****

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

**BÁO CÁO THỰC HÀNH**

Nhóm CKTXây dựng hệ thống phát hiện và tránhvật cản cho hệ tự hành

Ứng dụng xử lý ảnh số và video số 20\_23

*Giảng viên – TS. Lý Quốc Ngọc*

Thành phố Hồ Chí Minh – 2023

**MỤC LỤC**

[**THÔNG TIN CÁC THÀNH VIÊN TRONG NHÓM** 3](#_Toc138109537)

[**I.** **Tổng quan** 4](#_Toc138109538)

[**II.** **Đánh giá** 4](#_Toc138109539)

[**1.** **Nhận dạng hệ thống** 4](#_Toc138109540)

[**2.** **Theo dõi hiệu suất với bộ điều khiển** 4](#_Toc138109541)

[**3.** **Thi đua với bộ điều khiển bản ngã (MPC-CBF)** 5](#_Toc138109542)

[**4.** **Thi đua với bộ điều khiển bản ngã (iLQR)** 5](#_Toc138109543)

[**5.** **Thi đua với bộ điều khiển bản ngã (LMPC)** 6](#_Toc138109544)

[**6.** **Cuộc thi đua xe với người lập kế hoạch và điều khiển bản ngã** 7](#_Toc138109545)

[**7.** **Thời gian thực (đang được phát triển)** 9](#_Toc138109546)

[References 11](#_Toc138109547)

# **THÔNG TIN CÁC THÀNH VIÊN TRONG NHÓM**

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ Tên | Email | Ghi chú |
| 20120201 | Phạm Gia Thông | 20120201@student.hcmus.edu.vn |  |
| 20120259 | Nguyễn Văn Chung | 20120259@student.hcmus.edu.vn | **Nhóm trưởng** |

1. **Tổng quan**

Trong dự án này, pytest được sử dụng để kiểm tra mã một cách tự động sau khi đẩy mã mới vào kho lưu trữ. Hiện tại, ba tệp trong thư mục thử nghiệm được sử dụng để kiểm tra bộ điều khiển theo dõi pid hoặc mpc, bộ điều khiển mpc-cbf và công cụ lập kế hoạch trò chơi đua xe, tương ứng. Để kiểm tra các tính năng khác, hãy thêm tệp vào thư mục kiểm tra và cập nhật tệp tests.yml

Để cài đặt bạn nên tạo môi trường conda mới:

conda env create -f environment.ymlconda kích hoạt đua xe

Chạy lệnh sau trong thiết bị đầu cuối để cài đặt gói mô phỏng đua xe.

pip cài đặt -e .

1. **Đánh giá**
2. **Nhận dạng hệ thống**

Chạy

Python car\_racing/tests/system\_identification\_test.py

Điều này cho phép xác định động lực tuyến tính của xe đua bằng hồi quy.

1. **Theo dõi hiệu suất với bộ điều khiển**

Chạy

Python car\_racing/tests/control\_test.py --Ctrl-Policy MPC-LTI --track-layout l\_shape --simulation --plotting --animation

Điều này cho phép kiểm tra thuật toán để theo dõi. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| ctrl\_policy | xâu | PID, MPC-LTI, LQR | Chính sách kiểm soát |
| track\_layout | xâu | l\_shape, m\_shape, kính bảo hộ, hình elip | Bố cục theo dõi |
| Mô phỏng | hành động | store\_true | Tạo dữ liệu mô phỏng nếu đúng, nếu không hãy đọc dữ liệu mô phỏng từ các tệp hiện có |
| Vẽ | hành động | store\_true | Lưu cốt truyện nếu đúng |
| hoạt hình | hành động | store\_true | Lưu hoạt ảnh nếu đúng |

Thực hiện một thuật toán phản ứng để làm cho ô tô lái tự động quanh Levine Hall bản đồ. Có thể tự do triển khai bất kỳ thuật toán phản ứng nào bạn muốn, sẽ chỉ phải chỉnh sửa khoảng trống phản ứng follow.py. Ngoài ra còn có một bản đồ thử nghiệm (levine Blocked.pgm) để bạn đánh giá.

1. **Thi đua với bộ điều khiển bản ngã (MPC-CBF)**

Chạy

Python car\_racing/tests/mpccbf\_test.py --track-layout l\_shape --simulation --plotting --animation

Điều này cho phép kiểm tra thuật toán cho bộ điều khiển MPC-CBF. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| track\_layout | xâu | l\_shape, m\_shape, kính bảo hộ, hình elip | Bố cục theo dõi |
| Mô phỏng | hành động | store\_true | Tạo dữ liệu mô phỏng nếu đúng, nếu không hãy đọc dữ liệu mô phỏng từ các tệp hiện có |
| Vẽ | hành động | store\_true | Lưu cốt truyện nếu đúng |
| hoạt hình | hành động | store\_true | Lưu hoạt ảnh nếu đúng |

1. **Thi đua với bộ điều khiển bản ngã (iLQR)**

Chạy

Python car\_racing/tests/ilqr\_test.py --track-layout l\_shape --simulation --plotting --animation

Điều này cho phép kiểm tra thuật toán cho bộ điều khiển iLQR. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| track\_layout | xâu | l\_shape, m\_shape, kính bảo hộ, hình elip | Bố cục theo dõi |
| Mô phỏng | hành động | store\_true | Tạo dữ liệu mô phỏng nếu đúng, nếu không hãy đọc dữ liệu mô phỏng từ các tệp hiện có |
| Vẽ | hành động | store\_true | Lưu cốt truyện nếu đúng |
| hoạt hình | hành động | store\_true | Lưu hoạt ảnh nếu đúng |

1. **Thi đua với bộ điều khiển bản ngã (LMPC)**

Để lưu các trạng thái lịch sử và đầu vào được sử dụng cho MPC dựa trên học tập, trước tiên hãy chạy lệnh sau cho mỗi bố cục bản nhạc:

Python car\_racing/tests/lmpc\_test.py \--track-layout l\_shape --lap-number 7 --simulation --save-trajectory

Sau đó, bạn có thể chạy lệnh sau:

Python car\_racing/tests/lmpc\_test.py \--track-layout l\_shape --lap-number 10 --simulation --direct-lmpc --animation --plotting

Điều này cho phép kiểm tra thuật toán cho MPC dựa trên học tập. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| track\_layout | xâu | l\_shape, m\_shape, kính bảo hộ, hình elip | Bố cục theo dõi |
| lap\_number | Int | bất kỳ số nào lớn hơn 2 | Số vòng sẽ được mô phỏng |
| direct\_lmpc | hành động | store\_true | nếu đúng, trình mô phỏng sẽ khởi động bộ điều khiển LMPC trực tiếp bằng cách sử dụng quỹ đạo lưu trữ |
| zero\_noise | hành động | store\_true | Không có tiếng ồn trong cập nhật động nếu đúng |
| save\_trajectory | hành động | store\_true | nếu true và khi bộ điều khiển là LMPC, trình mô phỏng sẽ lưu trữ các trạng thái lịch sử và đầu vào |
| Mô phỏng | hành động | store\_true | Tạo dữ liệu mô phỏng nếu đúng, nếu không hãy đọc dữ liệu mô phỏng từ các tệp hiện có |
| Vẽ | hành động | store\_true | Lưu cốt truyện nếu đúng |
| hoạt hình | hành động | store\_true | Lưu hoạt ảnh nếu đúng |

1. **Cuộc thi đua xe với người lập kế hoạch và điều khiển bản ngã**

Để lưu các trạng thái lịch sử và đầu vào được sử dụng cho MPC dựa trên học tập, trước tiên hãy chạy lệnh sau cho mỗi bố cục bản nhạc:

Python car\_racing/tests/overtake\_planner\_test.py \--track-layout l\_shape --lap-number 7 --simulation --number-other-agents 0 --save-trajectory

Sau đó, bạn có thể chạy lệnh sau:

Python car\_racing/tests/overtake\_planner\_test.py \--track-layout l\_shape --lap-number 10 --simulation --direct-lmpc --animation --plotting --number-other-agents 3

Điều này cho phép kiểm tra thuật toán để cạnh tranh đua xe. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| track\_layout | xâu | l\_shape, m\_shape, kính bảo hộ, hình elip | Bố cục theo dõi |
| lap\_number | Int | bất kỳ số nào lớn hơn 2 | Số vòng sẽ được mô phỏng |
| direct\_lmpc | hành động | store\_true | nếu đúng, trình mô phỏng sẽ khởi động bộ điều khiển LMPC trực tiếp bằng cách sử dụng quỹ đạo lưu trữ |
| sim\_replay | hành động | store\_true | Nếu đúng, bằng cách thay đổi đường dẫn File, trình mô phỏng sẽ mô phỏng với các tham số khác nhau nhưng từ cùng điều kiện ban đầu |
| zero\_noise | hành động | store\_true | Không có tiếng ồn trong cập nhật động nếu đúng |
| diff\_alpha | hành động | store\_true | Nếu đúng, các giá trị alpha khác nhau sẽ được sử dụng cho cùng điều kiện ban đầu |
| random\_other\_agents | hành động | store\_true | Các tác nhân khác sẽ được tạo ngẫu nhiên nếu đúng |
| number\_other\_agents | Int | bất kỳ số nào lớn hơn 0, khi nó được đặt thành 0, thuật toán là LMPC | Số lượng tác nhân sẽ được tạo |
| save\_trajectory | hành động | store\_true | nếu true và khi bộ điều khiển là LMPC, trình mô phỏng sẽ lưu trữ các trạng thái lịch sử và đầu vào |
| multi\_tests | hành động | store\_true | Nếu TURE, 100 nhóm thử nghiệm được tạo ngẫu nhiên sẽ được mô phỏng |
| Mô phỏng | hành động | store\_true | Tạo dữ liệu mô phỏng nếu đúng, nếu không hãy đọc dữ liệu mô phỏng từ các tệp hiện có |
| Vẽ | hành động | store\_true | Lưu cốt truyện nếu đúng |
| hoạt hình | hành động | store\_true | Lưu hoạt ảnh nếu đúng |

Hiện tại, người lập kế hoạch đường đi và người lập kế hoạch trajecotry có sẵn cho thao tác vượt. Thay đổi self.path\_planner varibale trong base.py thành True cho phép bộ điều khiển mô phỏng với trình lập kế hoạch đường dẫn.

1. **Thời gian thực (đang được phát triển)**

Để khởi động trình mô phỏng, hãy chạy lệnh sau trong thiết bị đầu cuối:

Roslaunch car\_racing car\_racing\_sim.Launch track\_layout:=Goggle

Điều này cho phép bạn chạy nút giả lập và trực quan hóa. Thay đổi track\_layout, bạn có thể nhận được các bản nhạc khác nhau. Đường trung tâm của đường đua được vẽ bằng đường gạch ngang màu đỏ; Quỹ đạo tối ưu của đường đua được vẽ theo đường màu xanh lá cây. Để thêm xe mới có bộ điều khiển trong trình mô phỏng, hãy chạy các lệnh sau trong thiết bị đầu cuối mới:

Rosrun car\_racing vehicle.py --VEH-Name Vehicle1 --Màu xanh --VX 0 --VY 0 --WZ 0 --EPSI 0 --S 0 --EY 0

Rosrun car\_racing controller.py --Ctrl-Chính sách MPC-LTI --VEH-Tên xe1

Chúng cho phép khởi động các nút cho xe và bộ điều khiển tương ứng. Các đối số argparse được liệt kê như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên** | **Kiểu** | **Sự lựa chọn** | **Mô tả** |
| veh\_name | xâu | Tên tự xác định | Tên xe |
| màu | xâu | Tên của màu sắc | Màu sắc của xe trong hoạt hình |
| VS, VY, WZ, EPSI, S , EY | nổi | Trạng thái ban đầu | Trạng thái ban đầu của xe trong tọa độ Frenet |
| ctrl\_policy | xâu | PID, MPC-LTI, MPC-CBF, LMPC | Loại bộ điều khiển của xe |

# References

[1] Tối ưu hóa song song cho hành vi đua vượt với nhiều xe tự hành [IEEE][ arXiv]

[2] Thiết kế mô hình điều khiển dự đoán với các chức năng rào cản điều khiển để tránh chướng ngại vật trong các vấn đề đua xe [IEEE][ arXiv] [NorCal Control Workshop Talk]