3: Thực nghiệm và kết quả

# CHƯƠNG 1 - CƠ SỞ LÝ THUYẾT

## 1.1. Tổng quan về điều khiển bám quỹ đạo cho xe tự hành

Trí tuệ nhân tạo AI - Artificial Intelligence là một thuật ngữ hết sức quen thuộc và phổ biến trong thời đại số 4.0 ngày này. AI được áp dụng trong hầu hết các lĩnh vực có thể kể đến như nghiên cứu khoa học, kinh tế, xã hội… có thể nói rằng AI đang xâm chiếm xã hội loài người, rất nhiều thắc mắc nêu ra rằng liệu một ngày nào đó AI sẽ thay thế con người.

AI cho phép máy móc học hỏi từ kinh nghiệm, thích nghi với dữ liệu mới và thực hiện các nhiệm vụ giống như con người, chẳng hạn như hiểu ngôn ngữ, nhận diện giọng nói, phân tích hình ảnh, đưa ra quyết định và giải quyết vấn đề. Khác với các hệ thống truyền thống chỉ làm theo những chỉ dẫn cố định, AI có khả năng tự động cải thiện hiệu suất thông qua việc xử lý và học từ dữ liệu.

Trong lĩnh vực robotics và tích hợp cơ học, trí tuệ nhân tạo đang được sử dụng để thiết kế và điều khiển các robot và hệ thống tự động hóa. Khi được tích hợp với AI, các robot không chỉ đơn thuần là những cỗ máy hoạt động theo lập trình, mà có khả năng học hỏi, cảm nhận môi trường xung quanh, thích nghi với các tình huống mới và thực hiện các hành động phức tạp. Các ứng dụng của AI trong robotics rất đa dạng: từ xe tự hành, máy bay không người lái, robot vận chuyển hàng hóa trong kho, đến robot phục vụ trong nhà hàng, robot hỗ trợ phẫu thuật hay chăm sóc người già và bệnh nhân trong bệnh viện. Đây là minh chứng rõ nét cho khả năng kết hợp giữa trí tuệ nhân tạo và cơ học để giải quyết các vấn đề trong thế giới thực. Một trong những mục tiêu lớn của trí tuệ nhân tạo là phát triển trí tuệ nhân tạo mạnh (Strong Artificial Intelligence), nơi máy tính có khả năng suy luận, tự học và tự tiến hóa như con người. Đây là một lĩnh vực nghiên cứu và phát triển cực kỳ phức tạp và đầy thách thức, nhưng nếu thành công, nó có thể mang lại nhiều lợi ích to lớn cho nhân loại.

Trong những năm gần đây, cùng với sự phát triển mạnh mẽ của khoa học kỹ thuật và công nghệ trí tuệ nhân tạo, các hệ thống xe tự hành (Autonomous Vehicles – AVs) ngày càng thu hút sự quan tâm rộng rãi từ cộng đồng nghiên cứu cũng như các doanh nghiệp. Xe tự hành được kỳ vọng mang lại những lợi ích to lớn về an toàn giao thông, tối ưu hóa năng lượng, giảm thiểu chi phí vận hành, đồng thời mở ra các ứng dụng thực tiễn đa dạng như vận tải thông minh, robot dịch vụ, xe công nghiệp trong kho bãi hay phương tiện vận chuyển trong môi trường nguy hiểm.

Một trong những bài toán trọng tâm và mang tính nền tảng trong lĩnh vực xe tự hành chính là bài toán điều khiển bám quỹ đạo (trajectory tracking control). Bài toán này được hiểu là quá trình thiết kế và vận hành một bộ điều khiển sao cho xe di chuyển theo đúng quỹ đạo tham chiếu đã định sẵn, trong khi vẫn duy trì các đặc tính mong muốn về ổn định, chính xác và hiệu quả. Vấn đề đặt ra không chỉ là xe có thể đi theo đường dẫn cho trước, mà quan trọng hơn là duy trì sai số bám quỹ đạo ở mức tối thiểu trong suốt quá trình vận hành, ngay cả khi hệ thống phải đối mặt với các yếu tố bất định như sai số cảm biến, nhiễu môi trường, hay những biến động trong động lực học của xe.

Bám quỹ đạo có ý nghĩa đặc biệt quan trọng bởi vì nó gắn liền trực tiếp với an toàn vận hành và mức độ tin cậy của hệ thống xe tự hành. Trong thực tế, một phương tiện tự hành nếu không thể duy trì quỹ đạo ổn định và chính xác sẽ dễ dẫn đến va chạm, sai lệch khỏi lộ trình hoặc gây tiêu hao năng lượng quá mức. Do đó, các nghiên cứu về bài toán này không chỉ tập trung vào việc nâng cao độ chính xác của thuật toán điều khiển, mà còn quan tâm đến sự ổn định toàn cục của hệ thống, độ mượt của tín hiệu điều khiển, cũng như khả năng thích nghi với nhiều điều kiện vận hành khác nhau.

Trong nhiều nghiên cứu, bài toán bám quỹ đạo thường được chia thành hai thành phần chính:

- Bám vị trí (position tracking): đảm bảo rằng xe duy trì được khoảng cách nhỏ nhất đến quỹ đạo tham chiếu.
- Bám hướng (orientation tracking): bảo đảm rằng xe không chỉ đi đúng vị trí mà còn duy trì được hướng chuyển động phù hợp với quỹ đạo.

Sự kết hợp đồng thời của hai thành phần này cho phép hệ thống đạt được khả năng điều khiển chính xác, ổn định và bền vững hơn.

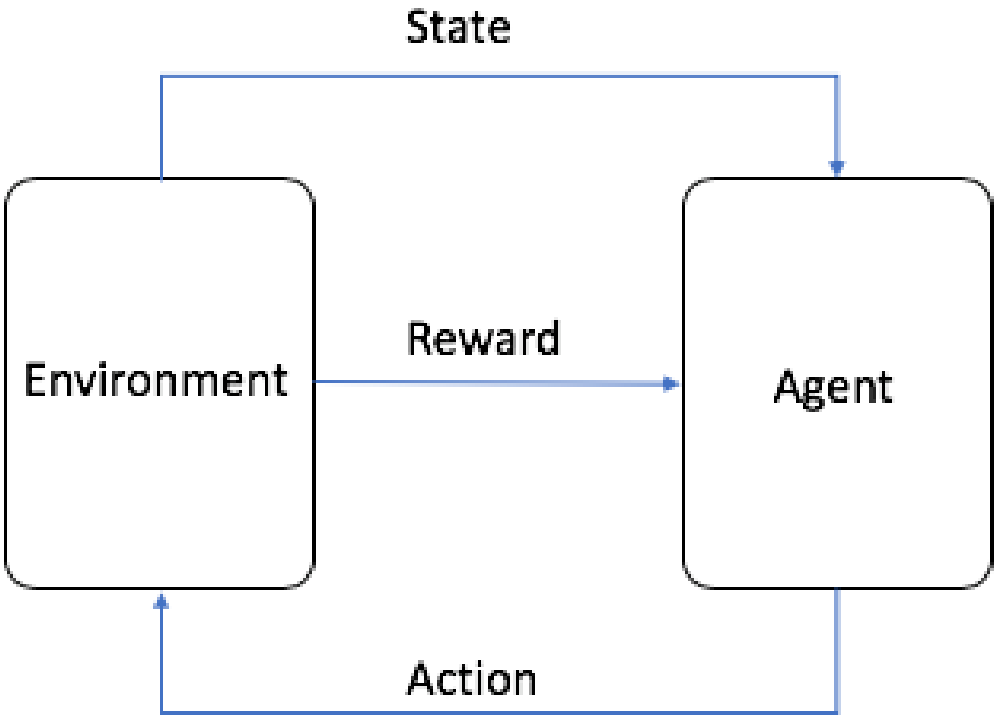
Trong bối cảnh thực tiễn, nhiều phương pháp điều khiển đã được phát triển để giải quyết bài toán bám quỹ đạo cho xe tự hành, từ các phương pháp kinh điển như điều khiển PID, điều khiển tối ưu, đến các phương pháp hiện đại như điều khiển mờ, điều khiển thích nghi, và đặc biệt là điều khiển dựa trên học tăng cường và mạng nơ-ron nhân tạo. Xu hướng kết hợp giữa các lý thuyết điều khiển truyền thống với trí tuệ nhân tạo đang mở ra nhiều hướng tiếp cận đầy triển vọng, hứa hẹn nâng cao hơn nữa hiệu năng của hệ thống trong các điều kiện phức tạp của môi trường thực tế.

### 1.2. Tổng quan về học tăng cường

Học tăng cường (Reinforcement Learning - RL) là một lĩnh vực trong học máy tập trung vào cách một tác nhân (agent) học cách đưa ra các quyết định tốt nhất thông qua thử nghiệm và sai sót. Khác với học có giám sát, RL không yêu cầu dữ liệu đầu vào/đầu ra cụ thể, mà thay vào đó, tác nhân tự khám phá môi trường bằng cách thực hiện các hành động và nhận phản hồi dưới dạng phần thưởng.

Học tăng cường dựa trên một mô hình toán học về việc ra quyết định gọi là quá trình quyết định Markov. Mô hình này sử dụng các bước thời gian riêng rẽ. Ở mỗi bước, tác tử thực hiện một hành động mới dẫn đến trạng thái môi trường mới. Tương tự, trạng thái hiện tại được quy cho chuỗi các hành động trước đó.

Thông qua thử và sai trong quá trình di chuyển qua môi trường, tác tử xây dựng một tập hợp các quy tắc hoặc chính sách nếu-thì. Các chính sách này giúp tác tử quyết định thực hiện hành động nào tiếp theo để có phần thưởng tích lũy tối ưu. Tác tử cũng phải lựa chọn giữa việc khám phá thêm môi trường để tìm hiểu phần thưởng cho hành động ở trạng thái mới hoặc chọn các hành động có phần thưởng cao đã biết từ một trạng thái nhất định. Điều này được gọi là đánh đổi giữa khám phá và khai thác.




*Hình 1.1. Minh họa về học tăng cường*

Học tăng cường bao gồm:

- Tác nhân (Agent): Thực hiện hành động và học từ môi trường.
- Môi trường (Environment): Hệ thống mà tác nhân tương tác.
- Trạng thái (State): Biểu diễn thông tin hiện tại của môi trường.
- Hành động (Action): Quyết định mà tác nhân thực hiện.
- Phần thưởng (Reward): Phản hồi từ môi trường dựa trên hành động.

- Chính sách (Policy): Quy tắc hoặc chiến lược tác nhân sử dụng để chọn hành động dựa trên trạng thái.
- Hàm giá trị (Value Function): Đánh giá lợi ích lâu dài từ một trạng thái hoặc cặp trạng thái-hành động.

**b) Mục tiêu**

Tác nhân học cách chọn các hành động để tối đa hóa tổng phần thưởng kỳ vọng:

$$J(\pi) = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{2.1}$$

trong đó:

- $\pi$ là chính sách,
- $r_t$ là phần thưởng tại thời điểm $t_t$,
- $\gamma \in (0,1)$ là hệ số chiết khấu, điều chỉnh tầm quan trọng của các phần thưởng trong tương lai.

### 1.3. Một số thuật toán học tăng cường tiêu biểu

❖ **Giải thuật học tăng cường với hệ tuyến tính**

Giải thuật học tăng cường là một lĩnh vực trong trí tuệ nhân tạo nghiên cứu cách một hệ thống tương tác với môi trường để đạt được mục tiêu hay tối ưu hóa một hàm phần thưởng. Hệ tuyến tính là một loại hệ thống được mô tả bằng các biểu thức tuyến tính, trong đó tín hiệu đầu vào và đầu ra đều là các vector và phép biến đổi giữa chúng là các phép toán tuyến tính.

Hệ thống tuyến tính thường được mô tả bằng phương trình trạng thái trong không gian liên tục:

$$x_{t+1} = A\,x_t + B\,u_t + w_t \tag{2.2}$$

với:

- $x_t$: trạng thái tại thời điểm t,
- $u_t$: hành động (đầu vào điều khiển),
- A, B: ma trận động lực học của hệ,
- $w_t$ : nhiễu (thường được giả định là Gaussian).

Hàm phần thưởng trong bối cảnh học tăng cường thường được định nghĩa dạng bậc hai:

$$r_t = -x_t^\top Q x_t - u_t^\top R u_t \tag{2.3}$$

trong đó $Q \geq 0$, $R > 0$ là các ma trận trọng số cho trạng thái và điều khiển.

Mục tiêu là tìm chính sách $u_t = \pi(x_t)$ tối ưu để tối đa hóa tổng phần thưởng kỳ vọng:

$$J(\pi) = \mathrm{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right] \tag{2.4}$$

với $\gamma \in (0,1)$ là hệ số chiết khấu

* Phương pháp học tăng cường cho hệ tuyến tính: Linear Quadratic Regulator (LQR)

LQR là một bài toán điển hình trong điều khiển tối ưu, với lời giải tối ưu dựa trên phương trình Riccati. Đối với hệ RL, nếu A, B, Q, R đã biết, bài toán có thể được giải bằng:

- **Phương trình Riccati:** Tìm ma trận P thông qua phương trình:

$$P = Q + A^\top P A - A^\top P B \left(R + B^\top P B\right)^{-1} B^\top P A \tag{2.5}$$

Trong đó:

- P là ma trận cần tìm (ma trận Riccati), kích thước là n × n (trong bài toán LQR, n là số chiều của trạng thái).

- A, B là ma trận của hệ động lực học tuyến tính $x_{t+1}=Ax_t+Bu_t$
- Q là ma trận trọng số cho trạng thái $x_t$.
- R là ma trận trọng số cho điều khiển $u_t$.

Phương trình Riccati này mô tả cách ma trận P phụ thuộc vào các yếu tố hệ thống (ma trận A, B) và ma trận trọng số Q, R.

- Chính sách tối ưu:

$$K = (R + B^T PB)^{-1} B^T PA \tag{2.6}$$

Trong RL, khi không biết trước A và B, các thuật toán học sẽ sử dụng dữ liệu thu thập từ môi trường để xấp xỉ K.

- Thuật toán Q-learning

Q-learning là một phương pháp học tăng cường không yêu cầu mô hình môi trường (model-free), hoạt động dựa trên giá trị (value-based). Thay vì sử dụng mô hình động lực của môi trường, thuật toán này cập nhật giá trị của hàm Q thông qua phương trình lặp Bellman. Đối với các thuật toán dựa trên chính sách (policy-based), giá trị Q được ước lượng thông qua một chính sách hiện tại và cải thiện dần theo thời gian cho đến khi đạt được chính sách tối ưu.

Phương trình Bell-man:

$$Q^*(s,a) = E_{s'}\left[ r + \gamma \max_{a'} Q(s',a')^* \;\middle|\; s,a \right] \tag{2.7}$$

Q-learning thuộc nhóm thuật toán off-policy, nghĩa là quá trình cập nhật giá trị của hàm Q không phụ thuộc trực tiếp vào chính sách hiện tại đang được sử dụng. Cụ thể, thuật toán vẫn có thể sử dụng thông tin từ các hành động ngẫu nhiên (exploration) để tối ưu hóa chính sách mục tiêu (optimal

policy). Điều này cho phép chủ thể (agent) khám phá nhiều lựa chọn khác nhau để dần tìm ra chính sách mang lại phần thưởng lớn nhất về lâu dài.

$$Q_{s_t,a_t} = Q_{s_t,a_t} + \alpha^* \left[ r_t + \gamma^* \max_a Q\,(s_t + 1, a) - Q_{s_t,a_t} \right] \quad (2.8)$$

Trong đó:

- Q*(s, a) là giá trị kỳ vọng về phần thưởng tích lũy khi thực hiện hành động a tại trạng thái s, và sau đó tiếp tục tuân theo chính sách tối ưu.
- α là tốc độ học của thuật toán thể hiện sự hội tụ của mô hình học;
- γ là hằng số chiết khấu xác định mức độ quan trọng được trao cho phần thưởng trước mắt và phần thưởng trong tương lai;
- $s_t$ và $a_t$ lần lượt biểu thị trạng thái và hành động được chọn tại thời điểm ttt. Trong Q-learning, hành động từ mỗi trạng thái được chọn thông qua một tiến trình ra quyết định Markov (Markov Decision Process - MDP), với mục tiêu lựa chọn hành động phù hợp nhất để tối ưu hóa phần thưởng tích lũy theo thời gian.

Với bộ giá trị (S, A, P, R, γ), với:

- S là một tập hợp các trạng thái,
- A là tập hợp các hành động mà tác nhân có thể chọn để thực hiện,
- P là Ma trận xác suất chuyển đổi,
- R là Phần thưởng được tích lũy bởi các hành động của tác nhân,
- γ là hệ số chiết khấu.

- Ma trận xác suất chuyển đổi:

$$P_{ss'}^a = P[\, S_{t+1} = s' \mid S_t = s, A_t = a \,] \quad (2.9)$$

- Hàm phần thưởng:

$$R_s^a = E[\, R_{t+1} \mid S_t = s, A_t = a \,] \quad (2.10)$$

Lưu đồ thuật toán Q-learning:

Đầu vào:

Tập trạng thái S = {1, …, $s_n$};

Tập hành động A = {1, …, $a_n$};

Hàm phần thưởng: $S\ x\ A\ \rightarrow\ []$

Khởi tạo các siêu tham số của thuật toán: α, γ ∈ [0,1];

Phương thức:

Khởi tạo Q: $S\ x\ A\ \rightarrow\ []$ ngẫu nhiên;

for giá trị Q chưa hội tụ:

do: đặt trạng thái s ∈ S;

for s không phải là trạng thái cuối:

do:

Lựa chọn hành động a mới (dựa vào chính sách tối ưu); Thực hiện hành động a;

Quan sát trạng thái s mới và thu nhận phần thưởng R;

Cập nhật;

$$Q_{s_t,a_t} = Q_{s_t,a_t} + \alpha^*\left[r_t + \gamma^* \max Q\ (s_t + 1, a) - Q_{s_t,a_t}\right] \quad (2.11)$$

Cập nhật trạng thái $s\ \rightarrow\ s$';

End for

End for

Đầu ra:

Hành động tốt nhất được lựa chọn a'.

Trong lý thuyết về giải thuật học tăng cường với hệ tuyến tính, mục tiêu là tìm ra các phương pháp và thuật toán để điều chỉnh các thông số của hệ tuyến tính sao cho hệ thống đạt được hiệu suất tối ưu trong môi trường tương tác. Một

phương pháp chính để đạt được điều này là sử dụng phương pháp Gradient Descent kết hợp với các kỹ thuật học tăng cường như Q-Learning, Actor-Critic, hoặc Deep Q-Network (DQN).

Trong quá trình huấn luyện, hệ tuyến tính được cung cấp với các tín hiệu đầu vào từ môi trường và tạo ra các tín hiệu đầu ra dựa trên các thông số hiện tại. Các phương pháp học tăng cường được sử dụng để điều chỉnh các thông số này dựa trên phần thưởng nhận được từ môi trường. Phần thưởng này có thể được định nghĩa thông qua một hàm phần thưởng, mục tiêu là tối đa hóa giá trị kỳ vọng của hàm phần thưởng này.

Sau khi hệ tuyến tính được huấn luyện, nó có thể được sử dụng để thực hiện các nhiệm vụ trong môi trường tương tác. Quá trình này có thể liên tục được lặp lại để cải thiện hiệu suất của hệ tuyến tính theo thời gian.

Giải thuật học tăng cường với hệ tuyến tính nghiên cứu cách sử dụng các phương pháp học tăng cường như Q-Learning, Actor-Critic, hoặc DQN để tối ưu hóa hệ tuyến tính trong môi trường tương tác. Điều này có thể đạt được thông qua việc sử dụng các phương pháp tối ưu hóa như Gradient Descent và các phương pháp huấn luyện phù hợp.

❖ **Giải thuật học tăng cường với hệ phi tuyến tính**

Giải thuật học tăng cường với hệ phi tuyến tính liên quan đến việc áp dụng các phương pháp giải thuật học tăng cường để tối ưu hóa hệ thống phi tuyến tính.

Trong giải thuật học tăng cường, mục tiêu là tìm cách tối ưu hóa hệ thống tương tác với môi trường để đạt được mục tiêu hoặc tối đa hoá hàm phần thưởng. Hệ phi tuyến tính là một loại hệ thống trong đó tín hiệu đầu vào và/hoặc đầu ra không tuân theo quy tắc tuyến tính. Thay vào đó, chúng được biểu diễn

bằng các hàm phi tuyến tính như các hàm phi tuyến tính, hàm phi tuyến tính đa tầng (multilayer), hoặc các mô hình phi tuyến khác.

---

Giải thuật VI (Value Iteration)

---

Giải thuật VI sau đây mô tả chi tiết các bước xấp xỉ trực tiếp hàm đánh giá tối ưu $V^*(xk)$ .

Khi có $V^*(xk)$, luật điều khiển tối ưu $u^*(x)$ được xấp xỉ

Bước 1: $\forall xk \in \Omega x$, khởi tạo $V^0(xk) = 0$, Gán $i = 0$

Bước 2: Xấp xỉ hàm đánh giá:

- $i \leftarrow i + 1$
- Lặp vòng $\forall xk \in \Omega x$; Cập nhật:

$$V^i(xk) = min\forall u \in U(xk)\,[r(xk,u) + \gamma V^{i-1}(f(xk,u))] \quad (2.12)$$

- Nếu thỏa tiêu chuẩn hội tụ sao cho $||V^i - V^{i-1}|| \leq \delta$ với $\delta$ là số dương đủ nhỏ, thì gán $V^*(xk) = V^i(xk)$ , $\forall xk \in \Omega x$ sau đó thực hiện Bước 3, ngược lại quay về Bước 2.

Bước 3: Xấp xỉ luật điều khiển tối ưu:

- Lặp vòng $\forall xk \in \Omega x$; Cập nhật:

$$u^*(xk) = argmin\forall u \in U(xk)\,[r(xk,u(xk)) + \gamma V^*(f\,xk,u(xk))] \quad (2.13)$$

---

- Giải thuật PI (Policy Iteration)

Giải thuật PI khởi động sử dụng luật điều khiển ổn định, sau đó xấp xỉ hàm đánh giá trong một bước và cải thiện luật điều khiển dựa vào hàm đánh giá vừa xấp xỉ ở bước tiếp theo. Các bước trong giải thuật PI được mô tả như sau:

---

Bước 1: $\forall xk \in \Omega x$ , khởi tạo luật điều khiển ổn định $u^0(xk)$

Gán $i = 0$

Bước 2: Xấp xỉ hàm đánh giá:

- Lặp vòng $\forall xk \in \Omega x$; Khởi tạo $V^0(xk) = 0$

Bước 3: Xấp xỉ hàm đánh giá ở bước $i + 1$ sử dụng luật điều khiển $ui$:

- $i \leftarrow i + 1$
- Lặp vòng $\forall xk \in \Omega x$; Cập nhật:

$$V^i(xk) = r(xk, u^{i-1}(xk)) + \gamma(V^{i-1} f(xk, u^{i-1}(xk)) \quad (2.14)$$

Bước 4: Xấp xỉ luật điều khiển tối ưu:

- Lặp vòng $\forall xk \in \Omega x$; Cập nhật:

$$u(i)(xk) = argmin \forall a \in U(x) [r(xk, a) + \gamma V^i(f(xk, a))]$$

- Nếu thỏa tiêu chuẩn hội tụ sao cho $\|V^i - V^{i-1}\| \leq \delta$ với $\delta$ là số dương đủ nhỏ thì gán $u^*(xk) = u^i(xk)$ và $V^*(xk) = V^i(xk)$ , kết thúc giải thuật, ngược lại quay về Bước 3.

---

- Giải thuật Adaptive Dynamic Programming (ADP)

Adaptive Dynamic Programming (ADP) là một nhánh của Học tăng cường (Reinforcement Learning – RL), xuất phát từ lý thuyết điều khiển tối ưu, nhằm giải quyết các bài toán điều khiển trong môi trường phi tuyến, không biết mô hình hệ thống, hoặc biến thiên theo thời gian.

ADP đặc biệt mạnh trong việc xấp xỉ hàm giá trị và học chính sách điều khiển cho hệ thống liên tục, thường là hệ điều khiển phi tuyến như robot, UAV, hoặc động học xe.

Tối ưu hóa hàm chi phí tích lũy:

$$J(s_0) = \sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \tag{2.15}$$

Trong đó:

- $s_t$: trạng thái
- $a_t$: hành động (đầu vào điều khiển)
- $r(s, a)$: phần thưởng hoặc chi phí (reward/cost)
- $\gamma \in [0,1)$: hệ số chiết khấu

Trong điều khiển tối ưu, ta có phương trình Bellman:

$$V(s) = \max_a \left[ r(s, a) + \gamma V\big(f(s, a)\big) \right] \tag{2.16}$$

Trong ADP, ta không biết hàm f(s, a) (động học hệ thống), cũng không biết hàm V(s) (hàm giá trị), vì vậy cần xấp xỉ cả hai.

ADP sử dụng 2 thành phần chính:

- Critic: Ước lượng hàm giá trị V(s) dùng để đánh giá chất lượng của một trạng thái.
- Actor: Học chính sách tối ưu $\boldsymbol{\pi(s) = a}$ là hàm sinh hành động tốt nhất tại mỗi trạng thái.

Các bước trong giải thuật ADP được mô tả như sau:

---

Bước 1: Mô tả bài toán điều khiển

Hệ thống có động học:

$$s_{t+1} = f(s_t, a_t) \tag{2.17}$$

Không biết chính xác hàm $f$, có thể đo được $s_t$, $a_t$, và $r_t$.

Bước 2: Xây dựng mạng nơ-ron xấp xỉ:

- Critic: học hàm giá trị V(s)

- Actor: học chính sách $a = \pi(s)$

Bước 3: Huấn luyện

- Sử dụng gradient descent để cập nhật Critic và Actor:

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t) \quad (2.18)$$

Critic học để giảm TD-error.

Actor cập nhật chính sách theo hướng cải thiện Critic.

---

- Giải thuật Fitted Value Iteration (FVI)

Fitted Value Iteration (FVI) là một biến thể của thuật toán Value Iteration truyền thống trong Reinforcement Learning (RL) và Dynamic Programming (DP). Mục đích là ước lượng hàm giá trị tối ưu V*(s) trong các hệ thống có không gian trạng thái lớn hoặc liên tục, không thể dùng bảng tra cứu (lookup table) được.

FVI sử dụng hồi quy (regression) với một bộ hàm xấp xỉ (ví dụ: mạng nơ-ron, RBF, cây quyết định, kernel methods) để học hàm giá trị từ tập dữ liệu mẫu.

Phương pháp này thuộc nhóm batch RL (học ngoại tuyến), phù hợp khi ta có tập dữ liệu quan sát trạng thái – hành động – trạng thái tiếp theo – phần thưởng.

Với Mục tiêu: tìm hàm giá trị tối ưu V*(s) thỏa mãn phương trình Bellman:

$$V^*(s) = \max_a E_{s'} [r(s,a) + \gamma V^*(s')] \quad (2.19)$$

Trong đó:

- Trạng thái $s \in S$ (liên tục hoặc lớn)

- Hành động $a \in A$
- Phần thưởng $r(s, a)$
- Động học ẩn $P(s' \mid s, a)$ (có thể không biết)

FVI giải quyết bằng cách:

- Lấy mẫu (sampling): Thu thập một tập mẫu $\mathcal{D} = \{(s_i, a_i, r_i, s_i')\}_{i=1}^{N}$ từ môi trường hoặc mô phỏng.
- Cập nhật giá trị trạng thái mẫu:
  Tính giá trị mục tiêu (target) cho mỗi mẫu:

$$y_i = r_i + \gamma \max_{a'} \widehat{V_k}(s_i') \tag{2.20}$$

- $\widehat{V_k}$ là ước lượng hàm giá trị ở bước lặp thứ k.
- Hồi quy (regression):
  Tìm hàm $\widehat{V_{k+1}}(s)$ sao cho gần nhất với các cặp ($s_i$, $y_i$):

$$\widehat{V_{k+1}} = \arg\min_{V \in \mathcal{F}} \sum_{i=1}^{N} (V(s_i) - y_i)^2 \tag{2.21}$$

- $\mathcal{F}$ là không gian hàm xấp xỉ (ví dụ: mạng nơ-ron).
- Lặp lại quá trình cho đến khi hội tụ hoặc đủ số bước.

Giải thuật học tăng cường với hệ phi tuyến tính nghiên cứu cách sử dụng các phương pháp và thuật toán giải thuật học tăng cường để tối ưu hóa hệ thống phi tuyến tính. Các mạng nơ-ron nhân tạo, thuật toán tối ưu hóa không gian trạng thái và các thuật toán học tăng cường không mô hình là những công cụ quan trọng trong lĩnh vực này.

### 1.4. Ứng dụng học tăng cường trong lĩnh vực xe tự hành

Trí tuệ nhân tạo (AI) đang mở ra những bước tiến vượt bậc trong lĩnh vực điều khiển robot và xe tự hành, biến những khái niệm tưởng chừng chỉ có trong khoa học viễn tưởng thành hiện thực. Nhờ vào sự phát triển nhanh chóng

của các thuật toán học máy (Machine Learning - ML), học sâu (Deep Learning - DL), thị giác máy tính (Computer Vision) và các cảm biến thông minh, AI đã trở thành yếu tố cốt lõi giúp các hệ thống robot và xe tự hành hoạt động một cách linh hoạt, tự chủ và an toàn trong môi trường thực tế.

Một trong những cơ hội lớn nhất của AI trong lĩnh vực này là khả năng giúp các hệ thống robot và phương tiện tự hành hiểu và thích nghi với môi trường xung quanh. Các thuật toán AI cho phép robot và xe tự hành xử lý thông tin từ nhiều nguồn cảm biến khác nhau như camera, radar, lidar và GPS để xây dựng bản đồ môi trường, xác định vật cản, nhận dạng tín hiệu giao thông, và dự đoán hành vi của con người hoặc các đối tượng khác. Nhờ đó, robot và xe tự hành có thể ra quyết định kịp thời, như tránh chướng ngại vật, điều chỉnh tốc độ, hay lựa chọn tuyến đường tối ưu để di chuyển an toàn và hiệu quả.

AI còn đóng vai trò quan trọng trong việc tăng cường khả năng tự học và thích nghi của hệ thống điều khiển robot và xe tự hành. Thông qua các kỹ thuật học tăng cường (Reinforcement Learning), robot có thể học hỏi từ các hành động trước đó để cải thiện hiệu suất hoạt động và thích nghi tốt hơn với các tình huống chưa từng gặp. Trong khi đó, xe tự hành sử dụng các mô hình học sâu để nhận diện và phân tích hình ảnh thời gian thực, giúp xe có thể hoạt động ổn định trong các điều kiện thời tiết và giao thông phức tạp, từ đô thị đông đúc đến các tuyến đường nông thôn hiểm trở.

Không chỉ dừng lại ở khả năng điều hướng và vận hành, AI còn tạo cơ hội cho tự động hóa các nhiệm vụ phức tạp mà trước đây chỉ con người mới có thể thực hiện. Trong lĩnh vực công nghiệp và dịch vụ, robot được điều khiển bằng AI có thể thực hiện các công việc có độ chính xác cao như lắp ráp linh kiện, kiểm tra chất lượng sản phẩm, hoặc phục vụ trong nhà hàng và bệnh viện. AI cho phép các robot này nhận diện đối tượng, tương tác với con người thông

qua xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP), và học hỏi từ hành vi người dùng để nâng cao trải nghiệm và hiệu suất làm việc.

Bên cạnh đó, trong lĩnh vực giao thông vận tải, AI đang mở ra một tương lai đầy tiềm năng với các hệ thống xe tự hành, giúp giảm thiểu tai nạn giao thông, giảm tải cho cơ sở hạ tầng giao thông hiện tại và nâng cao hiệu quả di chuyển. Xe tự hành có khả năng phân tích dữ liệu từ hệ thống giao thông theo thời gian thực, đồng thời phối hợp với các phương tiện khác thông qua mạng lưới thông minh (Vehicle-to-Vehicle - V2V và Vehicle-to-Infrastructure - V2I). Việc này giúp các phương tiện tự động hóa không chỉ di chuyển an toàn mà còn hỗ trợ giảm ùn tắc và tiết kiệm nhiên liệu.

AI cũng mang lại cơ hội to lớn trong việc phát triển robot hỗ trợ trong các lĩnh vực đặc thù như quân sự, cứu hộ và không gian. Robot điều khiển bằng AI có thể thực hiện các nhiệm vụ nguy hiểm thay con người như rà phá bom mìn, tìm kiếm nạn nhân trong khu vực thảm họa, hoặc tham gia vào các nhiệm vụ thăm dò ngoài không gian. Những hệ thống này được trang bị khả năng ra quyết định độc lập, phân tích môi trường trong điều kiện khắc nghiệt và phản ứng nhanh với các tình huống bất ngờ.

Để xây dựng một hệ thống điều khiển robot và xe tự hành hiệu quả, các kỹ thuật AI như thị giác máy tính, xử lý ngôn ngữ tự nhiên, biểu diễn và suy luận tri thức (Knowledge Representation and Reasoning - KRR), hệ thống chuyên gia (Expert Systems - ES), cùng với học tăng cường và học không giám sát, được tích hợp để nâng cao năng lực nhận thức và hành động của máy móc. Các công nghệ này giúp robot và phương tiện tự hành không chỉ "nhìn thấy" và "hiểu" thế giới xung quanh mà còn ra quyết định và học hỏi giống như con người.

Sự kết hợp giữa công nghệ phần cứng tiên tiến và các giải pháp phần mềm điều khiển thông minh đang tạo nên một hệ sinh thái robot và phương tiện tự

động ngày càng hoàn thiện. Trong tương lai không xa, "Robot và xe tự hành do AI điều khiển" sẽ không chỉ xuất hiện trong các nhà máy hoặc cơ sở nghiên cứu, mà còn trở thành một phần thiết yếu trong cuộc sống thường nhật, từ giao thông đô thị đến chăm sóc cá nhân và các dịch vụ công cộng.

Ứng dụng của học tăng cường rất đa dạng và có thể áp dụng trong nhiều lĩnh vực khác nhau:

- Robotics cho tự động hóa công nghiệp
- Công cụ tóm tắt văn bản, tác nhân đối thoại (văn bản, lời nói), giao diện trò chơi
- Ô tô tự lái
- Học máy và xử lý dữ liệu
- Hệ thống đào tạo sẽ đưa ra các hướng dẫn và tài liệu tùy chỉnh theo yêu cầu của học viên
- Bộ công cụ AI, sản xuất, ô tô, chăm sóc sức khỏe
- Điều khiển máy bay và điều khiển chuyển động Robot
- Xây dựng trí tuệ nhân tạo cho game máy tính

### 1.5. Kết luận chương 1

Trong Chương 1, luận án đã trình bày các cơ sở lý thuyết nền tảng phục vụ cho việc nghiên cứu và phát triển giải thuật điều khiển học tăng cường áp dụng cho xe tự hành. Trước hết, các khái niệm cơ bản về mô hình động học và động lực học của xe đã được giới thiệu, qua đó làm rõ những đặc thù phi tuyến và bất định của hệ thống. Tiếp đến, các nguyên lý cốt lõi của học tăng cường, bao gồm khái niệm trạng thái, hành động, phần thưởng và chính sách, được trình bày nhằm cung cấp cái nhìn tổng quan về cách thức mà thuật toán học tăng cường có thể học và tối ưu hoá điều khiển

# CHƯƠNG 2 - MÔ HÌNH BÀI TOÁN VÀ GIẢI PHÁP ĐỀ XUẤT

## 2.1. Mô hình toán học của xe tự hành

Để theo dõi các quỹ đạo mong muốn của robot di động bánh xe (WMR) với hướng tiến thay đổi theo thời gian, một thuật toán điều khiển bám dựa trên học tăng cường kết hợp mạng nơ-ron thích nghi được đề xuất cho hệ thống động lực phi tuyến rời rạc (DT) của WMR có hiện tượng trượt và lết bánh. Mô hình điển hình được chuyển đổi thành hệ thống phi tuyến rời rạc dạng affine, và ràng buộc của mô-men xoắn đầu vào kết hợp của robot được mở rộng thành đầu vào điều khiển vùng chết giả (PDZ).

Mạng nơ-ron (NNs) được giới thiệu làm mạng hành động để xấp xỉ các thành phần không xác định trong mô hình, bao gồm hiện tượng trượt, lết bánh và thành phần PDZ, trong khi một mạng nơ-ron khác được sử dụng làm mạng phê bình để xấp xỉ hàm tiện ích chiến lược. Sau đó, các quy luật thích nghi của mạng nơ-ron hành động và mạng nơ-ron phê bình được thiết kế thông qua phương pháp thích nghi dựa trên gradient tiêu chuẩn.

Tính bị chặn cuối cùng đồng nhất (UUB) của tất cả các tín hiệu trong hệ thống WMR phi tuyến rời rạc dạng affine được đảm bảo, đồng thời lỗi bám hội tụ về một tập nhỏ gần zero. Các mô phỏng số được thực hiện để xác nhận tính hiệu quả của phương pháp đề xuất.

Robot là một thiết bị tự động được thiết kế để thực hiện các hành vi của con người hoặc thay thế các chức năng cơ học. Dựa trên không gian công việc, robot có thể được phân loại thành robot mặt đất, trên không và dưới nước. Robot di động bánh xe (WMR), một loại robot di động mặt đất điển hình, đã được ứng dụng rộng rãi trong đời sống xã hội, nghiên cứu khoa học và các ứng dụng kỹ thuật.

Với sự phát triển của xã hội loài người, môi trường hoạt động và các yêu cầu nhiệm vụ của WMR ngày càng trở nên phức tạp, xét trên các khía cạnh về chức năng, cấu trúc tổ chức và thiết kế thuật toán điều khiển. Do đó, WMR đang đối mặt với những thách thức đáng kể, thu hút sự quan tâm của đông đảo các nhà nghiên cứu [1–10]. Tuy nhiên, các thuật toán điều khiển hiện tại vẫn chưa đủ đáp ứng các yêu cầu nhiệm vụ ngày càng phức tạp, dẫn đến nhu cầu cấp thiết về các thuật toán mới, điều này cũng đã nhận được sự chú ý của cộng đồng học thuật.

Xét WMR là một hệ thống phi tuyến điển hình, nhiều thuật toán điều khiển bám đã được đề xuất trong vài thập kỷ qua. Phương pháp tuyến tính hóa đầu vào-đầu ra cải tiến đã được đề xuất trong [11], và điều khiển bám dựa trên dữ liệu cho WMR đã được thực hiện trong [12–15]. Thuật toán bám dựa trên chế độ trượt cho hai loại WMR khác nhau được trình bày trong [16–18]. Để vượt qua vấn đề này, các thuật toán hiệu quả đã được đề xuất, bao gồm điều khiển bám phi tuyến dựa trên bộ quan sát trạng thái mở rộng và tránh chướng ngại vật trong [16], thuật toán bám mạnh mẽ dựa trên bộ quan sát trạng thái mở rộng tổng quát trong [17], và phương pháp điều khiển bám mạnh mẽ dựa trên thiết kế lại Lyapunov trong [18].

Song song với các thuật toán điều khiển truyền thống, cùng với sự ứng dụng rộng rãi của mạng nơ-ron (NN) [19–20], hệ thống logic mờ (FLS) [16–18] và mạng nơ-ron mờ (FNN) [29], thuật toán điều khiển thông minh thích nghi đã được phát triển để sử dụng lâu dài. Một bộ điều khiển quan sát trạng thái mờ thích nghi được thiết kế trong [30] cho các hệ thống rời rạc phi tuyến với các hàm không xác định và nhiễu bị chặn. Để đạt được cân bằng động ổn định và điều khiển bám, bộ điều khiển mạng nơ-ron thích nghi dựa trên bù động tuyến tính phản hồi đầu ra đã được thiết kế trong [21]. Cấu trúc điều khiển dựa

trên mạng nơ-ron được đề xuất trong [22] cung cấp cơ hội để xây dựng luật điều khiển kết hợp động học/mô-men cho các hệ thống robot phi holonomic.

Trong những năm gần đây, các thuật toán điều khiển dựa trên NN cho nhiều hệ thống robot đã đạt được những kết quả đáng kể, đặc biệt là trong nghiên cứu các thuật toán điều khiển cho WMR. Dựa trên phương pháp backstepping, [25] đã đề xuất một điều khiển thích nghi mới cho máy bay trực thăng tự động với bộ nhận diện nơ-ron mạnh mẽ mới. Hơn nữa, các ràng buộc phi holonomic của WMR có thể được xử lý hiệu quả bởi thuật toán điều khiển dựa trên NN. Một sơ đồ điều khiển bám mạnh mẽ dựa trên bộ quan sát nhiễu đã được thiết kế trong [10], và một điều khiển bám thích nghi được đề xuất trong [28] cho hệ động học phi tuyến của WMR có hiện tượng trượt và lết bánh với mô-men bị giới hạn.

Trong một thuật toán điều khiển hệ cơ học điển hình, mục tiêu tối ưu như hiệu quả bám tốt nhất hoặc tối ưu hóa hiệu suất tổng thể là điều cần đạt được. Tuy nhiên, các thuật toán được đề xuất hiện tại chưa đủ để đáp ứng mục tiêu này. Để khắc phục, các thuật toán điều khiển tối ưu đã được đề xuất, và điều khiển tối ưu dựa trên học tăng cường (RL) được chứng minh là một phần mở rộng quan trọng trong [19], mang lại những kết quả hữu ích. Hai loại phương pháp điều khiển tối ưu dựa trên RL cho hệ thống phi tuyến đa đầu vào đa đầu ra (MIMO) được trình bày trong [20,21].

Thuật toán RL tích phân dựa trên lặp chính sách được giới thiệu trong [22]. Để đạt được điều khiển phê bình cho hệ MIMO affine rời rạc với nhiễu bị giới hạn, một thuật toán điều khiển dựa trên bộ xấp xỉ RL trực tuyến được trình bày trong [23]. Một điều khiển bám phản hồi đầu ra dựa trên NN mới được đề xuất trong [24] nhằm mục đích bám các hệ phi tuyến rời rạc phản hồi nghiêm ngặt MIMO với các ràng buộc biên độ. Tương ứng, bộ điều khiển bám thích

nghi NN dựa trên RL mới được thiết kế trong [25] cho hệ phi tuyến rời rạc phản hồi không nghiêm ngặt.

Xét các thuật toán dựa trên RL trên đã cho thấy hiệu quả cao đối với hệ phi tuyến, nhiều thuật toán điều khiển thông minh dựa trên RL cho WMR đã được thực hiện trong những năm gần đây.

Tài liệu tham khảo [26] nhấn mạnh rằng phương pháp RL mở ra khả năng thiết kế các hành vi phức tạp và khó thực hiện cho robot, và việc ứng dụng các thuật toán RL vào các vấn đề robot mang lại cảm hứng, tác động và xác minh cho sự phát triển của RL. Tài liệu tham khảo [27] đề xuất một thuật toán điều khiển bám tích hợp động học và động lực dựa trên RL cho WMR phi holonomic, trong khi [28] giới thiệu một thuật toán bám đường tối ưu phân cấp dựa trên RL mới cho WMR. Các thuật toán này có hiệu quả tích cực đối với WMR không có hiện tượng trượt và lết bánh, tuy nhiên, trong các ứng dụng kỹ thuật thực tế, WMR thường đi kèm với hiện tượng trượt và lết bánh trong quá trình hoạt động, dẫn đến mất hiệu quả của các thuật toán trên.

Để khắc phục hạn chế này, một thuật toán điều khiển bám thích nghi mới dựa trên mạng nơ-ron học tăng cường (RLNN) được đề xuất trong nghiên cứu này. Trong quá trình điều khiển, mô hình WMR được chuyển đổi thành hệ rời rạc affine, trong đó hiện tượng trượt và lết bánh không xác định được xử lý riêng biệt, và các ràng buộc mô-men đầu vào của robot được mở rộng thành đầu vào điều khiển vùng chết giả (PDZ). Ba mạng NN được sử dụng làm mạng hành động để thích nghi với các thành phần không xác định, bao gồm trượt, lết bánh và vùng chết giả, trong khi một mạng NN khác được sử dụng làm mạng phê bình để xấp xỉ hàm tiện ích chiến lược. Các luật thích nghi được xác định thông qua phương pháp hạ gradient, và tính bị chặn của tất cả tín hiệu trong hệ WMR được đảm bảo bằng định lý ổn định Lyapunov.

Hiện tượng trượt và lết bánh không xác định được xấp xỉ riêng lẻ bằng các mạng hành động NN, thay vì giả định là nhiễu bị chặn. Phương pháp này giúp cải thiện độ chính xác của hệ WMR. Ràng buộc của mô-men đầu vào của robot được mở rộng thành PDZ, xử lý bằng kỹ thuật vùng chết.

Mô hình Đồ thị địa điểm (Graph-based Planning): Sử dụng đồ thị để lập kế hoạch đường đi và tránh vật cản.

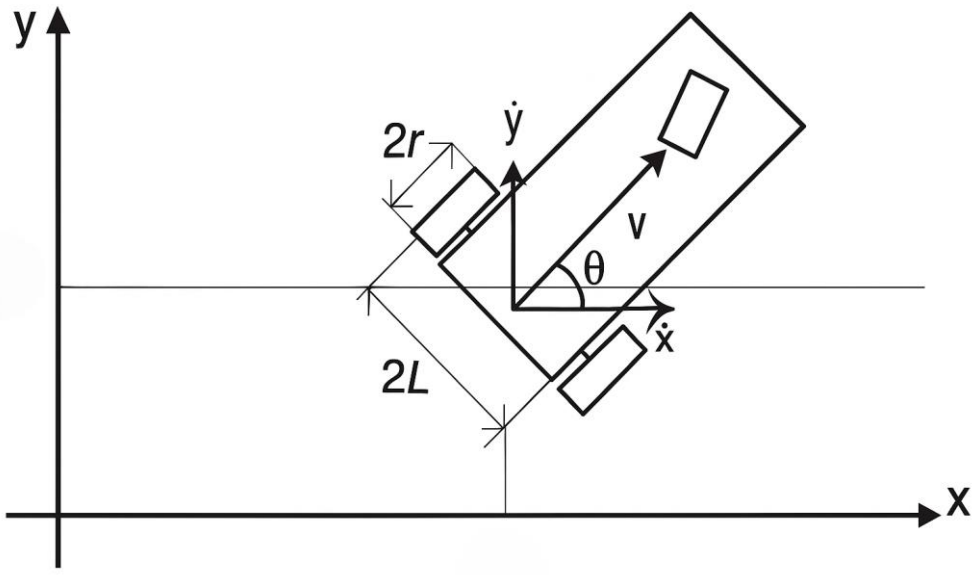
Mô hình Trajectory Planning: Xác định chuỗi các điểm trong không gian thời gian để điều khiển xe một cách an toàn và hiệu quả.

Mô hình Tương tác với Người lái (Human Interaction Model):

Mô hình Giao tiếp Người-Máy: Để dự đoán và phản ứng đúng đắn với hành vi của người lái và các phương tiện xung quanh.

- Mô hình Hệ thống (System Model):

Kết nối và Giao tiếp: Mô tả cách các thành phần của hệ thống (điều khiển, cảm biến, động cơ) tương tác với nhau.




*Hình 2.1. Mô hình xe tự hành*

Mỗi mô hình có thể được biểu diễn bằng các phương trình toán học và các thuật toán tương ứng. Việc tích hợp những mô hình này giúp xe tự hành có khả năng nhận biết môi trường xung quanh, lập kế hoạch, và thực hiện các hành động lái xe một cách an toàn và hiệu quả.

### 2.1.1. Mô hình động học tuyến tính

Trong học tăng cường (RL), mục tiêu cốt lõi của tác tử là thu nhận tín hiệu phản hồi từ môi trường, được gọi là giá trị thưởng hoặc phần thưởng. Tại mỗi thời điểm, tác tử nhận được một giá trị thưởng $t_r \in R$, còn được hiểu là giá trị phản hồi. Mục tiêu không chính thức của tác tử là tối đa hóa tổng phần thưởng tích lũy từ môi trường. Điều này đồng nghĩa với việc tối ưu hóa không chỉ dựa vào phần thưởng nhận được ngay tại thời điểm hiện tại mà còn xét đến phần thưởng tích lũy trong tương lai. Nói cách khác, tác tử hướng tới việc tối đa hóa tổng giá trị thưởng dự kiến trong dài hạn.

Trong nhiều trường hợp, đặc biệt là trong các bài toán quyết định Markov, giá trị phản hồi $t_r$ thường được mô hình hóa dưới dạng hàm số của mục tiêu. Các nghiên cứu phổ biến sử dụng một biểu thức tổng trọng số để biểu diễn giá trị phần thưởng này.

Đối với các bài toán có chuỗi hành động kéo dài vô hạn, người ta sử dụng hệ số chiết khấu γ, với$0 \leq \gamma \leq 1$, nhằm làm giảm dần giá trị của các phần thưởng trong tương lai. Khi đó, tổng giá trị mục tiêu được biểu diễn theo công thức chiết khấu như sau:

$$G_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \quad (2.22)$$

Hệ số γ cho phép xác định mức độ ảnh hưởng của những bước chuyển trạng thái tiếp theo đến giá trị phản hồi tại thời điểm đang xét. Giá trị của γ cho phép điều chỉnh giai đoạn Tác tử lấy các hàm tăng cường. Nếu γ gần 0, thì Tác tử chỉ

xem xét mục tiêu gần nhất, giá trị γ càng gần với 1 thì Tác tử sẽ quan tâm đến các mục tiêu xa hơn trong tương lai. Như vậy, thực chất bài toán quyết định Markov trong trường hợp này chính là việc lựa chọn các hành động để làm cực đại biểu thức (2.22).

Trong mọi trạng thái $s_t$, một Tác tử lựa chọn một hành động dựa theo một chính sách điều khiển π: $a_t = \pi(s_t)$. Hàm giá trị tại một trạng thái của hệ thống được tính bằng kỳ vọng toán học của hàm phản hồi theo thời gian. Hàm giá trị là hàm của trạng thái và xác định mức độ thích hợp của chính sách điều khiển π đối với Tác tử khi hệ thống đang ở trạng thái s. Hàm giá trị của trạng thái s trong chính sách π được tính như sau:

$$v_\pi(s) = E_\pi\{\, G_t \mid S_t = s \,\} = E_\pi[\, \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s \,] \qquad (2.23)$$

Trong đó, ký hiệu $E_\pi[.]$ được sử dụng để biểu diễn giá trị kỳ vọng của một biến ngẫu nhiên mà tác tử thu được khi thực hiện theo chính sách π tại bất kỳ thời điểm t nào.

$$\pi^* = \{\pi_0(s_0),\ \pi_1(s_1),\ \ldots,\ \pi_{N-1}(s_{N-1})\} \qquad (2.24)$$

Bài toán tối ưu trong học tăng cường được xây dựng với mục tiêu xác định chính sách điều khiển tối ưu, ký hiệu là $\pi^*$, nhằm cực đại hóa hàm giá trị tương ứng với trạng thái của hệ thống sau một số bước hữu hạn hoặc trong trường hợp chuỗi thời gian kéo dài vô hạn.

Sử dụng các phép biến đổi:

$$v_\pi(s) = E_\pi\{\, G_t \mid S_t = s \,\} = E_\pi[\, \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid S_t = s \,] \qquad (2.25)$$

Một chính sách tối ưu, kí hiệu $\pi^*$, sẽ là cho giá trị thưởng lớn nhất, hay:

$$v_{\pi^*}(s) \geq v_\pi(s) \qquad (2.26)$$

$$\pi^* = \text{argmax}_\pi\{v_\pi(s)\} \qquad (2.27)$$

Để thuận tiện trong ký hiệu, ta có thể viết gọn $v_* = v_{\pi^*}$. Khi đó, hàm giá trị tối ưu của một trạng thái bất kỳ được xác định như sau:

$$v_*(s) = \max_{\pi}\{ v_\pi(s)\} \quad (2.28)$$

Biểu thức trên được gọi là phương trình tối ưu Bellman, hay còn được biết đến dưới tên gọi phương trình quy hoạch động. Hàm giá trị $v_\pi$ mô tả giá trị kỳ vọng của một trạng thái khi tác tử tuân theo chính sách π. Thông thường, trạng thái kết thúc (terminal state) được quy ước có giá trị bằng 0. Song song với khái niệm trên, hàm $Q_\pi(s,a)$ được định nghĩa là giá trị kỳ vọng của việc thực hiện một hành động a tại trạng thái s, theo chính sách π.

Giá trị này được tính toán dựa trên giá trị trung bình theo kỳ vọng của tổng phần thưởng tích lũy khởi đầu từ trạng thái s, khi hành động đầu tiên là a, sau đó tiếp tục theo chính sách π.

$$Q_\pi(s,a) = E_\pi\{ G_t \mid s_t = s, a_t = a \} = E_\pi\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \} \quad (2.29)$$

Hàm $Q_\pi$ được gọi là hàm giá trị hành động tương ứng với chính sách π. Tương tự như hàm giá trị trạng thái $v_\pi$, hàm $Q_\pi$ có thể được ước tính thông qua dữ liệu kinh nghiệm thu thập từ quá trình tác tử tương tác với môi trường. Trong trường hợp sử dụng thuật toán Q-learning, giá trị hàm hành động được cập nhật bằng công thức sau:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\left[R_{t+1} + \gamma \max_{a_{t+1}} Q\,(s_{t+1},a_{t+1}) - Q(s_t,a_t)\right] \quad (2.30)$$

Trong công thức trên, γ đại diện cho hệ số chiết khấu, dùng để điều chỉnh mức độ ảnh hưởng của các phần thưởng trong tương lai lên giá trị Q hiện tại. Tham số α được gọi là hệ số học, đóng vai trò điều chỉnh tốc độ học tập hoặc mức độ cập nhật giá trị trong suốt quá trình huấn luyện. Thông thường, tỷ lệ

học tập α có giá trị trong khoảng [0,1], cho phép kiểm soát mức độ mà thông tin mới thay thế thông tin cũ. Khi α = 1 toàn bộ thông tin mới sẽ ghi đè hoàn toàn lên thông tin cũ, tức là tác tử chỉ chú trọng vào dữ liệu vừa thu thập được. Ngược lại, khi α nhỏ, việc cập nhật diễn ra chậm hơn, kết hợp thông tin mới với thông tin lịch sử.

Đối với hệ số chiết khấu $\gamma \in [0,1]$, nó phản ánh mức độ ưu tiên phần thưởng trong tương lai của tác tử. Nếu $\gamma = 0$, tác tử chỉ quan tâm đến phần thưởng ngay lập tức, bỏ qua mọi phần thưởng về sau. Ngược lại, khi γ tiến gần tới 1, tác tử sẽ đánh giá cao những phần thưởng dài hạn, khuyến khích việc tối ưu hóa các hành động để tối đa hóa lợi ích lâu dài.

### 2.1.2. Mô hình phi tuyến bất định

Khi xét đến mô hình có tính phi tuyến và bất định trong học tăng cường của xe tự hành, một phương pháp phổ biến là sử dụng học tăng cường dựa trên mô hình (model-based reinforcement learning) kết hợp với các kỹ thuật tối ưu hóa.

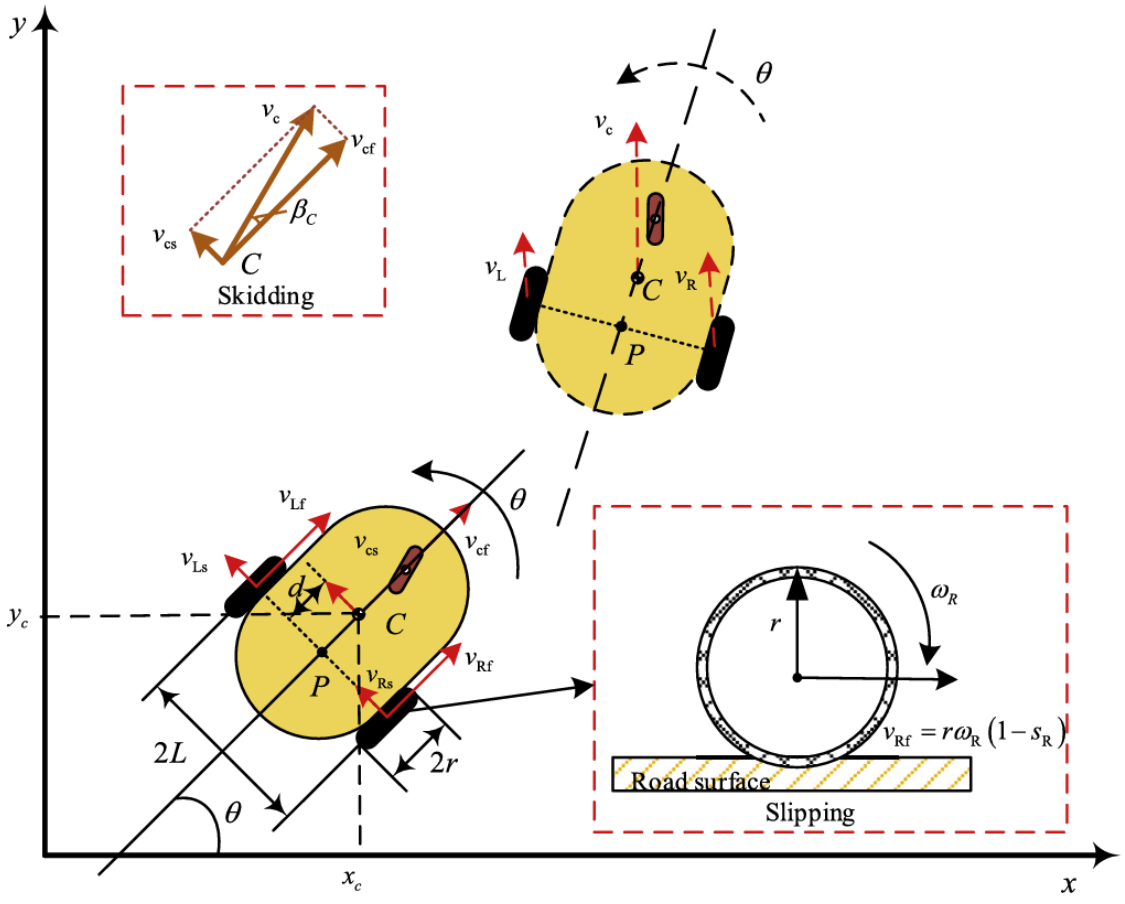
Như thể hiện trong Hình 2.2, hệ thống phi tuyến tính mà tôi đã chọn trong nghiên cứu này là một hệ thống động lực học phi toàn thể của hai WMR được kích hoạt với sự trượt ngang và trượt dọc, trong đó vcs biểu thị sự trượt ngang vận tốc và $v_{if} = r\omega_i(1 - s_i)$, $i = R, L$ biểu thị sự trượt về phía trước vận tốc tuyến tính của bánh xe bên phải và bên trái.

Để đảm bảo việc theo dõi hiệu suất của WMR và tối ưu hóa mức tiêu thụ năng lượng, vận tốc tuyến tính, vận tốc góc lái và mô men xoắn lái của đầu vào động cơ nên được tính đến và mô hình động lực học của WMR đã chọn được đưa ra như sau:

$$M(q)\ddot{q} + V(q,\dot{q})q + G(q) + \dot{\tau}_d = B(q)\tau - A^T(q)\lambda \qquad (2.31)$$

Trong đó $q = [xc, yc, \theta, \phi_R, \phi_L]^T \in R^{5\times1}$ là tư thế của xe trong hệ tọa độ tổng quát, $x_c$, $y_c$ và $\theta$ lần lượt biểu thị vị trí và góc hướng về phía trước, $x_c$, $y_c$ và $\theta$ biểu thị vị trí góc của bánh xe dẫn động, $M(q) \in R^{5\times5}$ biểu thị ma trận quán tính đối xứng xác định dương, $V(q, \dot{q}) \in R^{5\times5}$ là ma trận hướng tâm và Coriolis, $G(q) \in R^{5\times1}$ là vectơ hấp dẫn, $\tau_d \in R^{5\times1}$ biểu thị các nhiễu loạn bên ngoài bị giới hạn, $B(q) \in R^{5\times2}$ biểu thị ma trận biến đổi đầu vào, $A(q) \in R^{3\times5}$ ma trận liên quan đến các ràng buộc,

$\lambda \in R^{3\times1}$ là vectơ của lực ràng buộc và $\tau \in R^{2\times1}$ là vectơ mô men đầu vào được cung cấp bởi bộ truyền động.




*Hình 2.2. Mô hình xe tự hành 3 bánh với hệ phi tuyến [31]*

Trong trường hợp lý tưởng, có thể giả định rằng WMR không hoàn toàn bị ràng buộc mà không bị trượt, tức là $v_{if} = 0, i = R, L, c, s_j = 0, j = R, L.$

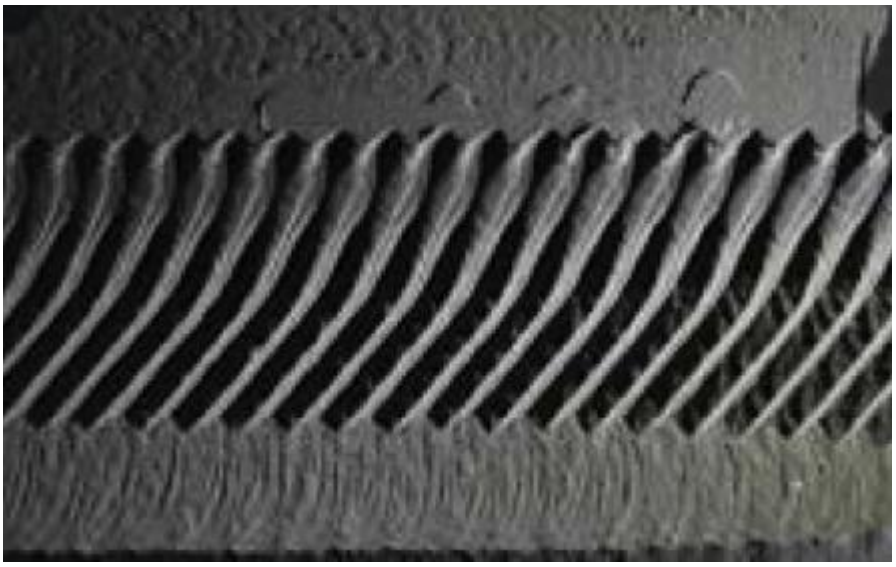
Để đảm bảo WMR chạy theo quỹ đạo mong muốn với tốc độ mong muốn, hãy thực hiện như sau phương trình ràng buộc phải được thỏa mãn

$$\begin{cases} -\dot{x_c}\sin\theta + \dot{y_c}\cos\theta - \dot{\theta}d = 0 \\ \dot{x_c}\cos\theta + \dot{y_c}\sin\theta + L\dot{\theta} = r\dot{\phi}_R \\ \dot{x_c}\cos\theta + \dot{y_c}\sin\theta - L\dot{\theta} = r\dot{\phi}_L \end{cases} \quad (2.32)$$

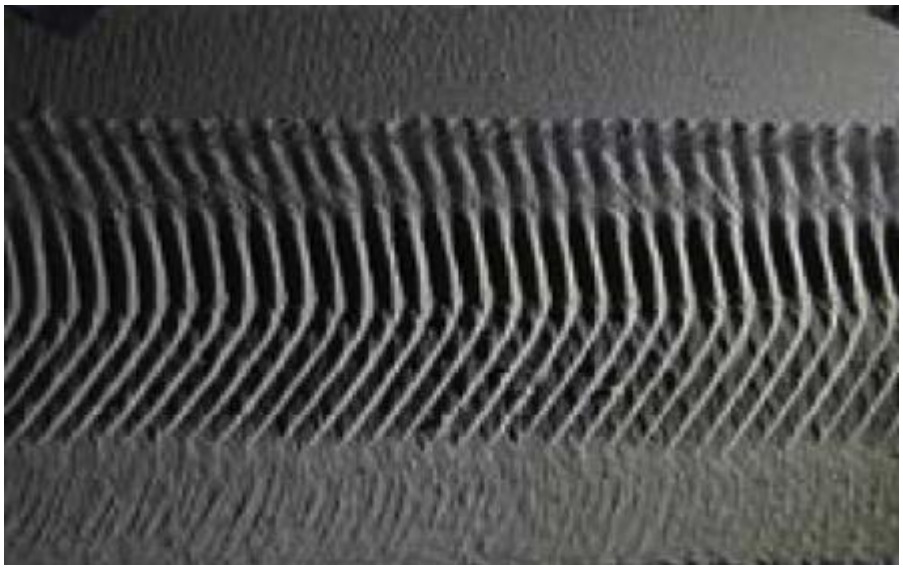
Trong đó $\dot{x_c}$ và $\dot{y_c}$ biểu diễn vận tốc của WMR theo quán tính hệ tọa độ, $\dot{\phi}_R$ and $\dot{\phi}_L$ biểu thị vận tốc góc của bánh xe dẫn động bên phải và bên trái, tương ứng, d là khoảng cách giữa tâm và tâm hình học của hai bánh xe dẫn động của hệ thống WMR, R biểu thị chiều rộng của WMR và r biểu thị bán kính của bánh xe.

*Hình 2.3. Vệt bánh xe khi không trơn trượt [31]*

*Hình 2.4. Vệt bánh xe khi trượt dọc và không trượt ngang [31]*



*Hình 2.5. Vệt bánh xe khi trượt ngang và không trượt dọc [31]*



*Hình 2.6. Vệt bánh xe khi trượt cả ngang và dọc [31]*

Tuy nhiên, những hạn chế này không phải lúc nào cũng được đáp ứng trong môi trường vận hành thực tế. Dựa trên sự so sánh giữa Hình 2.3 và 2.4, 2.5, 2.6 sự hiện diện của hiện tượng trượt và trượt trong WMR có tác động đáng kể đến vết xe của nó. Do đó, để thực hiện điều khiển theo dõi chính xác các WMR trong điều kiện làm việc thực tế, hiện tượng trượt và trượt cần được tính đến trong quá trình thiết kế thuật toán điều khiển tương ứng.

Hệ thống DT phi tuyến tính cho WMR với độ trượt và độ trượt không xác định có thể được tách thành hai phần. Một phần chứa độ trượt và độ trượt không xác định, và phần còn lại không có vận tốc ngang, chỉ duy trì vận tốc tuyến tính mong muốn theo hướng mong muốn. Trong nghiên cứu này, hai NN được sử dụng làm NN hành động để xử lý hai phần không xác định này, riêng lẻ.

Xem xét WMR như một hệ thống cơ học điển hình, mô men xoắn đầu vào của công suất động cơ, cấu trúc cơ học, hiệu suất truyền động và các khía cạnh khác của mô men xoắn đầu vào dẫn đến giới hạn ràng buộc đáng kể, được gọi là mô men xoắn bão hòa. Để thực hiện theo dõi góc thay đổi của WMR, mô

men xoắn bão hòa được mở rộng đến PDZ, có dạng tương tự với vùng chết và một thuật toán điều khiển theo dõi thích ứng dựa trên RL được thiết kế cho hệ thống DT phi tuyến tính afin được biến đổi.

- Xấp xỉ NN

Các hàm phi tuyến tính chưa biết có thể được xấp xỉ bằng NN ở mức độ mong muốn theo một số điều kiện. Bốn nhóm NN được sử dụng để ước tính gần đúng mục chưa biết hoặc không chắc chắn trong hệ thống. Sau đó, hàm f (ξ(k)) chưa biết, thỏa mãn tuyến tính trong tham số điều kiện có thể được xấp xỉ như

$$f(\xi(k)) = \omega^T \varphi\left(v^T s(k)\right) + \sigma(s(k)) \quad (2.33)$$

trong đó ω và v biểu thị trọng số mục tiêu của đầu ra và đầu vào của các lớp ẩn, tương ứng; $\varphi\left(v^T s(k)\right)$ là vectơ hàm kích hoạt, luôn được chọn là hàm cơ sở Gaussian, và được viết tắt là $\varphi(k)$; và $\sigma(s(k))$ là lỗi xấp xỉ hàm.

Để giảm thiểu hàm chi phí, ba NN khác được sử dụng khi hành động NNs để xấp xỉ mục mô hình chưa biết, các mục trượt và trượt không chắc chắn và PDZ. Sau đó, các lỗi xấp xỉ cho các NN hành động này có thể được xác định bằng cách sử dụng lỗi ước tính hàm $\varsigma_{a,1}(k)$, $\varsigma_{a,2}(k)$ $\tilde{\omega}_d(k)$, và các lỗi giữa hàm chi phí dài hạn mong muốn $\Xi_d(k)$ và phê bình thực tế tín hiệu $\hat{\Xi}(k)$.

Sau đó, các lỗi gần đúng của hành động NN được xác định là

$$z_{a,1}(k) = \sqrt{TE(k)m(k)}\,\overline{S}_{a,1}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\left(\hat{\Xi}(k) - \Xi_d(k)\right) \quad (2.34)$$

$$z_{a,2}(k) = \sqrt{TE(k)m(k)}\,\overline{S}_{a,2}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\left(\hat{\Xi}(k) - \Xi_d(k)\right) \quad (2.35)$$

$$z_d(k) = \sqrt{TE(k)m(k)}\,\widetilde{\omega_d}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\left(\hat{\Xi}(k) - \Xi_d(k)\right) \quad (2.36)$$

trong đó hàm chi phí dài hạn mong muốn được định nghĩa danh nghĩa là thấp nhất có thể, trong nghiên cứu này, tôi định nghĩa $\Xi_d(k) = 0$. Khi đó có thể viết lại như sau:

$$z_{a,1}(k) = \sqrt{TE(k)m(k)}\,\overline{S}_{a,1}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\hat{\Xi}(k) \quad (2.37)$$

$$z_{a,2}(k) = \sqrt{TE(k)m(k)}\,\overline{S}_{a,2}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\hat{\Xi}(k) \quad (2.38)$$

$$z_d(k) = \sqrt{TE(k)m(k)}\,\widetilde{\omega_d}(k) + \left(\sqrt{TE(k)m(k)}\right)^{-1}\hat{\Xi}(k) \quad (2.39)$$

Sau đó, ba hàm bậc hai của lỗi Bellman trình bày hàm mục tiêu được giảm thiểu bằng hành động NNs

$$E_{a,1}(k) = \frac{1}{2}z_{a,1}^T(k)z_{a,1}(k) \quad (2.40)$$

$$E_{a,2}(k) = \frac{1}{2}z_{a,2}^T(k)z_{a,2}(k) \quad (2.41)$$

$$E_d(k) = \frac{1}{2}z_d^T(k)z_d(k) \quad (2.42)$$

Xét theo phương pháp SGA, hành động này cân nhắc đến sự thích nghi có thể được định nghĩa là

$$\widehat{\omega_{a,1}}(k+1) = \widehat{\omega_{a,1}}(k) - \beta_{a,1}\varphi_{a,1}(k) \times \left[TE(k)m(k)S_{a,1}(k) + \hat{\Xi}(k)\right] \quad (2.43)$$

$$\widehat{\omega_{a,2}}(k+1) = \widehat{\omega_{a,2}}(k) - \beta_{a,2}\varphi_{a,2}(k) \times \left[TE(k)m(k)S_{a,2}(k) + \hat{\Xi}(k)\right] \quad (2.44)$$

$$\widehat{\omega_d}(k+1) = \widehat{\omega_d}(k) - \beta_d \times \left[TE(k)m(k)\widehat{\omega_d}(k) + \hat{\Xi}(k)\right] \quad (2.45)$$

Từ đó ta có thể thu được

$$\widehat{\omega_{a,1}}(k+1) = \widehat{\omega_{a,1}}(k) - \beta_{a,1}\varphi_{a,1}(k)\left[\hat{\Xi}(k) + z(k+1) - l_1 z(k) - TE(k)m(k) \times \left(S_{a,2}(k) + \widehat{\omega_d}(k)\right) - \varepsilon(k)\right] \quad (2.46)$$

Tuy nhiên, $\varsigma_{a,\,2}(k)$ $\tilde{\omega}\, d\,(k)$, và $\varepsilon(k)$ thường không có sẵn và a, chỉ khi khoảng thời gian lấy mẫu T được chọn đủ nhỏ và các nút của hành động đầu tiên NN được chọn một cách thích hợp.

$$\widehat{\omega_{a,1}}(k+1) = \widehat{\omega_{a,1}}(k) - \beta_{a,1}\varphi_{a,1}(k) \times \left[\hat{\Xi}(k) + z(k+1) - l_1 z(k)\right] \quad (2.47)$$

Tương tự luật thích ứng của hành động NN được sử dụng để xử lý các phần khác giữ vận tốc dọc theo cùng một hướng và không có vận tốc ngang có thể thu được

$$\widehat{\omega_{a,2}}(k+1) = \widehat{\omega_{a,2}}(k) - \beta_{a,2}\varphi_{a,2}(k) \times \left[\hat{\Xi}(k) + z(k+1) - l_1 z(k)\right] \quad (2.48)$$

luật thích nghi của hành động NN, rằng được sử dụng để xử lý PDZ chưa biết có thể thu được như

$$\widehat{\omega_d}(k+1) = \widehat{\omega_d}(k) - \beta_d\left[\hat{\Xi}(k) + z(k+1) - l_1 z(k)\right] \quad (2.49)$$

Dựa trên các giả định và nhận xét trước đó, sau đây định lý được thu được để kiểm chứng tính ổn định của hệ thống WMR.

Phương pháp đề xuất có thể đảm bảo tính giới hạn của NN phê bình ước tính trọng số và NN hành động ước tính trọng số. Ngoài ra, nếu bộ điều khiển là bị giới hạn, các lỗi theo dõi hội tụ về 0 và UUB của tất cả các tín hiệu trong hệ thống DT phi tuyến tính affine của WMR có thể là được đảm bảo.

### 2.2. Bài toán điều khiển bám quỹ đạo

Trong lĩnh vực điều khiển xe tự hành, bài toán bám quỹ đạo (trajectory tracking) được xem là một trong những mục tiêu then chốt, quyết định trực tiếp đến khả năng vận hành an toàn và chính xác của hệ thống. Nhiệm vụ đặt ra cho bộ điều khiển là bảo đảm cho xe di chuyển theo một quỹ đạo tham chiếu đã được xác định trước, đồng thời duy trì tính ổn định trong suốt quá trình hoạt

động, ngay cả khi xuất hiện nhiễu hoặc các yếu tố bất định từ môi trường thực tế.

Mục tiêu chính của bài toán bám quỹ đạo là làm cho xe không chỉ đi đúng theo đường dẫn mong muốn mà còn phải giữ sai số vị trí và sai số hướng ở mức nhỏ nhất có thể. Điều này đồng nghĩa với việc quỹ đạo thực tế của xe cần tiệm cận và duy trì gần với quỹ đạo tham chiếu, giúp nâng cao độ chính xác của quá trình điều khiển. Bên cạnh đó, bộ điều khiển cũng cần đảm bảo rằng tín hiệu điều khiển phát ra luôn trong giới hạn cho phép, tránh gây ra dao động mạnh hoặc vượt quá khả năng chịu tải của cơ cấu chấp hành.

Để đánh giá hiệu năng của bài toán bám quỹ đạo, một số tiêu chí thường được sử dụng gồm:

- Sai số bám quỹ đạo: Đây là thước đo quan trọng nhất, phản ánh mức độ chính xác khi xe di chuyển theo quỹ đạo tham chiếu. Sai số càng nhỏ chứng tỏ bộ điều khiển càng hiệu quả.
- Độ ổn định của hệ thống: Không chỉ cần chính xác, hệ thống còn phải duy trì trạng thái ổn định, tránh các dao động lớn có thể ảnh hưởng đến an toàn hoặc làm sai lệch quá trình điều khiển.
- Độ mượt của tín hiệu điều khiển: Tín hiệu điều khiển cần thay đổi một cách hợp lý, liên tục và mượt mà, nhằm giảm tải cơ học lên động cơ và kéo dài tuổi thọ phần cứng.
- Khả năng thích nghi: Bộ điều khiển cần duy trì hiệu quả ngay cả khi môi trường thay đổi hoặc khi xuất hiện nhiễu ngoài mong muốn, qua đó chứng minh tính bền vững của thuật toán.
- Hiệu suất năng lượng: Một bộ điều khiển tốt không chỉ chính xác mà còn phải tối ưu về năng lượng, giảm tiêu hao không cần thiết, đặc biệt quan trọng đối với các hệ thống robot di động hoạt động bằng pin.

## 2.3. Thiết kế giải pháp học tăng cường

### 2.3.1. Giải pháp học tăng cường cho mô hình tuyến tính

Giải thuật Q-learning áp dụng cho WMR:

Q-learning là một thuật toán học tăng cường theo cơ chế học giá trị hành động (action-value learning). Ý tưởng chính là xây dựng một hàm giá trị hành động Q(s, a), phản ánh "chất lượng" của việc lựa chọn hành động a tại trạng thái sss. Thông qua quá trình thử–sai và cập nhật dần, hàm Q(s, a) tiến gần đến giá trị tối ưu, từ đó suy ra chính sách điều khiển tốt nhất.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right] \quad (2.50)$$

Trong đó:

- $s_t$: trạng thái tại thời điểm ttt, phản ánh vị trí và hướng của xe.
- $a_t$: hành động được lựa chọn tại trạng thái $s_t$ (ví dụ: thay đổi vận tốc góc hay vận tốc tuyến tính).
- $r_{t+1}$: phần thưởng nhận được sau khi thực hiện hành động $a_t$.
- $\alpha$: tốc độ học (learning rate), quyết định mức độ điều chỉnh giá trị Q.
- $\gamma$: hệ số chiết khấu (discount factor), thể hiện tầm quan trọng của phần thưởng tương lai.

Trong nghiên cứu, mô hình xe tự hành được tuyến tính hóa xung quanh một quỹ đạo tham chiếu. Trạng thái hệ thống có thể bao gồm các biến: sai số vị trí theo phương ngang (lateral error), sai số góc hướng (heading error), và vận tốc. Các hành động điều khiển được mô hình hóa như việc thay đổi góc lái hoặc vận tốc tuyến tính.

Quá trình triển khai Q-learning cho mô hình tuyến tính:

- Xác định không gian trạng thái (State space):

  Trạng thái được rời rạc hóa để phù hợp với bảng Q. Ví dụ:

  - Sai số vị trí được chia thành các khoảng (âm, nhỏ, lớn).

- Sai số góc hướng được chia thành các mức (lệch trái, gần chuẩn, lệch phải).

- Xác định không gian hành động (Action space):
  Các hành động bao gồm một tập rời rạc các tín hiệu điều khiển, chẳng hạn:
  - Giữ nguyên vận tốc, tăng/giảm vận tốc.
  - Giữ nguyên góc lái, quay trái, quay phải.
- Thiết kế hàm thưởng (Reward function): Phần thưởng được thiết kế để khuyến khích xe giảm sai số bám quỹ đạo
- Chiến lược cân bằng thăm dò và khai thác (Exploration vs. Exploitation):
  Sử dụng chính sách ε-greedy, trong đó với xác suất ε\varepsilonε hệ thống sẽ chọn hành động ngẫu nhiên (thăm dò), còn lại sẽ chọn hành động tối ưu dựa trên giá trị Q hiện tại (khai thác).
- Huấn luyện và cập nhật bảng Q:
  Qua nhiều tập (episodes) mô phỏng, bảng Q dần dần được hoàn thiện và hội tụ về chính sách tối ưu.

### 2.3.2. Giải pháp học tăng cường cho mô hình phi tuyến bất định

Giải thuật ORADP áp dụng cho WMR:

Thuật toán OADP được phát triển dựa trên nền tảng của phương pháp lặp chính sách (PI). Tuy nhiên, điểm khác biệt của OADP nằm ở việc chỉ sử dụng một mạng nơ-ron (NN), nhờ đó quá trình cập nhật trọng số của mạng và các tham số điều khiển được tiến hành đồng thời trong cùng một vòng lặp. Điều này giúp giảm thiểu độ phức tạp trong tính toán và góp phần tăng tốc độ hội tụ của thuật toán.

Hàm mục tiêu về chất lượng được định nghĩa nhằm phản ánh sai lệch của hệ thống, có dạng:

$$J(e(0),u,d) = \int_0^\infty (e^T Q_1 e + u^T R u - \gamma^2 \|d\|^2) d\tau \quad (2.51)$$

trong đó $Q1 \in \mathbb{R}^{(2n-m)\times(2n-m)}$ là ma trận trọng số xác định dương, $\mathbb{R}^{(2n-m)\times(2n-m)}$ là ma trận trọng số đối xứng xác định dương, tham số $\gamma$ được lựa chọn sao cho $\gamma > \gamma *$ với $\gamma * > 0$ là giá trị nhỏ nhất của $\gamma$ đảm bảo rằng hệ thống kín vẫn duy trì được tính ổn định.

Hàm đánh giá cho WMR được xấp xỉ bởi NN dựa vào phương trình:

$$\hat{V}(e) = \widehat{W^T} \phi(e) \quad (2.52)$$

Dựa trên đó, ta có thể xác định được luật điều khiển tối ưu và luật nhiễu xấu nhất dưới dạng xấp xỉ như sau:

$$\hat{u}(e) = -\frac{1}{2} R^{-1} g^T(x) \phi_e^T(e) \widehat{W} \quad (2.53)$$

$$\hat{d}(e) = \frac{1}{2\gamma^2} k^T(x) \phi_e^T(e) \widehat{W} \quad (2.54)$$

trong đó $\phi_e(e) = \frac{\partial \phi(e)}{\partial e}$.

Luật cập nhật trọng số NN của ORADP:

$$\dot{\widehat{W}} = \begin{cases} \dot{\widehat{W}_1} & nếu\ \mathrm{e}_{\mathrm{t}+\mathrm{T}}^{\mathrm{T}} e_{t+T} \le \mathrm{e}_{\mathrm{t}}^{\mathrm{T}} \mathrm{e}_t \\ \dot{\widehat{W}_1} + W_{RB} & ngược\ lại \end{cases} \quad (2.55)$$

trong đó $et = e(t)$ và $e(t + T) = e(t + T)$, với $T$ là chu kỳ lấy mẫu, và

$$\dot{\widehat{W}_1} = \frac{-\alpha_1 \widehat{\sigma_e}}{\left(\widehat{\sigma_e^T} \widehat{\sigma_e} + 1\right)^2} \left( \int_t^{t+T} \left( e^T Q_1 e + \frac{1}{4} \widehat{W^T} \phi_e G \phi_e^T \widehat{W} - \frac{1}{4} \widehat{W^T} \phi_e K \phi_e^T \widehat{W} \right) d\tau + \Delta\phi^T(e) \widehat{W} \right) \quad (2.56)$$

$$W_{RB} = -\frac{\alpha_2}{2} \varphi_e \big(G(x) - K(x)\big) e \quad (2.57)$$

Trong đó

$$\widehat{\sigma_e} = \int_t^{t+T} \phi_e\big(f(x) + g(x)\hat{u} + k(x)\hat{d}\big)d\tau = \int_t^{t+T} d\left(\phi\big(e(\tau)\big)\right) = \phi(e_{t+T}) - \phi(e_t) = \Delta\phi\big(e(t)\big) \quad (2.58)$$

Các bước thực hiện:

- Bước 1: Chọn $Q(x)$, $R$; Chọn véc tơ hàm tác động $\phi$, nhiễu ống (Probing noise) $\xi$. Khởi tạo trọng số $W^0 = 0$ cho NN hàm đánh giá, tính $V^0 = u^0 = d^l = 0$; Gán các hệ số thích nghi $\alpha_1$, $\alpha_2$; Gán bước lặp dừng giải thuật $l\text{stop}$; Gán $\delta$ là số dương đủ nhỏ để tắt nhiễu PE; Gán $l = 0$;
- Bước 2: Cộng nhiễu $\xi$ vào tín hiệu điều khiển: $u^l \leftarrow u^l + \xi$, $d^l \leftarrow d^l + \xi$ để kích thích hệ thống theo điều kiện PE [100]; Cập nhật đồng bộ trọng số NN $W^{l+1}$ và tham số luật điều khiển và luật nhiễu:

$$\widehat{u^{(l+1)}} = -\frac{1}{2}R^{-1}g(x)^T\phi_x^T\widehat{W^{(l+1)}} \quad (2.59)$$

Tính hàm đánh giá:

$$\widehat{V^{(l+1)}} = \widehat{W^{(l+1)T}}\phi(x) \quad (2.60)$$

- Bước 3: Nếu $\|V^l - V^{l+1}\| < \delta$ gán $\xi = 0$. Nếu $l \leq lstop$ gán $l \longleftarrow l + 1$, quay lại Bước 2, ngược lại gán $V^* = V^{l+1}$ và $u^* = u^{l+1}$ sau đó dừng giải thuật.

## 2.4. Kết luận chương 2

Trong Chương 2, luận án đã trình bày chi tiết mô hình xe phi tuyến bất định, bao gồm các phương trình động học và động lực học, cùng với những giả thiết và điều kiện ràng buộc cần thiết để xây dựng hệ thống nghiên cứu. Bên cạnh đó, các yếu tố ảnh hưởng đến quá trình điều khiển xe tự hành cũng đã được phân tích, làm rõ vai trò của trạng thái, tín hiệu điều khiển, và các tham số bất định trong môi trường vận hành.

Trên cơ sở mô hình toán học, chương này cũng đã giới thiệu phương pháp nghiên cứu và giải thuật ORADP (Online Robust Adaptive Dynamic Programming), trong đó mạng nơ-ron được sử dụng để xấp xỉ hàm giá trị và tín hiệu điều khiển tối ưu. Quá trình thiết kế, điều kiện kích thích bền vững (PE) cũng như cách thức cập nhật trọng số NN đã được đề cập, nhằm đảm bảo giải thuật hội tụ và duy trì khả năng điều khiển ổn định.

# CHƯƠNG 3 - THỰC NGHIỆM VÀ KẾT QUẢ

## 3.1. Môi trường mô phỏng

Cấu hình máy tính cần thiết để thực hiện mô phỏng xe tự hành:

❖ Phần cứng:

- CPU: Tối thiểu: Intel Core i5 thế hệ 8 trở lên hoặc AMD Ryzen 5. Khuyến nghị: Intel Core i7 thế hệ 10 hoặc AMD Ryzen 7.
- RAM: Tối thiểu: 8 GB. Khuyến nghị: 16–32 GB (đặc biệt khi mô phỏng nhiều kịch bản hoặc huấn luyện dài).
- GPU (tăng tốc học sâu và mô phỏng đồ họa): Tối thiểu: NVIDIA GTX 1050Ti (4 GB VRAM). Khuyến nghị: NVIDIA RTX 3060/3070 (8 GB VRAM trở lên).
- Ổ cứng: SSD ≥ 256 GB (khuyến nghị 512 GB để lưu nhiều dữ liệu và kết quả).
- Hệ điều hành: Windows 10/11 64-bit.

❖ Phần mềm:

- Ngôn ngữ & Framework: Python 3.8/3.9.
    - Thư viện học sâu: PyTorch (khuyến nghị, dễ dùng cho RL) hoặc TensorFlow.
    - Thư viện hỗ trợ: NumPy, SciPy, Matplotlib, Gym (hoặc Gymnasium), Stable-Baselines3.
- Môi trường mô phỏng robot
    - MATLAB/Simulink (tích hợp tốt cho mô hình toán và điều khiển).
    - Hoặc Python + Matplotlib/Pygame để xây dựng môi trường 2D đơn giản.
    - Nếu cần 3D/robot thật: Webots hoặc CoppeliaSim (tương thích Windows).
- Công cụ hỗ trợ

- Anaconda (hoặc Miniconda): quản lý môi trường Python.
- Git: quản lý và lưu trữ mã nguồn.
- IDE: Visual Studio Code hoặc PyCharm để lập trình và gỡ lỗi.

## 3.2. Thực nghiệm với mô hình tuyến tính

### 3.2.1. Xây dựng giải thuật cho mô hình tuyến tính

Trong hệ thống nghiên cứu này, tôi thiết lập môi trường với các chướng ngại động và chướng ngại vật tĩnh. Cụ thể, chướng ngại vật động được mô phỏng là bốn vật thể chuyển động theo quỹ đạo tròn cố định nằm phía trước xe. Các vật thể này có khả năng chuyển động ngẫu nhiên, trong đó việc lựa chọn chuyển động tròn giúp hạn chế tối đa khả năng các vật thể di chuyển va chạm lẫn nhau. Đối với chướng ngại vật tĩnh, chúng được coi là các đường biên giới hạn khu vực chuyển động. Đồng thời, chúng đóng vai trò quyết định trạng thái va chạm của xe, tức là xác định xem xe có tiếp xúc với các vật thể tĩnh hay không. Khi xe di chuyển và xảy ra tiếp xúc với bất kỳ chướng ngại vật nào trong số này, tình huống đó được xem là va chạm.

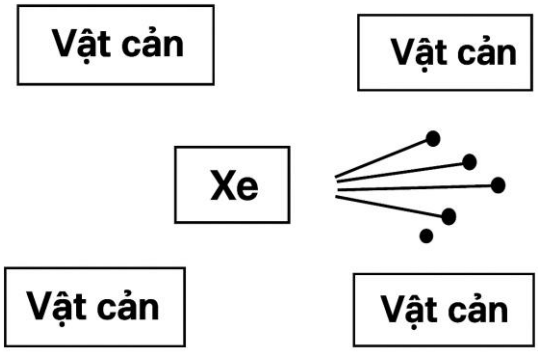
- Xây dựng bộ trạng thái và hàm phản hồi tương ứng

Việc xác định bộ trạng thái cũng như các tín hiệu phản hồi của hệ thống đóng vai trò then chốt, ảnh hưởng trực tiếp đến hiệu quả vận hành và khả năng học của hệ thống. Trong đề tài này, tôi lựa chọn bộ trạng thái bao gồm thông tin về vị trí và vận tốc của xe, kết hợp với thông tin về vị trí và vận tốc của các chướng ngại vật. Việc có hoặc không có thông tin về không gian hoặc bổ sung thông tin về chướng ngại vật sẽ tác động trực tiếp tới hiệu quả của quá trình huấn luyện mô hình. Do đó, lựa chọn bộ trạng thái được thực hiện dựa trên tập hợp tín hiệu từ các cảm biến. Các cảm biến này trả về giá trị nhị phân, với giá trị 1 tương ứng với việc phát hiện chướng ngại vật ở khoảng cách gần, còn giá trị 0 biểu thị không có vật cản trong vùng quét.

Trong Hình 3.1, mô hình huấn luyện được xây dựng nhằm mô phỏng quá trình điều khiển xe di chuyển trong môi trường có sự xuất hiện của chướng ngại vật động và tĩnh. Xe được trang bị các cảm biến bố trí ở phía trước mũi xe, giúp phát hiện vật cản nằm trong vùng quét. Các cảm biến này được đặt theo hình dạng vòng cung hoặc nửa hình tròn trước xe, mỗi cảm biến sẽ phụ trách quét một khu vực không gian nhất định.

Cơ chế hoạt động của cảm biến như sau: nếu bất kỳ điểm nào trên các chướng ngại vật nằm trong phạm vi phát hiện của cảm biến (thường được xác định bởi khoảng cách tối thiểu từ cảm biến đến các vật thể), tín hiệu trả về sẽ là 1, ngược lại nếu không có vật cản nào trong vùng quét thì cảm biến sẽ trả về giá trị 0. Điều này giúp hệ thống có thể nhanh chóng phát hiện và phân loại tình huống có nguy cơ va chạm.

Mục đích của việc sử dụng sơ đồ này là tạo điều kiện cho hệ thống học được cách nhận biết chướng ngại vật trong không gian hoạt động của xe, từ đó lựa chọn hành động phù hợp nhằm tránh va chạm. Thông tin từ các cảm biến chính là đầu vào quan trọng giúp mô hình học tăng cường cập nhật chính sách điều khiển một cách tối ưu nhất trong quá trình huấn luyện.




*Hình 3.1. Mô hình vật lý hệ thống xe tự hành*

Để xây dựng bộ trạng thái trong không gian nhiều chiều, tôi lựa chọn mô hình trạng thái dựa trên các tập hợp giá trị rời rạc. Cụ thể, trạng thái sss được định nghĩa dưới dạng: $s(u, v, x, y, z)$ trong đó u, v, x, y, z $\in\{0, 1\}$ đại diện cho năm cảm biến phát hiện chướng ngại vật được lắp đặt trên xe. Giá trị của mỗi cảm biến được mã hóa nhị phân: nếu cảm biến phát hiện có chướng ngại vật trong vùng quét thì nhận giá trị 1, ngược lại nếu không phát hiện chướng ngại vật thì nhận giá trị 0. Trạng thái s biểu diễn tình huống của hệ thống trước khi thực hiện hành động, còn $s'$ đại diện cho trạng thái hệ thống sau khi hành động được thực hiện.

Trong không gian trạng thái đã thiết lập, mỗi trạng thái sẽ tương ứng với một giá trị phần thưởng cụ thể. Để giải quyết bài toán này, tôi lựa chọn hàm phần thưởng dạng âm R, phản ánh mức phạt trong trường hợp xe va chạm hoặc tiếp xúc với chướng ngại vật. Cụ thể, nếu sau khi hành động được thực hiện, đầu xe nằm trong khu vực an toàn không có vật cản (tức là tất cả các cảm biến đều không phát hiện chướng ngại vật), trạng thái của cảm biến được biểu diễn bởi: $s(u, v, x, y, z) = \theta = [0\ 0\ 0\ 0\ 0]^T$ thì phần thưởng nhận giá trị 0, phản ánh trạng thái an toàn. Ngược lại, nếu sau khi thực hiện hành động, cảm biến phát hiện xe vẫn nằm trong vùng có chướng ngại vật (có ít nhất một cảm biến báo hiệu 1), phần thưởng sẽ nhận giá trị -1 (giá trị phạt) nếu như sau khi thực hiện hành động đầu xe không thoát khỏi vùng chứa chướng ngại vật, hay $s(u, v, x, y, z) \neq \theta$. Khi có va chạm, chương trình sẽ dừng episode đó lại và bắt đầu một episode mới.

- Xây dựng tập hành động và bảng Q

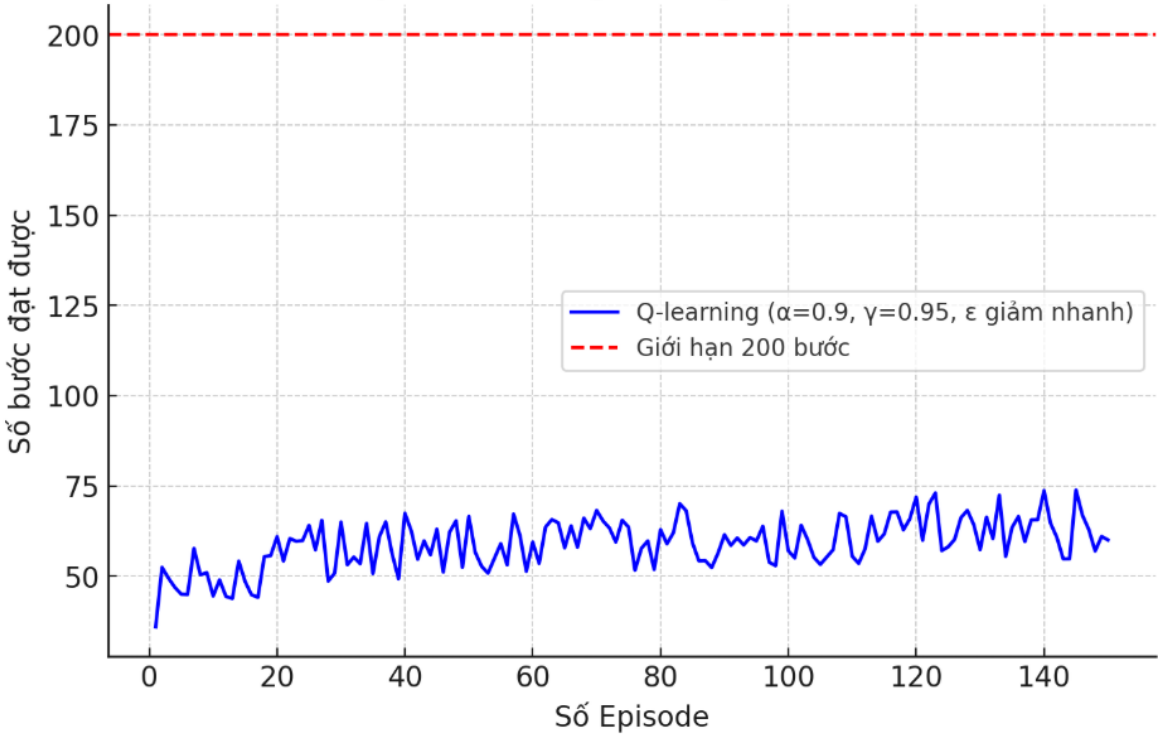
Để thiết kế tập hợp hành động cho xe tự hành, tôi lựa chọn ba hành động cơ bản là: di chuyển thẳng, rẽ trái và rẽ phải. Các hành động này được mã hóa

thành các chỉ số như sau: hành động rẽ trái được gán giá trị 1, đi thẳng gán giá trị 2, và rẽ phải có giá trị 3.

Với ba hành động khả thi tại mỗi trạng thái, bảng Q sẽ có kích thước được xác định bằng công thức: số trạng thái nhân với số hành động, cụ thể là (s_num) x (a_num) = $2^5$ x 3. Bảng Q có chức năng lưu trữ giá trị kỳ vọng của từng hành động trong mỗi trạng thái, từ đó giúp tác tử lựa chọn hành động tối ưu nhằm tối đa hóa phần thưởng kỳ vọng trong dài hạn. Quy mô của bảng Q tỷ lệ thuận với số lượng trạng thái và hành động, điều này cũng phản ánh mức độ phức tạp trong quá trình huấn luyện. Hệ thống dựa vào kết quả lưu trong bảng Q để đưa ra quyết định thực hiện hành động trong tập hành động nhằm đạt được mức thưởng tối đa trong dài hạn. Độ lớn của bảng Q được xác định dựa vào tích số của số lượng hành động với số lượng trạng thái.

### 3.2.2. Kết quả mô phỏng và phân tích

Trong môi trường mô phỏng vật lý, tác tử là xe tự hành được huấn luyện kỹ năng né tránh chướng ngại vật thông qua thuật toán Q-learning. Các tham số học tập bao gồm hệ số học $\alpha = 0.9$ và hệ số chiết khấu $\gamma = 0.95$. Ngoài ra, chiến lược chọn hành động được áp dụng là chính sách $\varepsilon$-greedy, giúp cân bằng giữa khai thác và khám phá hành động mới. Số bước lặp (steps) trong mỗi tập huấn luyện được giới hạn ở mức 200 bước, và quá trình huấn luyện được thực hiện trong 150 episode. Chương trình huấn luyện được chạy lặp lại 20 lần để lấy giá trị trung bình kết quả, và kết quả được trình bày trong Hình 3.2.




*Hình 3.2. Kết quả huấn luyện xe tự hành*

Hình 3.2 mô tả kết quả huấn luyện xe tự hành, cụ thể là số step trung bình đạt được qua từng episode của 20 lần thực hiện ngẫu nhiên. Quan sát Hình 3.2 cho thấy trong 2 episode đầu tiên, xe chỉ đi được dưới 45 step. Từ episode thứ 3 đến episode thứ 21, khả năng tránh vật thể của xe liên tục được cải thiện và đạt được số step cao dần. Tuy đạt được mức step tối đa bắt đầu từ episode thứ 21, mức tối đa này không phải lúc nào cũng đạt ở các episode tiếp theo. Mức này càng lúc càng có xác suất đạt cao hơn khi episode tăng.

Hình 3.2 minh họa kết quả huấn luyện của xe tự hành, cụ thể là số bước đi trung bình đạt được qua mỗi episode, được tính từ 20 lần thử ngẫu nhiên. Quan sát Hình 3.2 có thể thấy, trong hai episode đầu tiên, xe chỉ di chuyển được dưới 45 bước. Từ episode thứ 3 đến episode thứ 21, khả năng tránh vật cản của xe được cải thiện rõ rệt, thể hiện qua số bước tăng dần đều đến gần 200 bước. Tuy nhiên, dù đạt được mức tối đa là 200 bước từ episode 21 trở đi, nhưng không phải episode nào cũng chạm mức này. Sự dao động cho thấy việc đạt

được số bước cao phụ thuộc vào nhiều yếu tố, nhưng nhìn chung, xác suất đạt được số bước cao sẽ tăng theo số lần huấn luyện.

Nguyên nhân dẫn đến kết quả nêu trên có thể lý giải như sau: Trong giai đoạn đầu huấn luyện, xe chưa được trang bị đầy đủ kiến thức để né tránh các chướng ngại vật, bao gồm cả vật thể tĩnh và chuyển động. Điều này dễ khiến xe va chạm và phải kết thúc sớm. Cần lưu ý rằng chương trình sẽ kết thúc một episode nếu xe va chạm hoặc tốc độ của xe giảm xuống dưới 50 m/s. Vì vậy, ở những episode ban đầu, số bước di chuyển còn thấp là điều dễ hiểu. Quan sát Hình 3.2, ta thấy rằng số bước tăng dần theo thời gian huấn luyện, cho thấy khả năng thích ứng và né tránh chướng ngại của xe ngày càng cải thiện. Nói cách khác, khi số episode tăng lên, xe có xu hướng hoàn thành quãng đường dài hơn trước khi bị sự cố, phản ánh quá trình học tập hiệu quả của hệ thống.

Điều này có nghĩa rằng mặc dù xe di chuyển liên tục trong môi trường có các hướng ngại vật tĩnh và động, xe đã tự biết điều chỉnh kiến thức học hỏi được và ra các quyết định càng lúc càng chính xác, tránh được các vật thể. Khi giá trị step đạt mức ổn định gần 200, điều đó cho thấy xe có thể vận hành hiệu quả trong các môi trường phức tạp, đồng thời duy trì được số bước di chuyển tối đa trong nhiều lần huấn luyện kế tiếp. Những kết quả đạt được chứng minh rằng thuật toán được triển khai đã hoạt động tốt. Cụ thể, với thuật toán học tăng cường Q-learning, xe tự hành đã học được kỹ năng tránh né vật cản hiệu quả, cả trong trường hợp tĩnh và chuyển động.

Một điểm cần lưu ý thêm là với môi trường có độ phức tạp cao, như chứa nhiều vật thể và tốc độ di chuyển của xe lớn, thì khả năng tránh va chạm đôi khi vẫn bị hạn chế. Đây là lý do vì sao, ở một số episode dù số bước đạt cao, xe vẫn có thể gặp va chạm trước khi hoàn thành tối đa số bước. Điều này cho thấy rằng, việc kiểm chứng hiệu quả của thuật toán điều khiển học tăng cường

trong mô hình xe tự hành phi tuyến là cần thiết. Phải đánh giá không chỉ qua các chỉ số trung bình mà còn phải xét đến độ tin cậy và tính ổn định khi triển khai thuật toán trong điều kiện thực tế.

## 3.3. Thực nghiệm với mô hình phi tuyến bất định

### 3.3.1. Xây dựng giải thuật cho mô hình phi tuyến bất định

❖ Mô phỏng:

Trong quá trình áp dụng giải thuật ORADP, robot trong môi trường mô phỏng thực tế thường gặp phải một số vấn đề: mạng nơ-ron khởi tạo trọng số ban đầu một cách ngẫu nhiên hoặc không theo quy luật cụ thể, khiến quá trình điều khiển ban đầu trở nên khá đơn giản. Tuy nhiên, nếu thuật toán điều khiển không được tối ưu hóa hợp lý, hệ thống sẽ cần trải qua một giai đoạn học dài mới có thể đạt được tín hiệu điều khiển ổn định theo điều kiện PE. Đặc biệt, việc có nhiều điều kiện PE kích hoạt trong thời gian dài dễ làm hư hại các cơ cấu chấp hành cũng như phần cứng của robot.

Để tránh các vấn đề kể trên, trong giai đoạn đầu, thuật toán ORADP được triển khai cho robot mô phỏng trong một môi trường đã được thiết lập với các thông số cơ bản của robot thực nghiệm để thu được ma trận trọng số NN xấp xỉ sau khi hội tụ, sau đó sử dụng các thông số này cho bộ điều khiển trên robot thực nghiệm để quá trình học điều khiển tiếp tục với các thông tin thuận lợi cho trước.

❖ Quỹ đạo tham chiếu:

Quỹ đạo tham chiếu được phát sinh bởi vận tốc $\vartheta rd$ của robot ảo được chọn theo giả thiết khả vi liên tục

$$\vartheta_{rd} = \begin{bmatrix} v_{rd} \\ \omega_{rd} \end{bmatrix} = \begin{bmatrix} \sqrt{(A_1\cos(\omega_1 t))^2 + (A_2\cos(\omega_2 t))^2} \\ \frac{A_1\omega_1\sin(\omega_1 t)A_2\cos(\omega_2 t) - A_2\omega_2\sin(\omega_2 t)A_1\cos(\omega_1 t)}{(A_1\cos(\omega_1 t))^2 + (A_2\cos(\omega_2 t))^2} \end{bmatrix} \quad (3.1)$$

trong đó $\omega 1 = 0.04(rad/s)$, $\omega 2 = 0.02(rad/s)$, $A1 = 0.022(m/s)$, $A2 = 0.02(m/s)$. Có $\vartheta rd$ dễ dàng giải phương trình $q' = S(qd)\vartheta rd$ với điều kiện đầu $qd(0) = [xd(0), yd(0), \theta d(0)]^T = [0.1, -0.6, 0.6]^T$ ta có vị trí tham chiếu $qd = [xd, yd, \theta d]^T$.

❖ Thiết lập tham số học:

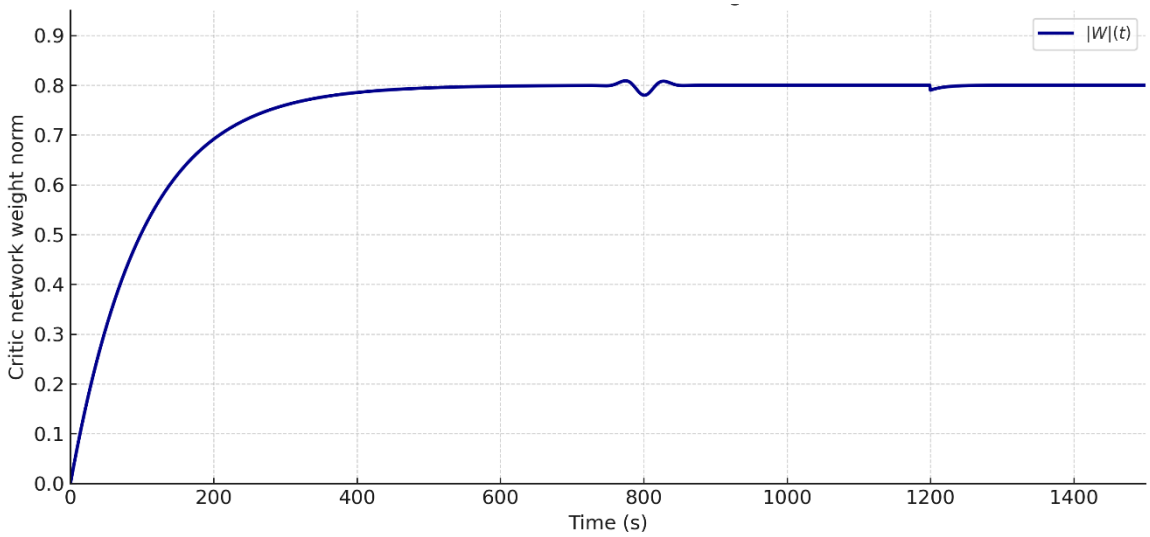
Véc tơ trọng số NN được định nghĩa $W = [W1, W2, \dots, W15]^T$ với các giá trị khởi tạo bằng không. Hằng số học được chọn là $\alpha 1 = 25$ và $\alpha 2 = 0.01$. Véc tơ hàm tác động NN, $\phi(e) \in \mathbb{R}15$ được chọn như sau:

$$\phi(e) = \left[e_x^2,\ e_x e_y,\ e_x e_\theta,\ e_x e_v,\ e_x e_\omega,\ e_y^2,\ e_y e_\theta,\ e_y e_v,\ e_y e_\omega,\ e_\theta^2,\ e_\theta e_v,\ e_\theta e_\omega,\ e_v^2,\ e_v e_\omega,\ e_\omega^2\right]^T \quad (3.2)$$

Chọn $R = I \in \mathbb{R}^{4\times4}$, $Q = I \in \mathbb{R}^{5\times5}$ và $\gamma = 1$. Điều kiện PE được áp dụng bằng cách thêm vào nhiễu $\xi = e^{-0.005}trand(t)$ vào ngõ vào điều khiển trong đó $rand(t)$ là hàm phát sinh tín hiệu ngẫu nhiên trong khoảng [-1,1]. Véc tơ trạng thái của robot: $q = [x, y, \theta]^T, \vartheta = [v, \omega]^T$, các thông số robot mô phỏng chọn trùng với thông số robot thực $r1 = 0.05(m)$, $b1 = 0.35(m)$ và $l = 0$, $m = 10(kg)$, $I = 5(kG.m2)$, trong đó $m$, $I$ lần lượt là tổng khối lượng và tổng mô men quán tính của khung robot, động cơ và bánh xe.

Nhiễu $\tau m$ giả lập trong khoảng $\pm 3(N.m)$. Vị trí và vận tốc khởi tạo của WMR lần lượt là $q(0) = [0.1, -0.6, 0]^T(m), \vartheta(0) = [0,0]^T$. Tham số T được chọn $0.01(s)$.

### 3.3.2. Kết quả mô phỏng và phân tích




*Hình 3.3. Sự hội tụ của trọng số NN trong quá trình học điều khiển*

Với việc áp dụng thuật toán ORADP cùng các tham số học đã được lựa chọn, quá trình học đã cho phép trọng số của mạng nơ-ron hội tụ. Cụ thể, các trọng số hội tụ được xác định tại thời điểm 300 giây. Tuy nhiên, do đặc điểm của quá trình học, một số điều kiện kích thích PE có thể xuất hiện bất kỳ lúc nào sau đó, ví dụ như ở mốc thời gian 800 giây. Dưới đây là giá trị của vector trọng số mạng nơ-ron tại thời điểm hội tụ 300 giây:

$$W = \begin{bmatrix} -56.5, 149.9, -340.1, -134.5, -240.3, -277.4, -213.5, -105.1, -70.2, 26.4, -55, \\ 126, -201.1, 257.3, 321.5 \end{bmatrix}^T \times 10^{-3} \quad (3.3)$$

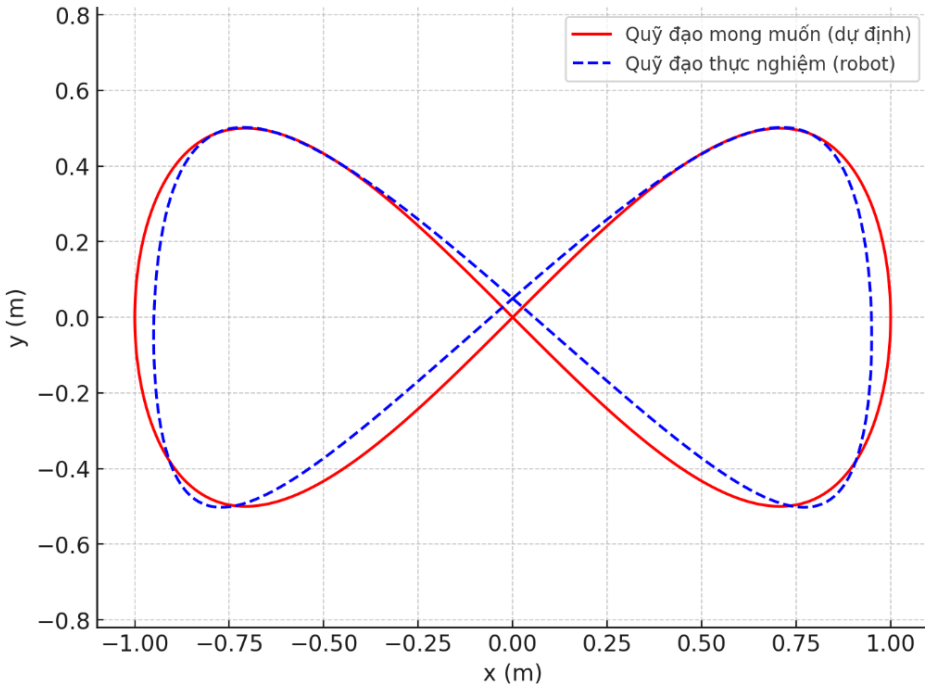
Sai số về vị trí và chất lượng bám tối ưu được ghi nhận tại thời điểm hội tụ sau 800 giây như đã nêu. Hệ thống vẫn đảm bảo tính ổn định lâu dài trong suốt quá trình hoạt động của hệ điều khiển, ngay cả khi robot thay đổi lộ trình định sẵn. Ví dụ, tại thời điểm 1200 giây, khi khối lượng của robot giảm từ 40 kg xuống còn 10 kg và mô men quán tính giảm từ 15 (kg.m²) xuống còn 5 (kg.m²), thì sai số bám chỉ dao động với biên độ nhỏ. Kết quả cho thấy các sai

số này đều ổn định và lặp lại, không có sự thay đổi lớn. Do đó, các thay đổi ngẫu nhiên tại một số thời điểm nhất định trong quá trình mô phỏng gần như không ảnh hưởng đáng kể đến độ chính xác của hệ thống điều khiển.

$$\hat{u} = -\frac{1}{2}R^{-1}\begin{bmatrix} \cos\theta & 0 & 0 & 0 \\ \sin\theta & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{mr_1} & \frac{b_1}{Ir_1} \\ 0 & 0 & \frac{1}{mr_1} & -\frac{b_1}{Ir_1} \end{bmatrix} \begin{bmatrix} 2e_x & 0 & 0 & 0 & 0 \\ e_y & e_x & 0 & 0 & 0 \\ e_\theta & 0 & e_x & 0 & 0 \\ e_\vartheta & 0 & 0 & e_x & 0 \\ e_\omega & 0 & 0 & 0 & e_x \\ 0 & 2e_y & 0 & 0 & 0 \\ 0 & e_\theta & e_y & 0 & 0 \\ 0 & e_\vartheta & 0 & e_y & 0 \\ 0 & e_\omega & 0 & 0 & e_y \\ 0 & 0 & 2e_\theta & 0 & 0 \\ 0 & 0 & e_\vartheta & e_\theta & 0 \\ 0 & 0 & e_\omega & 0 & e_\theta \\ 0 & 0 & 0 & 2e_\vartheta & 0 \\ 0 & 0 & 0 & e_\omega & e_\vartheta \\ 0 & 0 & 0 & 0 & 2e_\omega \end{bmatrix} \begin{bmatrix} -0{,}0565 \\ 0.1499 \\ -0.3401 \\ -0.1345 \\ -0.2403 \\ -0.2774 \\ -0.2135 \\ -0.1051 \\ -0.0702 \\ 0.0264 \\ -0.0550 \\ 0.1260 \\ -0.2011 \\ 0.2573 \\ -0.3215 \end{bmatrix} \quad (3.4)$$

$$\hat{d} = \frac{1}{2\gamma^2}\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{mr_1} & \frac{b_1}{Ir_1} \\ 0 & 0 & \frac{1}{mr_1} & -\frac{b_1}{Ir_1} \end{bmatrix} \begin{bmatrix} 2e_x & 0 & 0 & 0 & 0 \\ e_y & e_x & 0 & 0 & 0 \\ e_\theta & 0 & e_x & 0 & 0 \\ e_\vartheta & 0 & 0 & e_x & 0 \\ e_\omega & 0 & 0 & 0 & e_x \\ 0 & 2e_y & 0 & 0 & 0 \\ 0 & e_\theta & e_y & 0 & 0 \\ 0 & e_\vartheta & 0 & e_y & 0 \\ 0 & e_\omega & 0 & 0 & e_y \\ 0 & 0 & 2e_\theta & 0 & 0 \\ 0 & 0 & e_\vartheta & e_\theta & 0 \\ 0 & 0 & e_\omega & 0 & e_\theta \\ 0 & 0 & 0 & 2e_\vartheta & 0 \\ 0 & 0 & 0 & e_\omega & e_\vartheta \\ 0 & 0 & 0 & 0 & 2e_\omega \end{bmatrix} \begin{bmatrix} -0.0565 \\ 0.1499 \\ -0.3401 \\ -0.1345 \\ -0.2403 \\ -0.2774 \\ -0.2135 \\ -0.1051 \\ -0.0702 \\ 0.0264 \\ -0.0550 \\ 0.1260 \\ -0.2011 \\ 0.2573 \\ -0.3215 \end{bmatrix} \quad (3.5)$$

Chất lượng bám cho sai số lớn khi mới học nhưng sau đó mặc dù bị ảnh hưởng của nhiễu, thay đổi khối lượng và mô men quán tính nhưng sai số bám vẫn hội tụ tiệm cận về giá trị tối ưu.




*Hình 3.4. Quỹ đạo vị trí của robot thực nghiệm*

Hình 3.4 thể hiện quỹ đạo chuyển động của robot trong không gian với đường tham chiếu (hình số 8) và đường thực nghiệm thu được từ mô phỏng. Ở giai đoạn đầu sai số bám còn tương đối lớn, đặc biệt ở các đoạn cong hẹp. Điều này là do bộ điều khiển và mạng nơ-ron đang trong quá trình học và điều chỉnh trọng số để thích ứng với hệ động học của robot.

Ở giai đoạn ổn định sau khi quá trình học hội tụ, quỹ đạo thực nghiệm gần như trùng khớp với quỹ đạo tham chiếu. Sai số bám tiệm cận về giá trị nhỏ, cho thấy bộ điều khiển có khả năng bù trừ nhiễu và thay đổi tham số động học (khối lượng, mô men quán tính).

Sự trùng khớp giữa quỹ đạo tham chiếu và quỹ đạo thực nghiệm khẳng định rằng phương pháp điều khiển đề xuất có thể áp dụng cho các hệ thống robot tự hành thực tế, nơi các điều kiện vận hành luôn thay đổi.

### 3.4. Đánh giá ưu điểm của mô hình đề xuất

Kết quả thực nghiệm cho thấy mô hình điều khiển xe phi tuyến bất định sử dụng giải thuật ORADP có nhiều ưu điểm nổi bật. Trước hết, mô hình thể hiện khả năng thích nghi mạnh mẽ trước các yếu tố bất định và nhiễu trong môi trường hoạt động. Nhờ vào cơ chế học tăng cường và khả năng xấp xỉ hàm của mạng nơ-ron, bộ điều khiển có thể tự động điều chỉnh để duy trì ổn định mà không cần dựa vào mô hình toán học chính xác của hệ thống. Đây là một ưu điểm quan trọng, bởi trong thực tế việc xây dựng mô hình toán học đầy đủ cho robot di động phi tuyến thường rất phức tạp và không khả thi.

Thứ hai, mô hình giúp rút ngắn đáng kể thời gian huấn luyện trên robot thực. Thông qua giai đoạn mô phỏng, trọng số ban đầu của mạng nơ-ron được thiết lập ở trạng thái gần hội tụ, giúp bộ điều khiển khởi động nhanh hơn, tránh được giai đoạn học kéo dài vốn có thể gây ra dao động lớn hoặc tác động tiêu cực lên phần cứng. Cách tiếp cận này không chỉ tiết kiệm thời gian mà còn nâng cao độ an toàn và độ tin cậy trong quá trình triển khai thực nghiệm.

Thứ ba, sau khi đạt trạng thái hội tụ, hệ thống cho tín hiệu điều khiển có độ ổn định cao và độ dao động nhỏ. Điều này chứng minh khả năng duy trì chất lượng điều khiển bền vững, đồng thời đáp ứng tốt yêu cầu về tính ổn định trong vận hành lâu dài. Đặc biệt, cơ chế cập nhật liên tục của thuật toán cho phép hệ thống duy trì hiệu quả điều khiển ngay cả khi điều kiện môi trường thay đổi.
Ngoài ra, mô hình còn thể hiện tính linh hoạt và khả năng mở rộng ứng dụng. Do không phụ thuộc vào dạng cụ thể của hệ phi tuyến, phương pháp có thể được áp dụng cho nhiều loại phương tiện tự hành khác nhau như robot bánh, xe tự hành hoặc thậm chí các hệ thống cơ điện tử phức tạp hơn. Điều này mở

ra tiềm năng ứng dụng rộng rãi trong các lĩnh vực như giao thông thông minh, robot dịch vụ, hay tự động hóa công nghiệp.

### 3.5. Kết luận chương

Trong Chương 3, luận án đã triển khai giải thuật học tăng cường trên mô hình xe thực nghiệm để kiểm chứng trong điều kiện vận hành thực tế. Các kết quả thu được cho thấy:

- Xe tự hành đạt khả năng bám quỹ đạo ổn định, phản ứng tốt trước các nhiễu và bất định, đồng thời duy trì được tính bền vững của hệ thống điều khiển.
- Kết quả thực nghiệm khẳng định tính đúng đắn của mô hình mô phỏng, đồng thời chứng minh giải thuật đề xuất có thể ứng dụng hiệu quả trong điều khiển xe phi tuyến bất định.
- Một số hạn chế còn tồn tại như ảnh hưởng của nhiễu môi trường, sai số cảm biến và độ trễ truyền thông, cho thấy cần có các nghiên cứu mở rộng để tối ưu hơn trong thực tế.

Như vậy, chương này đã xác nhận tính khả thi của giải thuật trong môi trường thực nghiệm, đồng thời cung cấp cơ sở cho phần kết luận chung và định hướng phát triển của đề án.

# KẾT LUẬN

Trong đề án này, tôi đã nghiên cứu về giải thuật học tăng cường ứng dụng điều khiển bám quỹ đạo cho xe tự hành, một lĩnh vực đầy tiềm năng và đang phát triển trong lĩnh vực trí tuệ nhân tạo. Mục tiêu của nghiên cứu là khám phá và áp dụng các giải thuật học tăng cường để tạo ra chính sách điều khiển thông minh cho xe tự hành, giúp nâng cao hiệu suất lái xe và khả năng ứng phó với các tình huống phức tạp trên đường.

Nghiên cứu này đã đưa ra một đóng góp quan trọng trong việc áp dụng giải thuật học tăng cường cho xe tự hành. Kết quả nghiên cứu đã chứng minh rằng giải thuật học tăng cường có tiềm năng để cải thiện hiệu suất lái xe và tăng cường khả năng ứng phó với các tình huống phức tạp. Tuy nhiên, cần tiếp tục nghiên cứu và phát triển các phương pháp và giải thuật mới để đảm bảo an toàn và hiệu quả tối đa trong xe tự hành.

Hy vọng rằng nghiên cứu này sẽ cung cấp cơ sở để tiếp tục phát triển và tối ưu hóa giải thuật học tăng cường cho xe tự hành, đóng góp vào sự phát triển của công nghệ xe tự hành và đảm bảo an toàn và tiến bộ trong ngành giao thông. Các hướng nghiên cứu tiếp theo có thể tập trung vào việc kết hợp giải thuật học tăng cường với các phương pháp học sâu khác như mạng nơ-ron hồi quy, mạng nơ-ron tích chập và mạng nơ-ron sinh. Ngoài ra, cần tiếp tục nghiên cứu và phát triển các phương pháp thu thập dữ liệu hiệu quả để cải thiện quá trình huấn luyện và tăng tính tổng quát của giải thuật. Tuy vẫn còn nhiều thách thức phải đối mặt, nhưng sự kết hợp giữa trí tuệ nhân tạo và xe tự hành hứa hẹn sẽ thay đổi cách chúng ta di chuyển và cải thiện cuộc sống hàng ngày. Từ đó sẽ góp phần thúc đẩy sự phát triển của lĩnh vực này và cung cấp cơ sở để xây dựng các hệ thống xe tự hành an toàn, thông minh và hiệu quả hơn trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] H. Modi, N. V. Shastri, M. H. Shastri. 2015. Obstacle Avoidance System in Autonomous Ground Vehicle using Ground Plane Image Processing and Omnidirectional Drive, International Conference on Computing Communication Control and Automation, 594-598.

[2] S. Shoval, I. Zeitoun and E. Lenz. 1997. Implementation of a Kalman filter in positioning for autonomous vehicles, and its sensitivity to the process parameters, Int J Adv Manuf Technol, 13,738-746.

[3] Richard S. Sutton and Andrew G. Barto. (2016), Reinforcement Learning: An Introduction, The MIT Press Cambridge, Massachusetts London, England.

[4] K. Iagnemma, S. Kang, H. Shibly, S. Dubowsky, Online terrain parameter esti- mation for wheeled mobile robots with application to planetary rovers, IEEE Trans. Robot. 20 (5) (2004) 921–927.

[5] Zhang H, Luo Y,Liu D (2009) Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. IEEE Trans Neural Netw 20(9):1490–1503.

[6] C.M. Gifford, E.L. Akers, R.S. Stansbury, A. Agah, Mobile robots for polar remote sensing, The Path to Autonomous Robots, Springer, US, 2009, pp. 1–22.

[7] Yang Q, Jagannathan S (2007) Online reinforcement learning neural network controller design for nanomanipulation. In: Pro ceedings of IEEE symposium on approximate dynamic pro gramming and reinforcement learning. Honolulu, HI, pp 225–232.

[8] R. Balakrishna, A. Ghosal, Modeling of slip for wheeled mobile robots, IEEE Trans. Robot. Autom. 11 (1) (1995) 126–132.

[9] D. Wang, C.B. Low, Modeling and analysis of skidding and slipping in wheeled mobile robots: control design perspective, IEEE Trans. Robot. 24 (3) (2008) 676–687.

[10] Xu D, Zhao DB, Yi JQ, Tan XM (2009) Trajectory tracking control of omnidirectional wheeled mobile manipulators: robust neural network based sliding mode approach [J]. IEEE Trans Syst Man Cybern Part B 39(3):788–799.

[11] G. Campion, G. Bastin, B. Dandrea-Novel, Structural properties and classifica- tion of kinematic and dynamic models of wheeled mobile robots, IEEE Trans. Robot. Autom. 12 (1) (1996) 47–62.

[12] Zhang PC, Xu X, Liu C, Yuan Q (2009) Reinforcement learning control of a real mobile robot using approximate policy iteration [C]. ISNN 2009, Part III, Lecture Notes in Computer Science, LNCS 5553, pp 278–288

[13] Park BS, Yoo SJ et al (2010) A simple adaptive control approach for trajectory tracking of electrically driven nonholonomic mobile robots[J]. IEEE Trans Control Syst Technol 18(5):1199 1206.

[14] D.H. Kim, J.H. Oh, Tracking control of a two-wheeled mobile robot using in- put–output linearization, Control Eng. Pract. 7 (3) (1999) 369–373.

[15] J. Chen, W.E. Dixon, M. Dawson, M. McIntyre, Homography-based visual servo tracking control of a wheeled mobile robot, IEEE Trans. Robot. 22 (2) (2006) 406–415.

[16] C.B. Low, D. Wang, GPS-based tracking control for a car-like wheeled

mobile robot with skidding and slipping, IEEE/ASME Trans. Mechatron. 13 (4) (2008) 4 80–4 84.

[17] Mohareri O, Dhaouadi R et al (2012) Indirect adaptive tracking control of a nonholonomic mobile robot via neural networks[J]. Neurocomputing 88:54–66.

[18] J.X. Xu, Z.Q. Guo, T.H. Lee, Design and implementation of integral sliding-mode control on an underactuated two-wheeled mobile robot, IEEE Trans. Ind. Elec- tron. 61 (7) (2014) 3671–3681.

[19] ZhaoDB, DengXY, Yi JQ (2009) Motionand internal force control for omni-directional wheeledmobile robots [J]. IEEE ASME Trans Mechatron 14(3):382–387

[20] H.S. Kang, Y.T. Kim, C.H. Hyun, M. Park, Generalized extended state observer approach to robust tracking control for wheeled mobile robot with skidding and slipping, Int. J. Adv. Robot. Syst. 10 (3) (2013) 155.

[21] E.J. Hwang, H.S. Kang, C.H. Hyun, M. Park, Robust backstepping control based on a Lyapunov redesign for skid-steered wheeled mobile robots, Int. J. Adv. Robot. Syst. 10 (1) (2013) 26.

[22] D.P. Li, D.J. Li, Adaptive neural tracking control for nonlinear time-delay sys- tems with full state constraints, IEEE Trans. Syst. Man Cybern. Syst. 47 (7) (2017) 1590–1601.

[23] S.S. Ge, J. Zhang, T.H. Lee, Adaptive neural network control for a class of MIMO nonlinear systems with disturbances in discrete-time, IEEE Trans. Syst. Man Cybern. Part B Cybern. 34 (4) (2004) 1630–1645.

[24] Mahadevan S, Maggioni M (2006) Value function approximation with diffusion wavelets and Laplacian eigenfunctions[C]. In: Proceedings of the neural information processing systems (NIPS). MIT Press.

[25] S. Jagannathan, P. He, Neural-network-based state feedback control of a

nonlin- ear discrete-time system in nonstrict feedback form, IEEE Trans. Neural Netw. 19 (12) (2008) 2073–2087.

[26] Z.J. Li, C. Yang, Neural-adaptive output feedback control of a class of trans- portation vehicles based on wheeled inverted pendulum models, IEEE Trans. Control Syst. Technol. 20 (6) (2012) 1583–1591.

[27] Y.J. Liu, S.C. Tong, Optimal control-based adaptive NN design for a class of nonlinear discrete-time block-triangular systems, IEEE Trans. Cybern. 46 (11) (2016) 2670–2680.

[28] Liu D, Javaherian H, Kovalenko O, Huang T (2008) Adaptive critic learning techniques for engine torque and air-fuel ratio control [J]. IEEE Trans Syst Man Cybern Part B Cybern 38(4):988–993.

[29] P. Shih, B.C. Kaul, S. Jagannathan, J.A. Drallmeier, Reinforcement-learning-based output-feedback control of nonstrict nonlinear discrete-time systems with ap- plication to engine emission control, IEEE Trans. Syst. Man Cybern. Part B Cy- bern. 39 (5) (2009) 1162–1179.

[30] L. Zuo, X. Xu, C.M. Liu, Z.H. Huang, A hierarchical reinforcement learning ap- proach for optimal path tracking of wheeled mobile robots, Neural Comput. Appl. 23 (7–8) (2013) 1873–1883.

[31] Shu Li, Liang Ding, Haibo Gao, Chao Chen, Zhen Liu, Zongquan Deng. “Adaptive neural network tracking control-based reinforcement learning for wheeled mobile robots with skidding and slipping”. Neurocomputing, Vol. 283, pp. 20–30, 2018.