

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

GIANG VĂN TÍN

LUẬN VĂN TỐT NGHIỆP
ỨNG DỤNG THUẬT TOÁN TIME TO COLLISION
VÀO XE ĐIỆN TỰ LÁI

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

TP. HỒ CHÍ MINH, 2022

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN ĐIỀU KHIỂN TỰ ĐỘNG

GIANG VĂN TÍN - 1811279

LUẬN VĂN TỐT NGHIỆP
ỨNG DỤNG THUẬT TOÁN TIME TO COLLISION
VÀO XE ĐIỆN TỰ LÁI
(APPLICATION OF TIME TO COLLISION ALGORITHM TO
AUTONOMOUS ELECTRIC VEHICLES)

KỸ SƯ NGÀNH KỸ THUẬT ĐIỀU KHIỂN & TỰ ĐỘNG HÓA

GIẢNG VIÊN HƯỚNG DẪN
TS. NGUYỄN HOÀNG GIÁP

TP. HỒ CHÍ MINH, 2022

TRƯỜNG ĐẠI HỌC BÁCH KHOA TP. HỒ CHÍ MINH
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN: ĐIỀU KHIỂN TỰ ĐỘNG

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập - Tự do - Hạnh phúc

TP. HCM, ngày.... tháng..... năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ HƯỚNG DẪN**

Tên luận văn:

**ỨNG DỤNG THUẬT TOÁN TIME TO COLLISION
VÀO XE ĐIỆN TỰ LÁI**

Sinh viên thực hiện:

Giang Văn Tín

Cán bộ hướng dẫn:

1811279 TS. Nguyễn Hoàng Giáp

Đánh giá Luận văn

1. Về cuốn báo cáo:

Số trang _____

Số chương _____

Số bảng số liệu _____

Số hình vẽ _____

Số tài liệu tham khảo _____

Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

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

2. Về nội dung luận văn:

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

3. Về tính ứng dụng:

.....

.....

.....

.....

.....

.....

4. Về thái độ làm việc của sinh viên:

.....

.....

.....

.....

.....

.....

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Giang Văn Tín :...../10

Cán bộ hướng dẫn

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày.....tháng.....năm.....

**NHẬN XÉT LUẬN VĂN TỐT NGHIỆP
CỦA CÁN BỘ PHẢN BIỆN**

Tên luận văn:

**ỨNG DỤNG THUẬT TOÁN TIME TO COLLISION
VÀO XE ĐIỆN TỰ LÁI**

Sinh viên thực hiện:

Giang Văn Tín

Cán bộ phản biện:

1811279

Đánh giá Luận văn

5. Về cuốn báo cáo:

Số trang _____

Số chương _____

Số bảng số liệu _____

Số hình vẽ _____

Số tài liệu tham khảo _____

Sản phẩm _____

Một số nhận xét về hình thức cuốn báo cáo:

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

6. Về nội dung luận văn:

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

7. Về tính ứng dụng:

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

8. Về thái độ làm việc của sinh viên:

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

Đánh giá chung: Luận văn đạt/không đạt yêu cầu của một luận văn tốt nghiệp kỹ sư, xếp loại Giỏi/ Khá/ Trung bình

Điểm từng sinh viên:

Giang Văn Tín :/10

Người nhận xét

(Ký tên và ghi rõ họ tên)

TP. HCM, ngày.... tháng..... năm.....

ĐỀ CƯƠNG CHI TIẾT

TÊN LUẬN VĂN: ỨNG DỤNG THUẬT TOÁN TIME TO COLLISION

VÀO XE ĐIỆN TỰ LÁI

Cán bộ hướng dẫn: TS. Nguyễn Hoàng Giáp

Thời gian thực hiện: Từ ngày 03/01/2022 đến ngày 20/05/2022

Sinh viên thực hiện:

Giang Văn Tín - 1811279

Nội dung đề tài:

- Mục tiêu: thiết kế hệ thống điều khiển và ứng dụng thực tế lý thuyết Time to Collision vào phương tiện xe điện tự lái. Với khả năng tự động theo dõi đối tượng phía trước hướng di chuyển của xe và tự động điều khiển để hạn chế thấp nhất nguy cơ xảy ra va chạm.
- Đối tượng: phương tiện xe điện, cảm biến Radar, hệ thống phanh cứng, vi xử lý điều khiển.
- Phương pháp thực hiện: phân tích nghiên cứu lý thuyết, thực nghiệm đánh giá.
- **Kết quả mong muốn:** phương tiện xe tự lái có thể hoạt động thực tế, tự động điều khiển tránh, làm giảm khả năng xảy ra các tình huống tai nạn va chạm.

Kế hoạch thực hiện:

Thời gian	Công việc	Người thực hiện
Tháng 01/2022	Tìm hiểu tổng quan về đề tài, yêu cầu và nhiệm vụ đặt ra.	Giang Văn Tín

Tháng 01/2022	Tìm hiểu lý thuyết TTC (tham khảo các bài báo khoa học). Nghiên cứu và xây dựng TTC áp dụng riêng trên xe tự lái.	Giang Văn Tín
Tháng 02/2022	Thiết kế và xây dựng tủ điện điều khiển (nguồn, vi xử lý).	Giang Văn Tín
Tháng 02/2022	Xử lý dữ liệu Radar cần thiết cho tính toán TTC, theo dõi đối tượng.	Giang Văn Tín
Tháng 03/2022	Mô phỏng tính toán TTC và điều khiển tránh vật thể trên Gazebo Ubuntu.	Giang Văn Tín
Tháng 03/2022	Xây dựng GUI hiển thị theo dõi đối tượng trên Qt.	Giang Văn Tín
Tháng 03/2022	Kết nối ROS với máy tính để gửi kết quả TTC lên GUI điều khiển (C++).	Giang Văn Tín
Tháng 04/2022	Thực nghiệm lý thuyết trên phương tiện thực tế.	Giang Văn Tín
Tháng 05/2022	Đánh giá, hiệu chỉnh đạt kết quả mong muốn và viết báo cáo.	Giang Văn Tín
Xác nhận của Cán bộ hướng dẫn (Ký tên và ghi rõ họ tên)		TP. HCM, ngày tháng năm.....
Sinh viên (Ký tên và ghi rõ họ tên)		

DANH SÁCH HỘI ĐỒNG BẢO VỆ LUẬN VĂN

Hội đồng chấm luận văn tốt nghiệp, thành lập theo Quyết định số
ngày của Hiệu trưởng Trường Đại học Bách khoa TP.HCM.

1. – Chủ tịch.
2. – Thư ký.
3. – Ủy viên.
4. – Ủy viên.
5. – Ủy viên.

LỜI CẢM ƠN

Lời đầu tiên, em muốn gửi lời cảm ơn chân thành đến với quý Thầy Cô trường đại học Bách Khoa TP.HCM, đặc biệt là quý Thầy Cô khoa Điện - Điện tử đã tận tình giảng dạy và truyền đạt những kiến thức quý báu trong suốt thời gian em học tập và rèn luyện tại trường.

Em gửi Thầy Nguyễn Hoàng Giáp lời cảm ơn chân thành và sâu sắc đã hỗ trợ, hướng dẫn cho em tận tình, đưa ra những lời khuyên để em có thể hoàn thành được đề tài luận văn của mình. Đó là sự may mắn khi em được làm việc cùng với Thầy trong suốt thời gian qua.

Cảm ơn tất cả bạn bè đã giúp đỡ trong quá trình học tập. Cảm ơn các thành viên trong nhóm thực hiện đề tài AEV, cùng hỗ trợ nhau để hoàn thành nghiên cứu, cùng nhau phát triển để đề tài đạt được thành công tốt đẹp nhất.

Lời cuối cùng, em kính chúc quý thầy cô trường Đại học Bách Khoa dồi dào sức khỏe, đạt được nhiều thành công trong cuộc sống và sự nghiệp, cũng như luôn giữ ngọn lửa đam mê trong nghiên cứu và giảng dạy.

TP. Hồ Chí Minh, ngày tháng năm

Giang Văn Tín

DANH MỤC TỪ VIẾT TẮT

Từ khóa	Ý nghĩa
AEV	Autonomous Electric Vehicle
ADAS	Advanced Driver Assistance Systems
ADC	Analog to digital converter
CAS	Collision Avoidance Systems
CMOS	Complementary metal-oxide semiconductor
DPU	Data Path Units
DSP	Digital signal processor
EKF	Extended Kalman filter
FFT	Fast Fourier transform
MCU	Microcontrollers
MPC	Model Predictive Control
PID	Proportional–Integral–Derivative Controller
PWM	Pulse Width Modulation
ROS	Robot Operating System
RMS	Root Mean Square
SDK	Software development kit
SNR	Signal-to-Noise Ratio
TLV	Type Length Values
USB	Universal Serial Bus

MỤC LỤC

Chương 1. GIỚI THIỆU ĐỀ TÀI	2
1.1. Giới thiệu vấn đề nghiên cứu	2
1.1.1. Giới thiệu tổng quan xe điện tự hành	2
1.1.2. Sự cần thiết của đề tài nghiên cứu	2
1.1.3. Tình hình trong và ngoài nước	4
1.2. Mục tiêu của đề tài	9
1.2.1. Ý nghĩa khoa học	9
1.2.2. Phạm vi ứng dụng	10
Chương 2. CƠ SỞ LÝ THUYẾT.....	12
2.1. Mô hình động lực học xe điện 4 bánh	12
2.1.1. Hệ trục tọa độ gắn với xe – mô phỏng mô hình xe điện	12
2.1.2. Mô hình động lực học của xe điện	13
2.2. Thuật toán Time To Collision trên xe điện tự lái	23
2.3. Hệ thống điều khiển trên xe điện tự lái	27
Chương 3. THIẾT KẾ HỆ THỐNG PHẦN CỨNG	30
3.1. Tổng quan phần cứng trên xe điện tự lái	30
3.2. Tích hợp bộ điều khiển và cảm biến cho xe điện tự lái	31
3.3. Thiết kế tủ điện điều khiển trung tâm	34
3.3.1. Sơ đồ kết nối tổng quát tủ điện	34
3.3.2. Thiết kế tính năng điều khiển	35
3.4. Kết nối cảm biến Radar	36
3.5. Kết nối vi xử lý STM32F4	39

3.6.	Máy tính nhúng xử lý trung tâm.....	41
Chương 4.	THIẾT KẾ HỆ THỐNG PHẦN MỀM	44
4.1.	Tổng quan kết nối hệ thống phần mềm.....	44
4.2.	Xử lý dữ liệu cảm biến Radar	45
4.2.1.	Cấu hình ban đầu cho cảm biến	46
4.2.2.	Xử lý dữ liệu nhận về từ serial.....	49
4.3.	Thuật toán theo dõi đối tượng – Đầu vào dữ liệu thuật toán TTC	52
4.4.	Tính toán thời gian va chạm TTC, vận tốc, vị trí của các đối tượng.....	54
4.5.	Xây dựng bộ điều khiển tránh va chạm từ thuật toán TTC	59
4.6.	Thiết kế giao diện người dùng.....	62
Chương 5.	KẾT QUẢ THỰC HIỆN	64
5.1.	Kết quả thực hiện phần cứng.....	64
5.2.	Kết quả xử lý cảm biến Radar	66
5.2.1.	Kết quả phát hiện, theo dõi đối tượng	66
5.2.2.	Kết quả xác định vị trí đối tượng.....	69
5.2.3.	Kết quả xác định chuyển động đối tượng	72
5.3.	Kết quả điều khiển	73
5.3.1.	Kết quả mô phỏng	73
5.3.2.	Kết quả thực tế thuật toán TTC – Điều khiển tránh va chạm đối tượng tĩnh.....	76
5.3.3.	Kết quả thực tế thuật toán TTC – Điều khiển tránh va chạm đối tượng đang chuyển động	78
5.3.4.	Giao diện người dùng sử dụng	81
Chương 6.	ĐÁNH GIÁ, KẾT LUẬN, HƯỚNG PHÁT TRIỂN	83

6.1.	Đánh giá kết quả	83
6.2.	Kết luận đề tài.....	84
6.3.	So sánh sản phẩm ứng dụng trên thị trường.....	84
6.4.	Hướng phát triển.....	88

DANH MỤC HÌNH VẼ

Hình 1-1 Hình ảnh Kar-go tại triển lãm Goodwood (2019).....	5
Hình 1-2 Self-driving golf car được phát triển tại đại học Waterloo (Canada).....	6
Hình 1-3 Xe tự lái của FPT Software tại F-Town, quận 9 TP HCM	8
Hình 1-4 Hệ thống hỗ trợ lái xe tiên tiến ADAS.....	9
Hình 2-1 Hệ trục tọa độ gắn với xe	12
Hình 2-2 Quy ước thông số mô hình của xe	14
Hình 2-3 Thông số của chuyển động đánh lái	14
Hình 2-4 Liên hệ giữa các thông số góc và vận tốc góc trên mô hình.....	17
Hình 2-5 Vị trí tâm quay của xe khi đánh lái	18
Hình 2-6 Quá trình chuyển động của ô tô đến gần một vật thể đứng yên.....	24
Hình 2-7 Sơ đồ đơn giản tính giao điểm vectơ chuyển động của hai xe: $\alpha < 90^\circ$	25
Hình 2-8 Trường hợp ban đầu của va chạm, $\alpha < 90^\circ$ và $\alpha > 90^\circ$	26
Hình 2-9 Trường hợp ban đầu của va chạm, $\alpha = 180^\circ$	27
Hình 2-10 Tổng quan hệ thống điều khiển trên xe điện tự hành	27
Hình 2-11 Hệ thống các cảm biến được tích hợp vào xe điện hỗ trợ quá trình tự hành	28
Hình 3-1 Tổng quan kết nối phần cứng trên xe điện tự lái	30
Hình 3-2 Hệ thống điều khiển chưa có khả năng tự lái.....	31
Hình 3-3 Hệ thống điều khiển đã tích hợp khả năng tự lái	32
Hình 3-4 Tổng quan hệ thống chấp hành trên xe điện 4 chỗ Sanyo.....	33
Hình 3-5 Cấu trúc nguồn điện của xe	34
Hình 3-6 Sơ đồ kết nối các khối điều khiển bên trong tủ điện	35
Hình 3-7 Kết nối cảm biến Radar	36
Hình 3-8 mmwaveRadar của hãng Mistral.....	37
Hình 3-9 Kết nối vi xử lý STM32F4.....	39
Hình 3-10 Board STM32F407 Discovery	40
Hình 3-11 Kết nối máy tính nhúng trong hệ thống.....	42

Hình 3-12 Máy tính nhúng Jetson Xavier NX.....	42
Hình 4-1 Tổng quan kết nối hệ thống cho thuật toán TTC.....	44
Hình 4-2 Robot Operating System	45
Hình 4-3 Lưu đồ thực hiện xử lý dữ liệu trả về từ radar.....	46
Hình 4-4 Sơ đồ hoạt động của một chu kỳ tín hiệu quét (chirp) radar.....	47
Hình 4-5 Dữ liệu nhận được từ một gói tin radar	50
Hình 4-6 Thuật toán theo dõi Group Tracking	53
Hình 4-7 Khảo sát chuyển động của đối tượng	54
Hình 4-8 Khảo sát chuyển động của đối tượng khi velY > 0	56
Hình 4-9 Khảo sát chuyển động của đối tượng khi velY < 0	56
Hình 4-10 Khảo sát chuyển động của đối tượng khi velY > 0	57
Hình 4-11 Khảo sát chuyển động của đối tượng khi velY < 0	57
Hình 4-12 Khảo sát chuyển động của đối tượng khi velY trùng phương chuyển động	58
Hình 4-13 Khảo sát chuyển động của đối tượng khi velX = 0	58
Hình 4-14 Khảo sát chuyển động của đối tượng khi velY = 0	59
Hình 4-15 Giải thuật hoạt động của bộ điều khiển	60
Hình 4-16 Mô hình an toàn tích hợp	61
Hình 4-17 Tổng quan giao diện người dùng.....	63
Hình 5-1 Tủ điện điều khiển đặt trên xe.....	64
Hình 5-2 Chi tiết bên trong tủ điều khiển	65
Hình 5-3 Hình ảnh cảm biến radar trên xe điện	66
Hình 5-4 Kết quả phát hiện, theo dõi đối tượng thực tế (trái) và GUI (phải)	66
Hình 5-5 Kết quả phát hiện, theo dõi đối tượng người đi bộ	67
Hình 5-6 Kết quả phát hiện, theo dõi đối tượng xe máy	68
Hình 5-7 Kết quả phát hiện, theo dõi đối tượng xe ô tô.....	68
Hình 5-8 Kết quả phát hiện đối tượng bị nhiễu do cấu hình không tối ưu	69
Hình 5-9 Khoảng cách tối đa cảm biến có thể phát hiện đối tượng.	71
Hình 5-10 Mô phỏng môi trường hoạt động.....	74

Hình 5-11 Kết quả mô phỏng điều khiển tránh vật thể	75
Hình 5-12 Xe điện tự động giảm tốc khi đối tượng quá gần.....	76
Hình 5-13 Kết quả thực tế điều khiển tránh vật thể tĩnh.....	77
Hình 5-14 Đối tượng đang chuyển động đâm thẳng trực diện vào xe điện.....	79
Hình 5-16 Kết quả thực tế điều khiển tránh vật thể đang chuyển động	79
Hình 5-17 Màn hình radar được tích hợp trên GUI hiện tại	81
Hình 6-1 Telematics FMC125	85
Hình 6-2 TREK - 130 module ADAS.....	87
Hình 6-3 Cảnh báo lấn làn đường.....	87

DANH MỤC BẢNG

Bảng 3-1 Thông số xe điện 4 chỗ	32
Bảng 3-2 Thông số kỹ thuật cảm biến radar.....	38
Bảng 3-3 Thông số kỹ thuật vi xử lý STM32F407	41
Bảng 3-4 Thông số kỹ thuật của Jetson Xavier NX:.....	43
Bảng 4-1 Cấu trúc frameheader	50
Bảng 4-2 Cấu trúc frameTLV - type 1.....	51
Bảng 4-3 Cấu trúc frameTLV - type 7.....	51
Bảng 4-4 Cấu trúc frameTLV type - 10	52
Bảng 5-1 Giá trị khảo sát vị trí vật cản	70
Bảng 5-2 Kết quả xác định vị trí vật cản	70
Bảng 5-3 Giá trị khảo sát tốc độ của đối tượng	72
Bảng 5-4 Kết quả xác định tốc độ của đối tượng	73

TÓM TẮT LUẬN VĂN

Hiện nay xe điện tự hành ngày càng phát triển và được ứng dụng rộng rãi trong nhiều lĩnh vực đời sống. Các ứng dụng tự động này được phát triển nhằm thay thế con người thực hiện những nhiệm vụ chuyên biệt, giải phóng con người khỏi những tác vụ không thực sự cần đến sự kiểm soát của con người. Tuy nhiên quá trình tự động không cần con người kiểm soát cần có độ an toàn cao hơn, giám sát chặt chẽ để có thể ứng dụng vào sâu bên trong thực tiễn.

Luận văn trình bày quá trình nghiên cứu, xây dựng hệ thống tự động cho xe điện. Phát triển hướng đến mục tiêu an toàn cho mọi đối tượng tham gia giao thông. Tăng khả năng tự lái và hạn chế thấp nhất nguy cơ xảy ra va chạm khi xe điện đang hoạt động. Đạt được mục tiêu trên, nghiên cứu dùng cảm biến radar kết hợp ứng dụng thuật toán TTC để xây dựng hệ thống tránh va chạm, triển khai trên mô hình xe thực tế.

Kết quả đạt được sau khi thực hiện đề tài là hệ thống tránh va chạm có thể điều khiển được chuyển động của xe điện. Phát hiện, theo dõi đối tượng phía trước, cảnh báo và có khả năng tự đánh giá nguy hiểm để bảo vệ an toàn cho phương tiện. Ứng dụng được lý thuyết điều khiển vào thực tế. Đề tài nghiên cứu là cơ sở cho phát triển hệ thống điều khiển an toàn trên mô hình xe điện hiện tại sau này.

Chương 1. GIỚI THIỆU ĐỀ TÀI

1.1. Giới thiệu vấn đề nghiên cứu

1.1.1. Giới thiệu tổng quan xe điện tự hành

Trong những năm gần đây, thế giới ngày càng được nghe nhiều về sự phát triển của các phương tiện tự động không người lái. Có thể kể ra những ví dụ như robot chuyển hàng tự động trong nhà máy, xe ô tô tự lái, máy bay không người lái... Các ứng dụng tự động này được phát triển nhằm thay thế con người thực hiện những nhiệm vụ chuyên biệt, giúp nâng cao hiệu quả lao động hoặc đơn giản đó là giải phóng con người khỏi những tác vụ không thực sự cần sự kiểm soát của con người.

Ngày nay, cùng với việc phát triển của khoa học kỹ thuật, khả năng tính toán tăng cao của các bộ xử lý, kỹ thuật cơ điện tử tiên tiến, thiết bị cảm biến đa dạng và độ chính xác ngày càng cao, cùng với sự phát triển của các thuật toán mới, các yêu cầu cơ bản như đã nêu ở trên hầu như phần nào đó có thể đáp ứng được. Bên cạnh đó, xu thế xe chạy bằng điện cũng đang là một hướng đi được chú trọng phát triển của hầu hết tất cả các hãng xe cũng như những robot khác. Các hãng xe được cho là dẫn đầu xu thế khi mà trình độ phát triển của xe điện càng cao, mức độ tự lái càng cao. Đã có rất nhiều tập đoàn sản xuất xe hơi và công nghệ lớn trên thế giới đã tham gia vào cuộc chạy đua phát triển xe hơi công nghệ tự lái thông minh (gọi tắt là xe tự lái, xe tự hành) mà không cần đến bất cứ can thiệp nào từ bàn tay của con người, trong đó có những tên tuổi nổi bật như Tesla, Daimler, Google, ...

1.1.2. Sự cần thiết của đề tài nghiên cứu

Tuy nhiên, những phương tiện xe tự lái cho đến bây giờ đều ít nhiều vẫn còn tồn tại những hạn chế khi chưa thể hoàn toàn thay đổi con người trong việc xử lý các tác vụ phức tạp, di chuyển trong điều kiện môi trường phức tạp, hay những tình huống bất ngờ xảy ra. Bởi vậy, hầu hết các sản phẩm xe tự lái đều

chỉ thực hiện những nhiệm vụ chuyên biệt, điều kiện và môi trường hoạt động có tính đồng bộ. Người ta ước tính rằng 90% các vụ tai nạn là do lỗi của con người, chủ yếu là do mất tập trung, phán đoán sai hoặc thiếu hiểu biết về tình huống [1]. Do đó, có nhiều phương pháp đang được giới thiệu để giúp cải thiện nhận thức của người lái xe về môi trường và các hệ thống xử lý thông tin, cảnh báo người lái xe về các mối nguy tiềm ẩn và thậm chí hành động nếu cần thiết.

Mức độ an toàn được đánh giá thông qua khả năng xảy ra xung đột và mức độ nguy hiểm của xung đột. Mức độ nguy hiểm của xung đột được đánh giá bằng nguy cơ xảy ra va chạm giữa hai hay nhiều phương tiện. Sự xung đột là một tình huống mà hai hoặc nhiều xe tiến gần về phía nhau theo không gian và thời gian kéo theo nguy cơ va chạm nếu các xe này không thay đổi chuyển động (tốc độ hoặc quỹ đạo). Trên thế giới đã có rất nhiều chỉ tiêu gián tiếp được đưa ra để đánh giá nguy cơ xảy ra va chạm giữa hai xe như thời gian đến xung đột (Time to Collision – TTC), thời gian tiền xâm lấn (Post-Encroachment Time – PET), khoảng thời gian (Gap Time – GT), gia tốc giảm tốc (Decelerate Rate) [2] ... Trong số các chỉ tiêu này, thời gian đến xung đột Time to Collision - được định nghĩa là thời gian để hai xe có thể va chạm nếu chúng giữ nguyên tốc độ và quỹ đạo hiện tại - được dùng khá rộng rãi. Do vậy, trong đề tài này, TTC sẽ được lựa chọn để làm chỉ tiêu đánh giá về mức độ an toàn.

Ưu điểm của phương pháp này là rất trực quan, có thể phân tích nhiều khía cạnh của tai nạn như tần suất và mức độ nghiêm trọng của tai nạn, ứng phó với các tình huống rủi ro và điều chỉnh các biện pháp an toàn cho người ngồi trên xe và đặc điểm của vụ va chạm. Các hệ thống này sử dụng thông tin được thu nhận bởi các cảm biến để chúng có thể hoạt động trên hệ thống điều khiển và bảo vệ nhằm giảm xác suất và hậu quả của tai nạn. Chúng cho phép thực hiện các biện pháp trước vài giây. Một số hoạt động là phanh tự động, tác động tự động trên hệ thống lái để cải thiện góc tác động, kích hoạt bộ cảng

trước, chuẩn bị túi khí ... hoặc triển khai các biện pháp giảm thiểu tác động của tai nạn cho người đi bộ.

1.1.3. Tình hình trong và ngoài nước

1.1.3.1. Các nghiên cứu xe điện tự lái

Về xe tự lái, có nhiều các nghiên cứu được công bố liên quan đến việc phát triển hệ thống vận hành và điều khiển trên thế giới như nghiên cứu của Jie Ji, Amir Khajepour, William Melek, Yan-Jun Huang [3]. Nội dung của nghiên cứu này là bám quỹ đạo (path planning) và tránh va chạm (tracking) được phát triển để duy trì bám quỹ đạo và tránh va chạm cho các phương tiện tự lái. Đối với path planning, một quỹ đạo mong muốn được xây dựng dưới dạng kết hợp của các hàm lượng giác, của đường và hàm số mũ, để tránh va chạm giữa xe với chướng ngại vật. Tiếp theo, để tracking quỹ đạo dự kiến nêu trên, bộ điều khiển bám theo đường đi mang tên Multi-constrained Model Predictive Control (MMPC), nó tính toán góc lái phía trước để tránh xe va chạm khi đang di chuyển xe vượt chướng ngại vật. Cũng theo nghiên cứu này, mô phỏng Simulink và CarSim được thực hiện trong trường hợp có chướng ngại vật chuyển động. Kết quả mô phỏng được công bố cho thấy phương pháp được đề xuất trong nghiên cứu trên có hiệu quả đối với nhiều tình huống lái xe và bộ điều khiển bám đường đi dựa trên MMPC cung cấp hiệu suất và duy trì khả năng cơ động tốt.

Một số nghiên cứu ở nước ngoài đã được hoàn tất và bước vào giai đoạn chạy thử nghiệm, đánh giá sản phẩm và phân phối trên thị trường, tiêu biểu có thể kể đến Kar-go: Xe tự hành kiêm robot giao hàng tự động tại Anh. Xe robot Kar-go được thiết kế để giải quyết vấn đề giao hàng ở chặng cuối, nhằm tiết kiệm chi phí cho người dùng và cũng góp phần bảo vệ môi trường. Kar-go là xe điện sạc chở được 48 kiện hàng và có thể di chuyển trong khu vực thành thị cũng như các con ngõ không có tín hiệu GPS, nó sử dụng trí tuệ nhân tạo để học hỏi tuyến đường. Chiếc xe này có thể lái từ địa chỉ của người gửi tới chỗ người

nhận và bàn giao bưu kiện tự động bằng robot có sẵn. Khởi đầu là bán tự động, xe sẽ được tiếp tục cải tiến để tăng dần mức độ tự động. Ngoài ra, Kar-go cũng có một lái xe có thể can thiệp bất kỳ lúc nào để đảm bảo an toàn. Chiếc xe này sẽ tập trung giao các kiện hàng có kích thước nhỏ bằng hộp đựng giày và thử nghiệm giao thuốc y tế là mục đích đầu tiên khi đại dịch Covid xảy ra.



Hình 1-1 Hình ảnh Kar-go tại triển lãm Goodwood (2019)

Một sản phẩm tương tự khác là Xe tự lái trong sân gôn tại Đại học Waterloo (Canada) - có tốc độ tối đa là 20[km/h]. Theo giới thiệu thì người lái xe có thể trở thành một hành khách có công việc duy nhất là ngả lưng và tận hưởng chuyến đi. Sản phẩm này không dành cho đường phố công cộng, thay vào đó, chiếc xe phù hợp với những môi trường có nhịp độ chậm như sân bay, công viên giải trí... Như vậy, mục tiêu của chiếc xe tự lái này không phải để cạnh tranh với ô tô mà là đưa công nghệ này vào một nơi an toàn hơn, dễ áp dụng hơn và vận chuyển mọi người trong môi trường riêng tư, tốc độ thấp. Về công nghệ, chiếc xe sử dụng GPS của nó để đi theo một bản đồ định trước được lập trình trong máy tính. Một chùm tia laser được đặt trên nóc giúp xe phát hiện và tránh khỏi những chướng ngại vật.



Hình 1-2 Self-driving golf car được phát triển tại đại học Waterloo (Canada)

1.1.3.2. Các nghiên cứu Time To Collision công bố.

Ý tưởng về tính toán thời gian va chạm Time To Collision (TTC) lần đầu tiên được đề xuất bởi Hayward [4]. Định nghĩa đó là thời gian cần thiết để hai phương tiện va chạm nếu chúng tiếp tục ở tốc độ hiện tại và trên cùng một con đường. Hydén cho rằng giá trị TTC thấp hơn tương ứng với mức độ xung đột cao hơn [5]. Mặc dù điểm này đã được tranh luận trong các tài liệu đánh giá an toàn, nhưng có vẻ như rõ ràng rằng các giá trị TTC thấp hơn tương ứng với xác suất va chạm cao hơn [6] [7]. Do đó, TTC thường được coi là một biện pháp chính và hiệu quả trong đánh giá an toàn giao thông, đặc biệt là trong đánh giá xung đột. Năm 2003, FHWA đã phát hành một báo cáo giới thiệu việc sử dụng các mô hình mô phỏng giao thông để có được các biện pháp an toàn thay thế [8]. Trong tài liệu đó, các thuật toán tính toán để tính toán các biện pháp an toàn thay thế cho các loại xung đột khác nhau đã được mô tả và các sơ đồ ví dụ được cung cấp để minh họa các phép tính bằng đồ thị.

Bên cạnh vai trò là một chỉ số an toàn, việc sử dụng TTC như một tín hiệu để ra quyết định trong giao thông đã được một số nghiên cứu đề xuất. Ví dụ, Horst

nhận thấy rằng cả quyết định bắt đầu phanh và bản thân việc kiểm soát quá trình phanh đều có thể dựa trên thông tin liên quan đến TTC [9].

1.1.3.3. Tình hình trong nước:

Gần đây, vào năm 2020, một đề tài được luận văn được công bố với tên “Xe tự hành bám quỹ đạo, phát hiện và tránh vật cản sử dụng trí tuệ nhân tạo” của nhóm sinh viên Huỳnh Trung Tín, Phan Thị Thanh Hiền tại Đại học Bách Khoa thành phố Hồ Chí Minh. Về kết quả đạt được: Bộ điều khiển bám quỹ đạo Stanley Controller tối ưu tính toán xử lý trong vi điều khiển. Độ chính xác tốt với 3[cm] sai số bám quỹ đạo. Về khả năng phát hiện, theo dõi và xác định vị trí của vật cản, các thuật toán phát hiện, theo dõi và xác định vị trí của vật cản được lựa chọn dựa trên tốc độ xử lý nhanh, nhưng độ chính xác chưa phải là cao nhất. Tuy nhiên, đối với mô hình xe tự hành trong đề tài này, tốc độ di chuyển chậm và chỉ xem xét các vật cản ở khoảng cách gần, các thuật toán được lựa chọn vẫn đáp ứng đủ nhu cầu phát hiện, theo dõi và xác định vị trí vật cản của mô hình. Về việc điều khiển mô hình tránh vật cản, với thuật toán tránh vật cản được xây dựng dựa trên phương pháp The Curvature – Velocity Method, mô hình thực hiện tốt việc tránh các vật cản đứng yên. Tuy nhiên, đối với các vật cản chuyển động, do thuật toán chưa xét đến vận tốc của các vật cản, trong một số trường hợp mô hình sẽ di chuyển chấn phía trước hướng chuyển động của vật cản gây mất an toàn. Ngoài ra, nhóm thực hiện đề tài còn tạo ra một giao diện người dùng để dễ dàng điều khiển.

FPT Software đã giới thiệu mô hình thử nghiệm xe tự lái do công ty tự nghiên cứu phát triển từ tháng 8 năm 2016 là “**Xe tự lái của FPT Software**”. Nhóm nghiên cứu đã tiến hành lắp ráp và lập trình xe tự hành trong 3 tháng với các linh kiện nhập từ nhiều nước. Mẫu xe tự lái ở khu vực TPHCM đã chạy trong vòng 200 giờ trong khuôn viên tòa nhà F-Town (thuộc Khu công nghệ cao TP.HCM) để kiểm tra nhiều tình huống khác nhau xảy ra trên đường đi. Công nghệ xe tự lái của FPT đang sử dụng phối hợp nhiều ứng dụng và công

nghệ mới để thu thập dữ liệu, theo dõi môi trường xung quanh. Xe có thể tự động nhận diện vật cản, leo trèo bãy đất, bãy cát, leo dốc đến 45 độ, lội nước ngập sâu 10cm, lội sình lầy, lội tuyết... Chiếc ô tô tự lái này chạy với tốc độ 20[km/giờ], có khả năng điều khiển chạy thẳng, tránh xe chạy ngược chiều, chuyển làn ở các khúc cua. Tuy vậy, sản phẩm này vẫn đang trong quá trình phát triển, chưa sản xuất và phân phối trên thị trường.



Hình 1-3 Xe tự lái của FPT Software tại F-Town, quận 9 TP HCM

Có thể đề cập đến hệ thống hỗ trợ lái xe ADAS - viết tắt của Advanced Driver Assistance Systems có nghĩa là hệ thống hỗ trợ lái xe tiên tiến. Đây là một hệ thống điện tử hỗ trợ người điều khiển phương tiện lái xe an toàn và thuận tiện hơn.



Hình 1-4 Hệ thống hỗ trợ lái xe tiên tiến ADAS

Hệ thống an toàn trên ô tô ADAS đã được chứng minh có thể làm giảm tỷ lệ tai nạn khi tham gia giao thông bằng cách giảm khả năng mắc lỗi của người điều khiển phương tiện thông qua các chức năng cảnh báo thông minh như: Cảnh báo va chạm phía trước, cảnh báo lệch làn đường, phát hiện điểm mù, phát hiện tài xế buồn ngủ, kiểm soát đổ đèo, hệ thống nhìn ban đêm, hệ thống hỗ trợ đỗ xe... Theo nghiên cứu của Hiệp hội Bảo hiểm An toàn Đường bộ Hoa Kỳ, hệ thống tránh va chạm có thể giảm 27% tai nạn ô tô từ phía sau; hệ thống cảnh báo lệch làn đường có thể giảm 21% số vụ tai nạn thương vong; hệ thống phát hiện điểm mù có thể giảm 14% tai nạn va chạm trong làn đường

Hiện nay, hệ thống an toàn trên ô tô ADAS đã được các nhà sản xuất trang bị trên hầu hết các mẫu xe. Điển hình như các dòng xe ô tô tại Việt Nam của VinFast: VinFast President, VinFast Lux SA2.0, VinFast Lux A2.0, VinFast Fadil. Bên cạnh đó, ngày càng có nhiều tính năng mới tiên tiến được tích hợp vào ADAS giúp người lái tự tin khi tham gia giao thông. [10]

1.2. Mục tiêu của đề tài

1.2.1. Ý nghĩa khoa học

Ứng dụng thuật toán Time to Collision để tính toán thời gian xảy ra va chạm vào phương tiện xe điện tự lái. Với khả năng tự động theo dõi đối tượng phía trước hướng di chuyển của xe và tự động điều khiển để hạn chế thấp nhất nguy cơ xảy ra va chạm (hệ thống tránh va chạm).

Ngoài ra, nghiên cứu còn mở rộng thêm vào việc tìm những giải pháp cải thiện độ chính xác của kết quả điều khiển, giúp xe vừa hoạt động ổn định và an toàn, có khả năng phân tích, xử lý khi có lỗi xảy ra.Thêm vào đó, thuật toán xử lý ảnh được ứng dụng để đóng góp vào việc nhận diện môi trường xung quanh trong quá trình phương tiện di chuyển.

1.2.2. Phạm vi ứng dụng

Để làm rõ mục đích cuối cùng của đề tài, phân biệt với các hướng nghiên cứu khác trên thị trường hiện nay, dưới đây là phạm vi ứng dụng của đề tài:

- Xe tự lái có phạm vi hoạt động là những khu vực cố định đã được xác định sẵn trong bản đồ số, không phải là tất cả những cung đường cũng không phải trên các xa lộ. Trong khu vực cố định đó đã xác định sẵn vị trí những mục tiêu, có thể là vị trí đón trả khách, vị trí giao nhận hàng hóa. Phương tiện có thể được lên lịch trước cho những nhiệm vụ di chuyển trong phạm vi cho phép hoạt động hoặc có thể nhận lệnh trực tiếp từ giao diện máy tính trên xe. Ví dụ khu vực hoạt động: sân golf, xe bus chuyển cố định, khu tham quan du lịch, sân bay.
- Sản phẩm nâng cao trải nghiệm của người sử dụng khi chỉ cần tham gia ít hơn đến quá trình vận hành phương tiện khi xe có thể trực tiếp đưa ra một số quyết định trong quá trình di chuyển.
- Phục vụ cho các ứng dụng cần nguồn nhiên liệu sạch, ở đây là sử dụng điện, thêm vào đó là mức độ tiếng ồn thấp khi hoạt động.
- Tập trung vào hướng ứng dụng của sản phẩm trong thực tế.

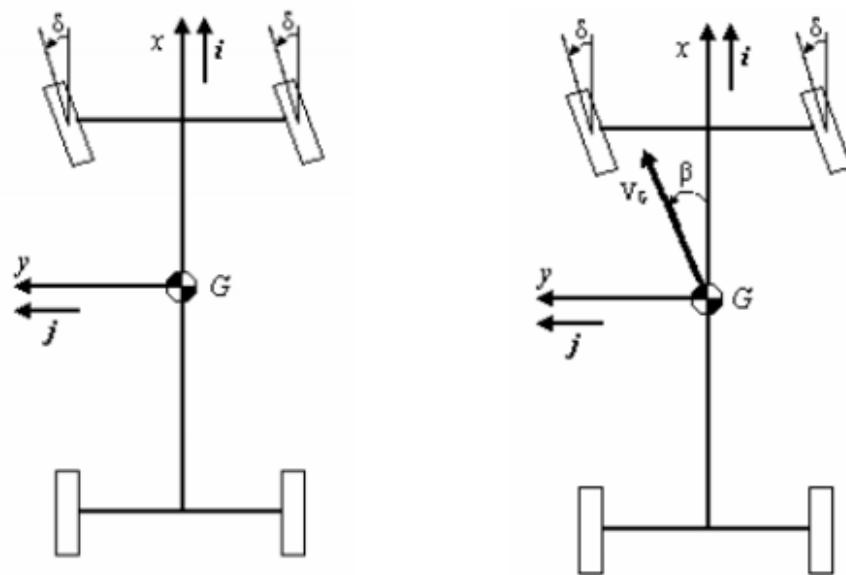
- Phương tiện có khả năng nhận biết được những đối tượng cơ bản trong môi trường xung quanh, theo dõi, xác định khoảng cách an toàn với các phương tiện và đối tượng khác. Qua đó đưa ra quyết định phù hợp để vận hành an toàn.

Chương 2. CƠ SỞ LÝ THUYẾT

2.1. Mô hình động lực học xe điện 4 bánh

2.1.1. Hệ trục tọa độ gắn với xe – mô phỏng mô hình xe điện

Hệ tọa độ được dùng là Body Axis, được gán cố định với xe. Gốc tọa độ của nó nằm ở trọng tâm của xe, và các hướng được đặc trưng với các véc tơ (i , j , k), trục x được xác định song song với trục dọc của xe và hướng thẳng về phía trước, trục z là trục giao với đường và hướng lên, trục y vuông góc với đường và hướng sang bên trái. Theo quy ước, vận tốc góc yaw tại G là dương nếu nó ngược chiều kim đồng hồ (ISO 8855). Trong hầu hết các nghiên cứu liên quan đến xử lý và điều khiển hướng, chỉ xem xét mặt phẳng $x-y$ của xe. Trục thẳng đứng, z , thường được sử dụng trong nghiên cứu các vấn đề về độ ổn định của xe.



Hình 2-1 Hệ trục tọa độ gắn với xe

Các định nghĩa có liên quan cho các biến liên quan trong nghiên cứu này:

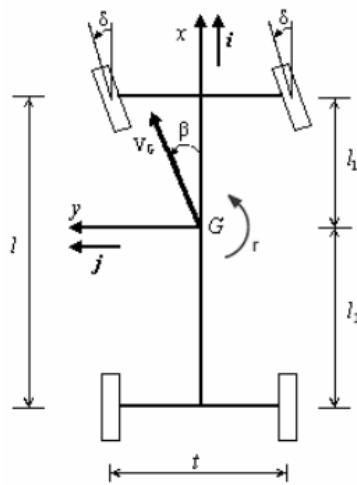
- Hướng chuyển động theo chiều dọc (Longitudinal direction): hướng chuyển động tịnh tiến của xe. Có hai cách khác nhau để xác định hướng di chuyển tịnh tiến của xe, một cách lấy góc nhìn từ thân xe và

một cách khác là nhìn từ một điểm chuẩn cố định bên ngoài xe. Trước đây Longitudinal direction thường được sử dụng khi xử lý gia tốc và vận tốc của xe, sau này nó được sử dụng trong thông tin vị trí của xe liên quan đến điểm xuất phát và điểm kết thúc mong muốn.

- Hướng chuyển động theo chiều ngang (Lateral direction): hướng chuyển động theo phương ngang của xe. Một lần nữa, có hai cách lấy góc nhìn từ thân xe và cách nhìn từ một điểm chuẩn cố định. Các nghiên cứu thường cho thấy hướng này phức tạp hơn hướng dọc vì các giá trị của gia tốc ngang hoặc vận tốc ngang có thể làm giảm độ ổn định và khả năng điều khiển của xe.
- Góc lệch của bánh xe hay góc đánh lái (Tyre slip angle): Là góc lệch giữa bánh trước của xe so với trục dọc của xe.
- Góc lệch (Sideslip angle): là góc giữa trục x và véc tơ vận tốc biểu thị vận tốc tức thời của xe tại điểm đó trên đường đi. Cần nhấn mạnh rằng góc lệch này khác với góc đánh lái liên quan đến lốp xe. Thường thì góc lệch của xe được tính bằng tỷ số giữa vận tốc dài và vận tốc ngang.

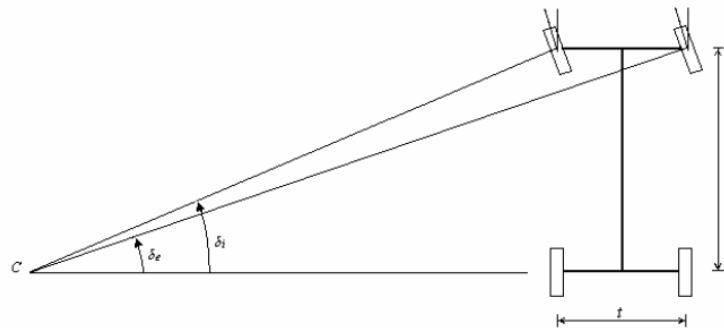
2.1.2. Mô hình động lực học của xe điện

Từ các phát biểu ở trên, chúng ta có thể tiến hành xây dựng mô hình toán học, mô hình được tham khảo từ [4]. Coi xe có khối lượng m (kg) và trọng tâm G , chiều dài l (m), ngang t (m), khoảng cách từ tâm G đến trục bánh trước là l_1 (m), khoảng cách từ tâm G đến trục bánh sau là l_2 (m), góc lệch β (rad), góc lệch bánh trước δ (rad).



Hình 2-2 Quy ước thông số mô hình của xe

- V_G là tốc độ tuyệt đối của xe tính ở trọng tâm, vị trí của trọng tâm là cố định qua độ dài l_1 và l_2 .
- Chuyển động đánh lái được thể hiện trong hình dưới đây. Lưu ý rằng góc lái của bánh xe bên trong δ_i lớn hơn góc lái của bánh xe bên ngoài δ_e .



Hình 2-3 Thông số của chuyển động đánh lái

Từ đây ta có quan hệ động học:

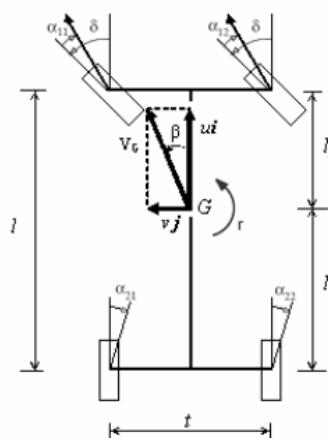
$$\frac{t}{l} = \frac{1}{\tan(\delta_e)} - \frac{1}{\tan(\delta_i)} \quad (1)$$

Và

$$\delta_e = \delta_i - \frac{t}{l} \delta_i^2 + O(\delta_i^3) \quad (2)$$

Với góc lái nhỏ, ta có δ_i xấp xỉ δe và bằng δ , theo các phát biểu ở trên, chúng ta có thể tiến hành lập công thức toán học cho mô hình bằng ba nhóm phương trình:

- Phương trình tương đương



Đặt góc α là chênh lệch giữa hướng vectơ vận tốc của bánh xe so với trục dọc và góc lệch tổng quát của xe cũng so với trục dọc. Góc này sẽ mang giá trị dương khi quay ngược chiều kim đồng hồ. Giả sử xe là một khối cứng khi trọng tâm không thay đổi so với toàn bộ các thành phần của xe, do đó vận tốc của trọng tâm V_G và vận tốc góc r đặc trưng bằng góc lệch α_{ij} của bốn bánh xe (với $i = 1, 2$ chỉ bánh xe trước hoặc sau và $j = 1, 2$ chỉ bánh xe trái hoặc phải). Ta biểu thị vận tốc tuyệt đối của trọng tâm V_G so với vectơ tham chiếu i, j và k , vectơ vận tốc được mô tả như sau:

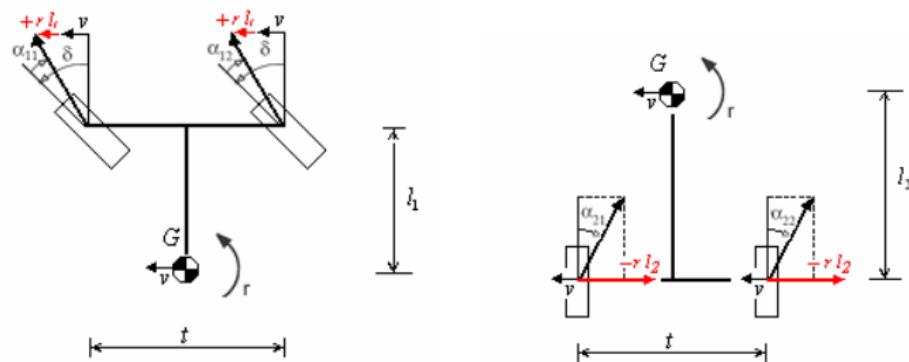
$$V_G = u_i + v_j \quad (3)$$

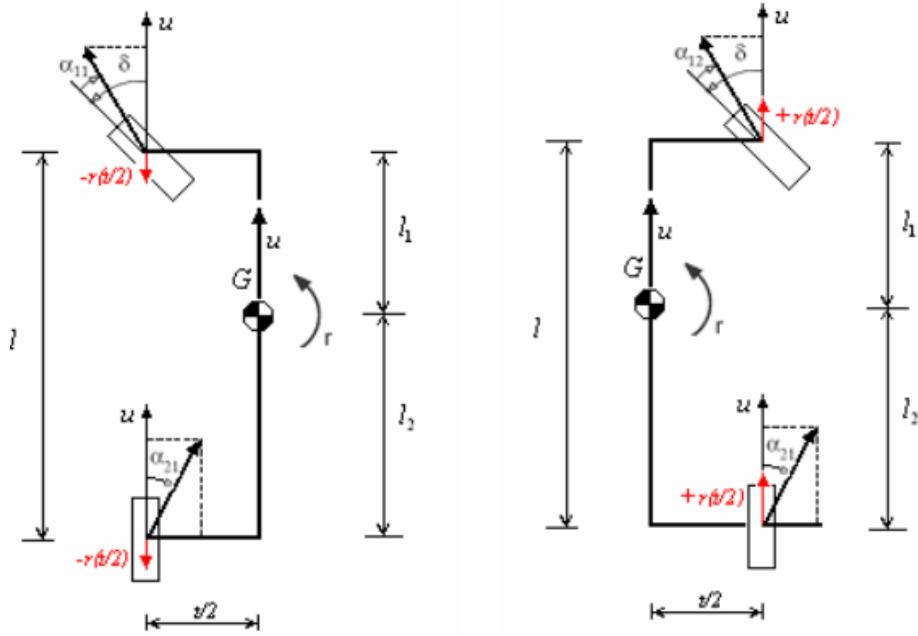
Thành phần dọc của vectơ vận tốc u được gọi là “feed velocity” hay vận tốc dọc, trong khi thành phần bên v được gọi là “lateral velocity” hay vận tốc ngang. Vì trọng tâm không chuyển động trong hệ tọa độ nên công thức đã nêu ở trên chỉ mô tả các thành phần vận tốc theo hai hướng. Về mặt toán học, góc lệch

của cả xe trong hệ tọa độ so với trục x (trục dọc của xe) được mô tả bằng phương trình sau:

$$\beta = \arctan\left(\frac{v}{u}\right) \quad (4)$$

Nếu chúng ta xem xét vận tốc tuyệt đối của xe tại trọng tâm, chúng ta có thể liên kết góc trượt α_{ij} với vận tốc góc r cùng các thành phần tịnh tiến của vận tốc u và v . Nếu xét theo phương ngang, tại bánh xe phía trước, nó có vận tốc bằng $v + r.l_1$, của bánh sau bằng $v - r.l_2$. Nếu xét theo phương dọc, các bánh xe ở bên trái (đối với trục x) có vận tốc bằng $u - r(t/2)$ và các bánh ở bên phải có vận tốc bằng $u + r(t/2)$.





Hình 2-4 Liên hệ giữa các thông số góc và vận tốc góc trên mô hình

Các góc lệch là hàm của hai biến chuyển động cơ bản, v (hoặc β) và r . Mối quan hệ của chúng được cho dưới đây:

$$\begin{aligned} \tan(\delta - \alpha_{11}) &= \frac{v + rl_1}{u - r(t/2)} \\ \tan(\delta - \alpha_{12}) &= \frac{v + rl_1}{u + r(t/2)} \\ \tan(\delta - \alpha_{21}) &= \frac{v - rl_2}{u - r(t/2)} \\ \tan(\delta - \alpha_{22}) &= \frac{v - rl_2}{u + r(t/2)} \end{aligned} \quad (5)$$

Trong điều kiện bình thường, ta có:

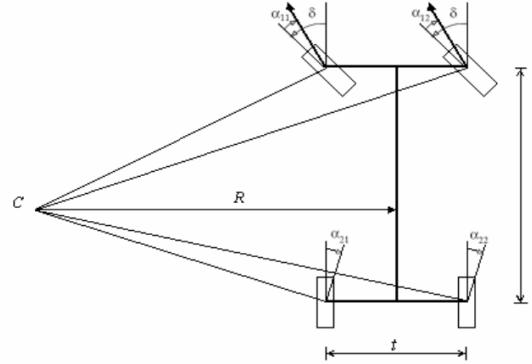
$$u = |r| \frac{t}{2} \quad (6)$$

Điều này có nghĩa là các bánh xe quay trên cùng một trục có cùng góc lệch. Vì lý do này, có thể biểu thị bằng góc lệch α_f đối với một trong hai bánh trước và α_r là góc lệch đối với bánh sau. Theo đó:

$$\tan(\delta - \alpha_f) = \frac{v + rl_1}{u - r(t/2)} \quad (7)$$

$$\tan(\delta - \alpha_r) = \frac{v - rl_2}{u}$$

Như thể hiện trong các hình dưới đây, vị trí của tâm quay của xe khi đánh lái:



Hình 2-5 Vị trí tâm quay của xe khi đánh lái

Để đơn giản hóa, ta có thể coi rằng vận tốc dọc luôn lớn hơn vận tốc ngang và vận tốc khác, về mặt toán học, ta có:

$$u = |v + rl_1| \quad (8)$$

$$u = |v - rl_2|$$

Theo đó, ta có thể xấp xỉ:

$$\delta - \alpha_f = \frac{v + rl_1}{u} \quad (9)$$

$$\delta - \alpha_r = \frac{v - rl_2}{u}$$

Cuối cùng, chúng ta có Phương trình gia tốc tuyến tính:

$$\begin{aligned}\alpha_f &= \delta - \frac{v + rl_1}{u} \\ \alpha_r &= -\frac{v - rl_2}{u}\end{aligned}\tag{10}$$

Trong đó:

- δ : Góc lái (rad)
 - l_1 : Khoảng cách trọng tâm đến trục trước của xe (m)
 - l_2 : Khoảng cách trọng tâm đến trục sau của xe (m)
 - r : Vận tốc quay của xe tại trọng tâm (rad/s)
 - v : Vận tốc ngang của xe (m/s)
 - u : Vận tốc dài (dọc) của xe (m/s)
- Phương trình cân bằng

Về cơ bản, phương trình cân bằng chỉ có thể được viết sau khi tính toán gia tốc và ước lượng các lực và momen tác dụng lên xe.

Xác định gia tốc:

Biểu thức của gia tốc tại trọng tâm, a_G , có thể thu được ngay lập tức suy ra từ biểu thức vận tốc V_G , là hàm của thời gian t :

$$a_G = \frac{dV_G}{dt} = \dot{u}i + urj + \dot{v}j - vri = (\dot{u} - vr)i + (\dot{v} + ur)j = a_x i + a_y j\tag{11}$$

Trong đó: $\frac{d_i}{dt} = r_j$ và $\frac{d_j}{dt} = -r_i$

Theo đó đó, gia tốc của trọng tâm được chia thành các gia tốc dài và gia tốc ngang:

$$\begin{aligned}a_x &= \dot{u} - vr \\ a_y &= \dot{v} + ur\end{aligned}\tag{12}$$

Lực tác động và mô men:

Về cơ bản có ba loại lực tác động lên xe:

- Lực hấp dẫn tác động vào trọng tâm
- Lực của lốp tiếp xúc với mặt đất ở bốn bánh xe
- Lực cản của không khí.

Theo hệ tọa độ cố định ($x, y, z; G$), ba lực trên được đặt là X, Y, Z . Có thể viết phương trình cân bằng dùng Định luật Newton về vật thể cứng chuyển động trong chuyển động phẳng, có khối lượng m và mô men quán tính J theo trục z . Tổng hợp lại ta có các quan hệ sau:

$$\begin{aligned} ma_x &= X \\ ma_y &= Y \\ J\dot{r} &= Z \end{aligned} \tag{13}$$

Trong đó X, Y và Z lần lượt là tổng hợp của các lực theo phương dọc, phương ngang và quay.

Lực quan trọng nhất tác dụng lên xe là lực kéo lốp - đất, được xác định bởi F_{xij} và F_{yij} là các thành phần dọc và ngang của lực cản vào điểm tiếp xúc với mặt đất. Lực cản khí động học tỷ lệ với vận tốc bình phương và lực theo phương ngang. Theo giả thuyết về góc lái, và đặc biệt đối với góc lái dưới 15 độ, chúng ta có thể tuyến tính hóa các mối quan hệ cân bằng. Các phương trình cân bằng được hiển thị dưới đây:

$$\begin{aligned} m(\dot{u} - vr) &= (F_{x11} + F_{x12}) - (F_{y11} + F_{y12})\delta + (F_{x21} + F_{x22}) - F_{xaero} \\ m(\dot{v} + ur) &= (F_{x11} + F_{x12})\delta + (F_{y11} + F_{y12}) + (F_{y21} + F_{y22}) \\ J\dot{r} &= [(F_{x11} + F_{x12})\delta + (F_{y11} + F_{y12})]I_1 - (F_{y21} + F_{y22})I_2 + \\ &\quad - [(F_{x11} - F_{x12})\delta + (F_{x21} - F_{x22}) - (F_{y11} + F_{y12})\delta]\frac{t}{2} \end{aligned} \tag{14}$$

Trong đó F_{xaero} là lực khí động học do lực cản không khí tác dụng lên xe.

$$F_{xa} = \frac{1}{2} \rho S C_x u^2 \quad (15)$$

Do tính đối xứng của xe, ta có thể ký hiệu chung các lực tác dụng lên cùng một phương, như được thể hiện trong các phương trình sau:

$$\begin{aligned} F_{x1} &= F_{x11} + F_{x12} \\ F_{x2} &= F_{x21} + F_{x22} \\ F_{y1} &= F_{y11} + F_{y12} \end{aligned} \quad (16)$$

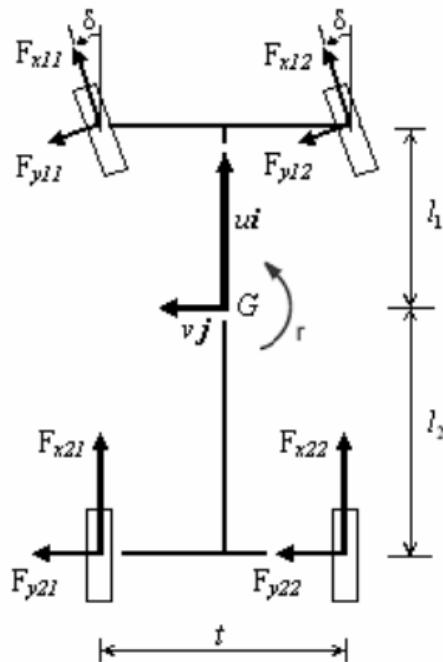
Nếu tính đến việc mô men kéo cho hai bánh xe ở cùng một trục bằng một bộ vi sai thông thường (không bị khóa tự động) thì mô men cho cả hai bánh xe là như nhau, từ đó ta có thể có một phương trình đơn giản khác:

$$\begin{aligned} F_{x11} &= F_{x12} \\ F_{x21} &= F_{x22} \end{aligned} \quad (17)$$

Ngay cả khi các bánh xe ở cùng một trục làm việc với cùng một góc trượt, các lực tác dụng lên phương ngang ở cùng một trục không có cùng giá trị, vì lý do này, $F_{y11} \neq F_{y12}$.

$$\begin{aligned} m(\dot{u} - vr) &= F_{x1} - F_{y1}\delta + F_{x2} - F_{xa} \\ m(\dot{v} + ur) &= F_{x1} + F_{y1} + F_{y2} \\ J\dot{r} &= [F_{x1}\delta + F_{y1}]l_1 - F_{y2}l_2 \end{aligned} \quad (18)$$

- Phương trình chuyển động bánh xe



Để hoàn thành mô hình động lực học của xe, chúng ta phải xác định tác động lên mỗi bánh xe. Các lực tác dụng lên phương ngang F_{yij} , lực tác dụng lên phương dọc F_{xij} và tải trọng F_{zij} . Chỉ xét đến sự biến thiên hàm về góc lệch, chúng ta sẽ xây dựng mô hình xe từ chuyển động của các bánh xe và có thể tích hợp phương trình chuyển động. Mô hình cho bánh xe được đơn giản hóa nhất là một hàm tuyến tính, biểu diễn giữa lực theo phương ngang (hay lực vào cua) và góc trượt. Các liên kết này được thể hiện trong phương trình sau:

$$F_{yij} = C_{\alpha_{ij}} \alpha_{ij} \quad (19)$$

Khi $\alpha_{i1} = \alpha_{i2}$ ta có:

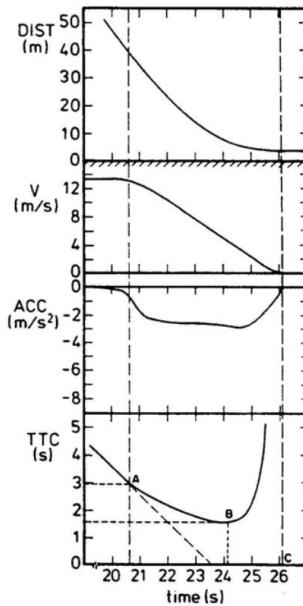
$$F_{yi} = C_{\alpha_i} \alpha_i \quad (20)$$

Trong điều kiện lý tưởng, C_i là độ cứng khi vào cua của lốp, làm việc với góc trượt nhỏ, có độ lớn tầm $15 \div 20$ độ trên đường khô ráo.

2.2. Thuật toán Time To Collision trên xe điện tự lái

Thời gian xảy ra va chạm Time To Collision (TTC) thường được sử dụng làm thước đo đánh giá rủi ro cho các phân tích an toàn giao thông. TTC thường là một yếu tố quan trọng trong quá trình ra quyết định quản lý quỹ đạo của người lái xe. TTC đánh giá cường độ tương tác giữa các phương tiện. Tuy nhiên, việc tính toán TTC không đơn giản. Dự đoán về tương tác trong tương lai giữa các phương tiện liên quan đến việc tạo quỹ đạo dự đoán cho phương tiện chủ thể cũng như tất cả các phương tiện khác có thể xảy ra tương tác để xem liệu va chạm có thể xảy ra hay không.

Trong nghiên cứu về kỹ thuật xung đột giao thông, TTC đã được chứng minh là một thước đo hiệu quả để đánh giá mức độ nghiêm trọng của xung đột. Hayward đã định nghĩa TTC là thời gian cần thiết để hai xe va chạm nếu chúng tiếp tục ở tốc độ hiện tại và trên cùng một đoạn đường [4]. Để minh họa TTC, Hình 2-7 cho thấy điều gì sẽ xảy ra khi một chiếc ô tô đến gần một vật thể đứng yên. TTC_{br} đại diện cho thời gian còn lại tối thiểu để xe tiếp tục chuyển động tính tại thời điểm bắt đầu hành động né tránh. TTC_{min} là thời gian tối thiểu đạt được trong quá trình tiếp cận của hai xe trong trường hợp có nguy cơ va chạm. TTC_{min} được coi là chỉ số cho mức độ nghiêm trọng của một vụ va chạm. Về nguyên tắc, TTC_{min} càng thấp thì nguy cơ va chạm càng cao TTC_{min} cho biết mức độ sắp xảy ra va chạm trên thực tế.

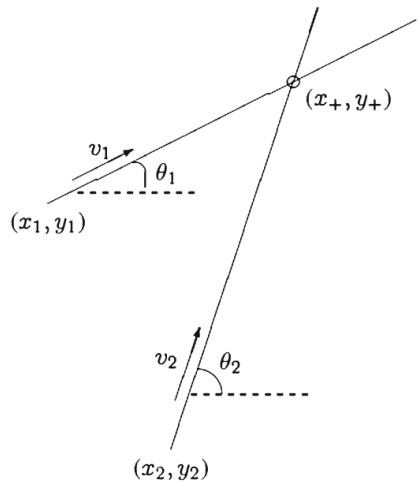


Hình 2-6 Quá trình chuyển động của ô tô đến gần một vật thể đứng yên

Hình trên thể hiện quá trình chuyển động của một ô tô đến gần một vật thể đứng yên; DIST - khoảng cách tới vật thể, V - vận tốc, ACC - gia tốc và TTC - thời gian để va chạm dựa trên sự không đổi của tốc độ và góc hướng. Điểm A cho thời điểm TTC_{br} và điểm B là thời điểm TTC_{min} .

Chi tiết về tính toán của TTC:

Một quy trình để tính toán thời gian va chạm giữa hai xe được trình bày. Dữ liệu được xem xét là vị trí ban đầu của các phương tiện, tốc độ và hướng của chúng, cùng với giả thiết rằng các phương tiện là hai điểm lý tưởng Hình 2-8.



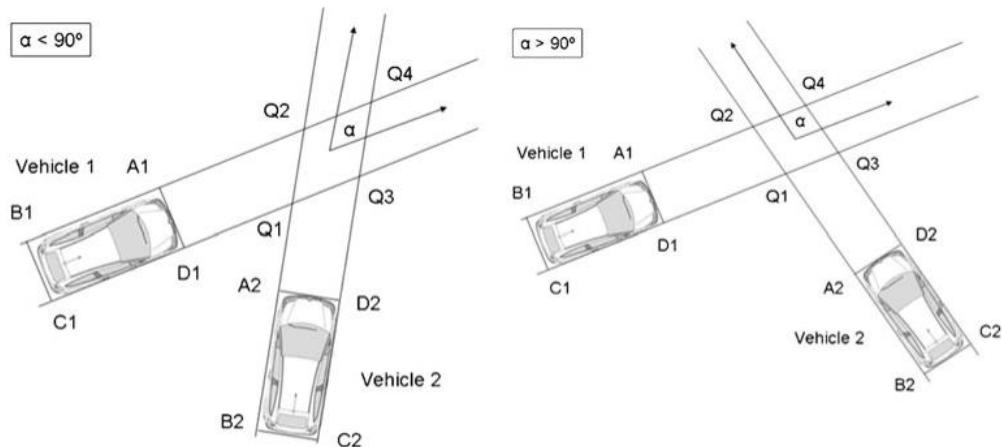
Hình 2-7 Sơ đồ đơn giản tính giao điểm vectơ chuyển động của hai xe: $\alpha < 90^\circ$

$$x_+ = \frac{(y_2 - y_1) - (x_2 \tan \theta_2 - x_1 \tan \theta_1)}{\tan \theta_1 - \tan \theta_2} \quad (21)$$

$$y_+ = \frac{(x_2 - x_1) - (y_2 \cot \theta_2 - y_1 \cot \theta_1)}{\cot \theta_1 - \cot \theta_2}$$

Sau khi tìm thấy điểm giao cắt, một quy trình được thiết lập để tính toán thời gian va chạm, bao gồm tính toán thời gian mỗi phương tiện đi đến giao điểm. Khi hai thời điểm này trùng nhau, đó là thời điểm va chạm (TTC). Để tính đến một biên độ an toàn bù đắp cho những đơn giản hóa được thực hiện.

Có thể phân biệt hai trường hợp va chạm có thể tính đến góc α giữa vectơ chuyển động của cả hai xe: $\alpha < 90^\circ$ và $\alpha > 90^\circ$. Trong phạm vi nghiên cứu, ở đây ta chỉ xét đến trường hợp $\alpha > 90^\circ$



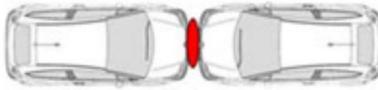
Hình 2-8 Trường hợp ban đầu của va chạm, $\alpha < 90^\circ$ và $\alpha > 90^\circ$

Để tính toán giá trị TTC, vi sai tốc độ tại thời điểm t được giả định là không đổi trong quỹ đạo va chạm giả định của các phương tiện cho đến khi t tức thời. Giá trị TTC càng cao, tình huống càng an toàn. Thời gian xảy ra va chạm của tổ hợp người lái xe thứ i tại thời điểm t đối với điểm giao cắt có thể được tính bằng:

$$TTC_i = \frac{X(t) - X_i(t)}{\dot{X}_i(t)} \quad (22)$$

Trong đó, X là vị trí giả định điểm giao cắt xảy ra va chạm. X_i là vị trí của xe thứ i. \dot{X}_i là tốc độ của xe thứ i. Khi những thời điểm TTC_i trùng nhau, đó là thời điểm va chạm.

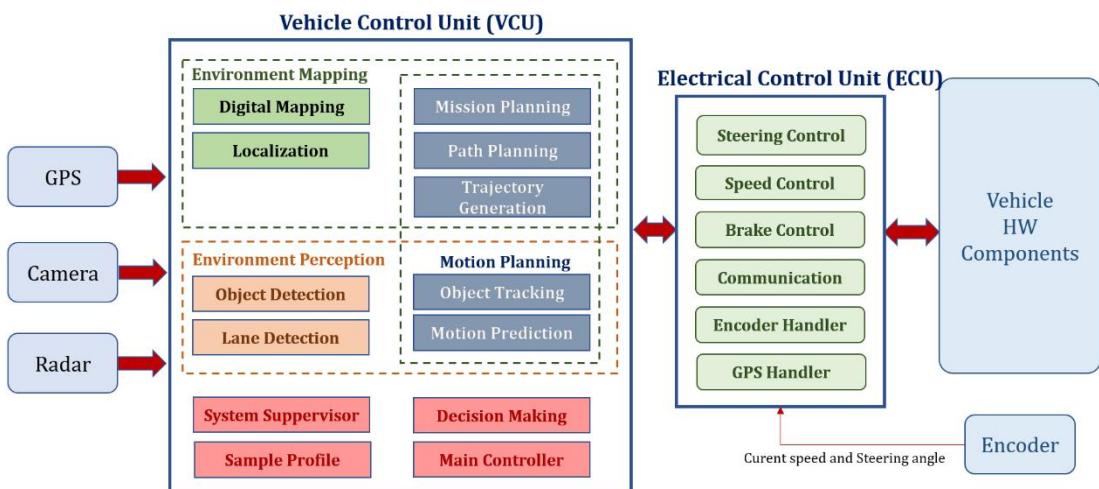
Ngoài ra, khi trường hợp $\alpha = 180^\circ$, ta có thể sử dụng công thức sau để tính thời gian xảy ra va chạm giữa hai đối tượng:



Hình 2-9 Trường hợp ban đầu của va chạm, $\alpha = 180^\circ$

$$TTC = \frac{|X_1(t) - X_2(t)|}{|\dot{X}_1(t) - \dot{X}_2(t)|} \quad (23)$$

2.3. Hệ thống điều khiển trên xe điện tự lái

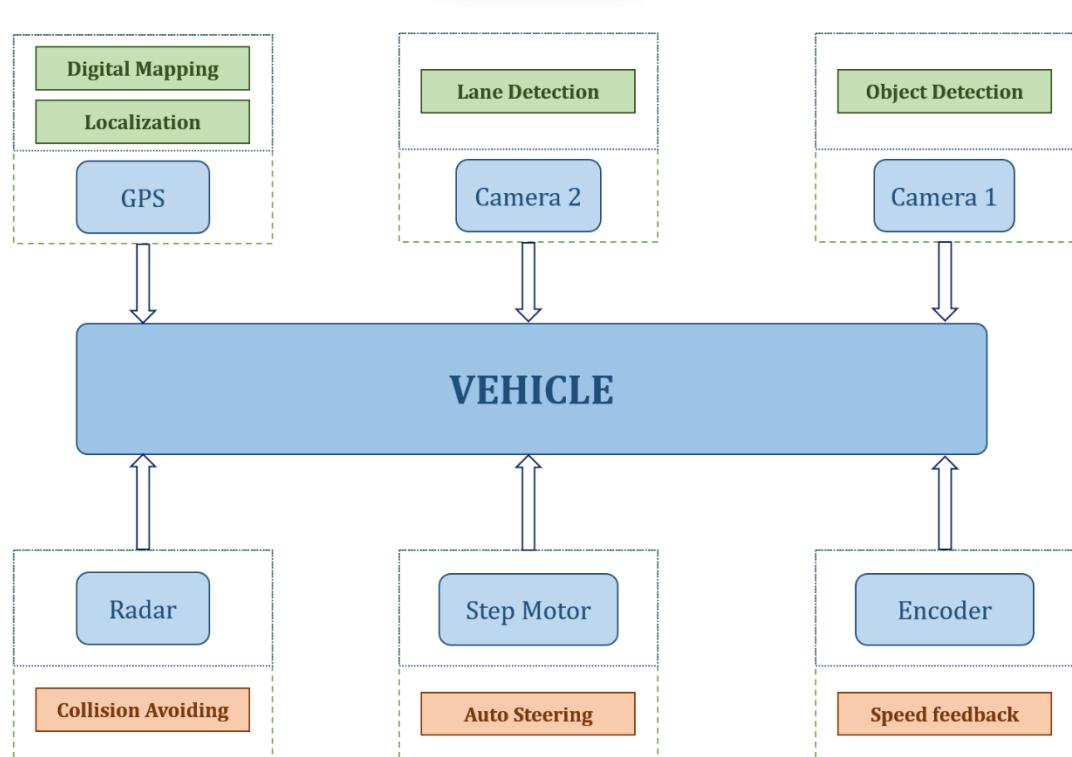


Hình 2-10 Tổng quan hệ thống điều khiển trên xe điện tự hành

Hệ thống điều khiển được xây dựng cho xe điện bao gồm hai phần điều khiển chính và các kết nối cảm biến tích hợp như sơ đồ:

- Thứ nhất, bộ điều khiển trung tâm là Vehicle Control Unit (VCU), đây là một máy tính có tốc độ xử lý tốt đủ khả năng thực hiện nhiều tác vụ cũng như các thuật toán mà những vi điều khiển khó lòng đáp ứng như các thuật toán về xử lý ảnh phát hiện làn đường, hay phát hiện vật thể...

- Phần thứ hai là các board mạch điện tử - Electrical Control Unit (ECU) - có chức năng giao tiếp với bộ độ điều khiển trung tâm, nhận các lệnh điều khiển và điều khiển các thiết bị chấp hành trên xe.Thêm vào đó, các cảm biến được tích hợp thêm vào xe để đọc về các trạng thái hoạt động, những ECU sẽ đảm nhận luôn những công việc này. Trong thực tế, trên xe ô tô thường có rất nhiều những ECU như vậy, mỗi một ECU thường mang một chức năng chuyên biệt, như có ECU chỉ mang nhiệm vụ điều khiển thiết bị chấp hành motor, có ECU chỉ có nhiệm vụ đọc cảm biến, có ECU chỉ điều khiển hệ thống đèn điện trên xe.
- Để cung cấp các dữ liệu đầu vào là các thông số hoạt động của xe hay góc đánh lái, các thông số về môi trường hoạt động của xe như làn đường, vật cản, các cảm biến chuyên dụng được tích hợp vào xe để hỗ trợ thu thập những dữ liệu nêu trên.



Hình 2-11 Hệ thống các cảm biến được tích hợp vào xe điện hỗ trợ quá trình tự hành

Sau quá trình thực hiện tích hợp khả năng tự hành, xe điện hiện đã được thay đổi tích hợp thêm một số module: động cơ bước - hộp số có trả về thông số tốc độ để điều khiển vô lăng, encoder để thực hiện đọc tốc độ của xe, camera để thực hiện phát hiện vật thể, camera thực hiện nhiệm vụ phát hiện làn đường, radar thực hiện phát hiện đối tượng để tránh va chạm, bộ định vị GPS để xác định vị trí của xe. Một tủ điện thực hiện kết nối các module lại với nhau và kết nối với hệ thống điện trên xe.

Chương 3. THIẾT KẾ HỆ THỐNG PHẦN CỨNG

3.1. Tổng quan phần cứng trên xe điện tự lái



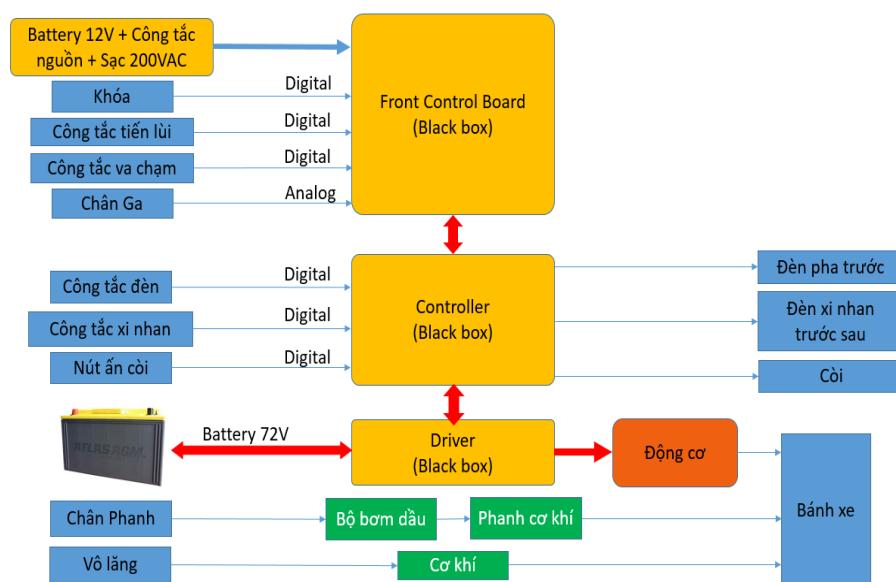
Hình 3-1 Tổng quan kết nối phần cứng trên xe điện tự lái

Mô hình ban đầu của xe điện bao gồm phần khung xe cơ khí có khả năng di chuyển do người dùng điều khiển. Để ứng dụng cho đề tài nghiên cứu, các cảm biến cũng như một số ngoại vi được tích hợp thêm vào mô hình sẵn có để phục vụ cho việc tự lái.

- Động cơ hộp số thay đổi góc vô lăng điều khiển hướng di chuyển xe điện.
- Cảm biến camera thực hiện phát hiện vật thể và phát hiện làn đường giúp bám quỹ đạo di chuyển.
- Cảm biến radar phát hiện đối tượng tránh nguy cơ xảy ra tai nạn va chạm.
- Bộ định vị GPS xác định vị trí hiện tại của xe. Tủ điện điều khiển trung tâm thực hiện kết nối các module lại với nhau và kết nối với hệ thống điện trên xe.

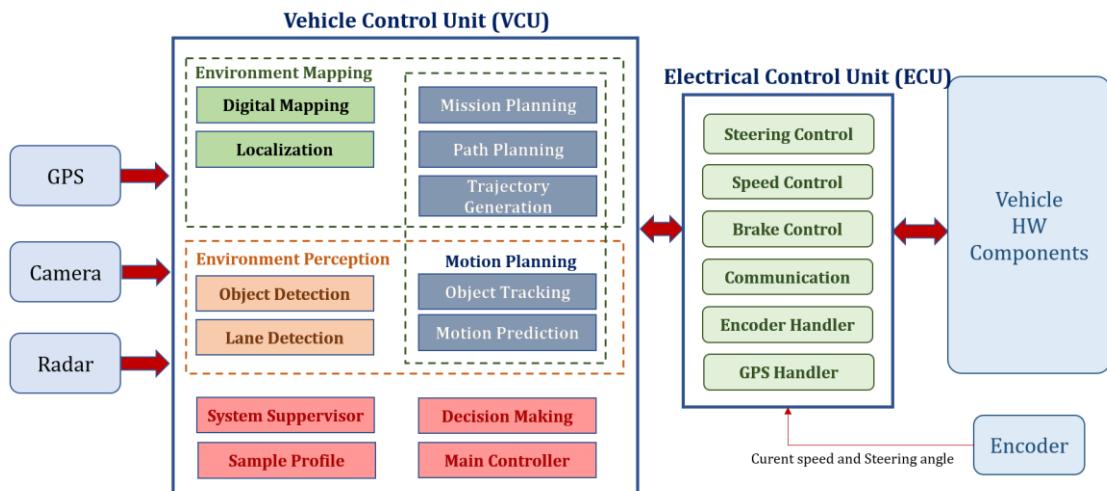
3.2. Tích hợp bộ điều khiển và cảm biến cho xe điện tự lái

Để chuyển phần điều khiển bằng tay bằng điều khiển bằng điện thì các tín hiệu ngõ vào cơ khí cần được thay thế bằng tín hiệu điều khiển mới. Quan trọng bậc nhất là các tín hiệu công tắc tiến lùi, tín hiệu chân phanh, tín hiệu chân ga và góc đánh lái cho xe. Không những thế, để thỏa điều kiện điều khiển tự động, xe cần được trang bị thêm những thiết bị và cảm biến mới. Như vậy một hoặc nhiều những mạch điện tử mới cần được phát triển để đọc các cảm biến và tạo ra các tín hiệu điều khiển nói trên.



Hình 3-2 Hệ thống điều khiển chưa có khả năng tự lái

Sau khi lắp đặt thêm các cảm biến, đi lại dây nối điều khiển, kết hợp với tủ điều khiển xử lý ta được sơ đồ tổng quát:



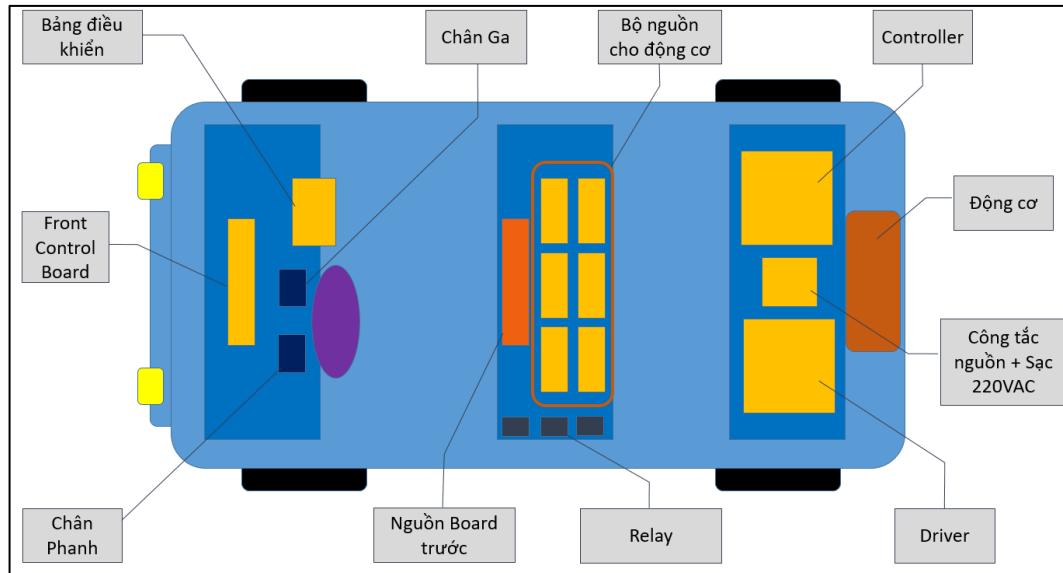
Hình 3-3 Hệ thống điều khiển đã tích hợp khả năng tự lái

- Thông số kỹ thuật xe điện tự lái

Bảng 3-1 Thông số xe điện 4 chỗ

Thông số	Giá trị
Khối lượng	570kg
Khả năng chở vật nặng	960kg
Công suất	3600W
Vận tốc tối đa	~25km/h
Quãng đường đi được trên 1 lần sạc	40 - 50 km
Thời gian sạc	6 – 8 tiếng
Nguồn điện chính	6 ắc quy 12V mắc nối tiếp. 1 ắc quy phụ 12V cho mạch điện
Trợ lái	Có hỗ trợ

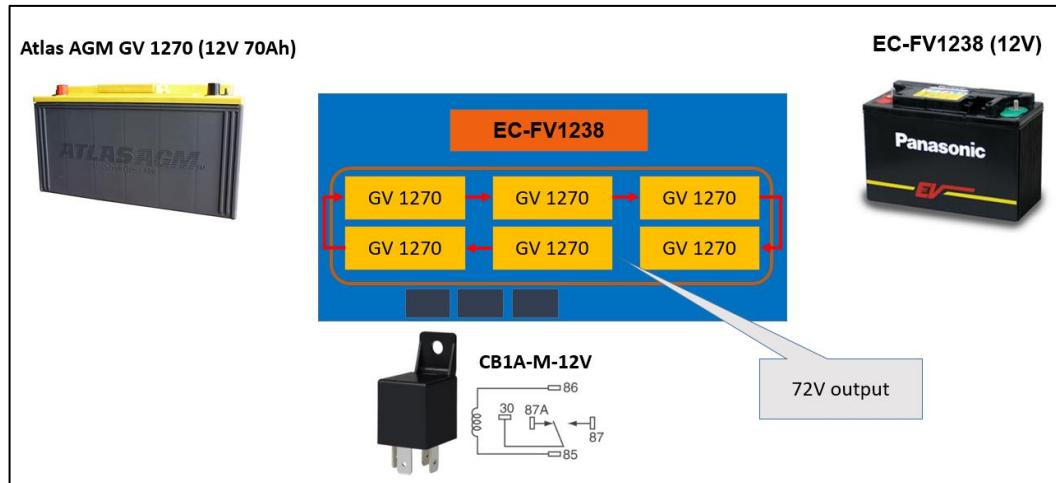
- Tổng quan cơ cấu chấp hành trên xe điện:



Hình 3-4 Tổng quan hệ thống chấp hành trên xe điện 4 chỗ Sanyo

Về tổng quan, hệ thống điện của xe được chia làm 3 phần chính được sắp xếp từ mũi xe đến đuôi xe theo thứ tự sau:

- Phần mạch điều khiển phía trước xe và giao diện người dùng. Mạch điều khiển trong phần này hỗ trợ việc đọc các tín hiệu từ giao diện người dùng trên phần đầu xe. Bảng điều khiển gồm các nút ấn và công tắc điều khiển, phần khóa xe và đèn báo dung lượng pin còn lại.
 - Phần nguồn điện của xe, phần này sẽ được nói rõ hơn ở dưới, về cơ bản nó gồm 1 nguồn điện chính cấp cho động cơ hoạt động và 2 là một nguồn điện phụ để cấp cho các thiết bị điện tử trên xe bao gồm cả phần mạch điều khiển phía trước.
 - Phần Controller và Driver ở đuôi xe là vùng điều khiển hầu hết các thiết bị tiêu thụ công suất lớn như động cơ, các bộ đèn chiếu sáng, còi và xi nhan trước sau. Ngoài ra còn có một mạch điện với công tắc kích hoạt xe à dây cắm sạc. Động cơ chính của xe cũng nằm trong vùng này, kết nối với bộ truyền động để quay 2 bánh xe sau tạo động lực cho xe di chuyển.
- Nguồn điện trên xe (72VDC)



Hình 3-5 Cấu trúc nguồn điện của xe

Phần nguồn điện chính để cấp nguồn cho Driver lái động cơ xe là một bộ nối tiếp 6 ắc quy Atlas AGM GV 1270. Mỗi ắc quy này có điện áp ngõ ra là 12V – 70Ah, cân nặng lên tới 24Kg. Một nguồn điện phụ là một ắc quy EC-FV1238 của hãng Panasonic, với thông số 12V-38Ah, nó là nguồn năng lượng cho các thành phần điện tử khác trên xe để tránh sử dụng chung nguồn với phần công suất.

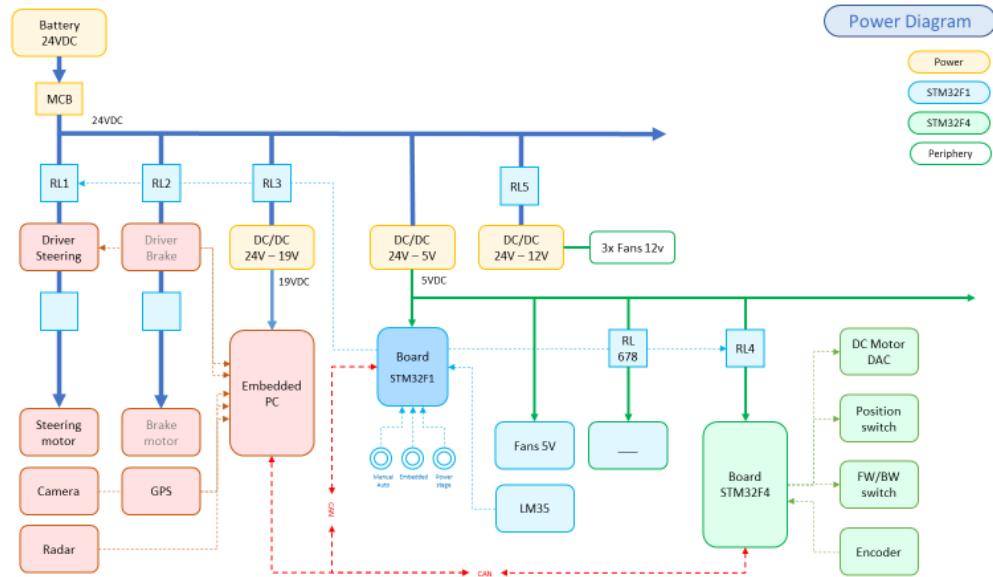
Với thiết kế sẵn của xe sử dụng năng lượng điện, nó đáp ứng được yêu cầu về giảm phát thải, giảm tiếng ồn trong quá trình di chuyển nên xe điện thích hợp với xu thế hiện tại, còn chiếc xe này rất phù hợp với những người dùng mong muốn có 1 chuyến đi trải nghiệm trong các sân golf hay khu vui chơi giải trí. Đối với đề tài này, một chiếc xe với nền tảng chạy bằng điện càng phù hợp hơn cho quá trình nghiên cứu phát triển vì nó dễ dàng tích hợp và kết nối với các phần tử điện tử được phát triển thêm sau này.

3.3. Thiết kế tủ điện điều khiển trung tâm

3.3.1. Sơ đồ kết nối tổng quát tủ điện

Tủ điện điều khiển sẽ đảm nhiệm vai trò là nơi phân phối nguồn cho các thiết bị, cảm biến, xử lý các giải thuật, thu thập dữ liệu, xuất xung cho động cơ,

truyền thông và xử lý các tác vụ khẩn cấp. Từ đó, ta chia tủ điện điều khiển thành các khối, mỗi khối đảm nhiệm các chức năng, nhiệm vụ khác nhau và được giao tiếp, thông tin với nhau qua chuẩn giao tiếp CAN.



Hình 3-6 Sơ đồ kết nối các khối điều khiển bên trong tủ điện

Hệ thống sẽ bao gồm 2 phần nguồn chính:

- Nguồn 72V DC: Phần nguồn điện chính để cấp nguồn cho Driver lái động cơ xe là một bộ nối tiếp 6 ắc quy Atlas AGM GV 1270. Mỗi ắc quy này có điện áp ngõ ra là 12V - 70Ah, cân nặng lên tới 24Kg. Một nguồn điện phụ là một ắc quy EC-FV1238 của hãng Panasonic, với thông số 12V-38Ah, nó là nguồn năng lượng cho các thành phần điện tử khác trên xe để tránh sử dụng chung nguồn với phần công suất.
- Nguồn 24V DC sử dụng cho mạch công suất điều khiển trên xe (sơ đồ phía trên). Thực hiện giảm áp lần lượt xuống 19V, 12V, 5V để cung cấp nguồn chính cho hoạt động của máy tính nhúng xử lý, các ngoại vi và các vi điều khiển. Nguồn có dung lượng 38Ah đảm bảo cho các thiết bị có thể hoạt động liên tục trong suốt thời gian xe vận hành.

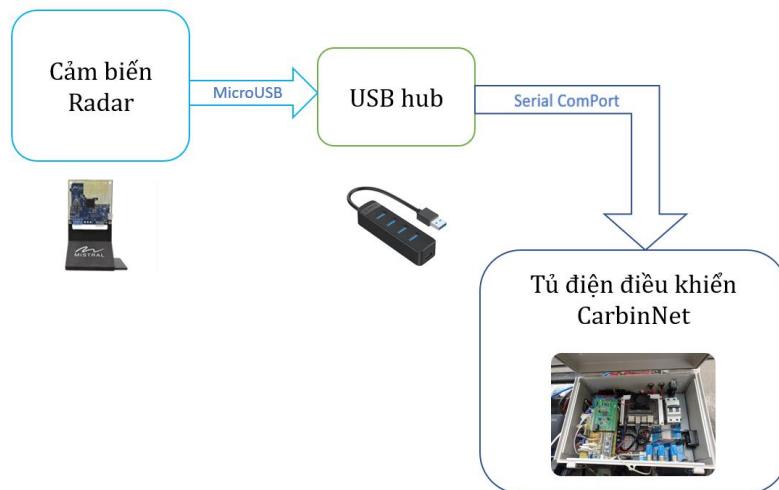
3.3.2. Thiết kế tính năng điều khiển

Khối nguồn sẽ được quản lý, cung cấp đến tất cả thiết bị xử lý trung tâm và điều khiển thông qua giám sát của vi xử lý STM32F1. Nhiệm vụ đọc các tác lệnh điều khiển, ngoại vi như nút nhấn, cảm biến nhiệt độ, nhận lệnh điều khiển từ CANbus. Từ đó xử lý đóng ngắt các relay tương ứng với cung cấp nguồn đến các thiết bị như máy tính nhúng, vi xử lý STM32F4, quạt làm mát, driver,

Máy tính nhúng kết nối với các thiết bị cảm biến radar, camera, GPS, ... thông qua kết nối USB. Ngoài ra, kết nối với hệ thống điều khiển CANbus bằng USBCAN, gửi nhận lệnh điều khiển trên mạng, kiểm tra trạng thái các thiết bị kết nối, nhận dữ liệu cảm biến từ các vi xử lý.

Vi xử lý STM32F4 kết nối với cảm biến encoder, các phần tử ngoại vi trên xe để thực hiện đóng ngắt, chuyển trạng thái các công tắc phục vụ cho mục đích tự động hóa xe điện. STM cũng được kết nối với hệ thống điều khiển CANbus thông qua module hỗ trợ, gửi nhận lệnh điều khiển trên mạng, kiểm tra trạng thái các thiết bị kết nối...

3.4. Kết nối cảm biến Radar



Hình 3-7 Kết nối cảm biến Radar

"mmWaveRadar Mistral 60GHz" của hãng Mistral với ăng-ten trên PCB dựa trên cảm biến FMCW Radar chip đơn có độ phân giải cực cao IWR6843

ES2.0 của Texas Instruments. Chức năng thu thập dữ liệu cho đầu vào thuật toán TTC. Theo dõi các đối tượng chuyển động phía trước xe tự lái. Kết nối với tủ điện thông qua cổng USB và được máy tính nhúng xử lý, điều khiển hoạt động.

Module mmwaveRadar là lý tưởng cho các hệ thống tự hiệu chỉnh, siêu chính xác. Chúng hỗ trợ các giao diện linh hoạt như USB, CAN, UART và SPI và có thể được cấp nguồn qua USB hoặc Header. Ngoài ra, đầu ra dữ liệu thô được cung cấp thông qua các cổng LVDS, cho mục đích gỡ lỗi và phát triển. Cũng có thể bật tính năng thu dữ liệu Raw ADC bằng cách ghép nối với bộ điều hợp thu dữ liệu thời gian thực DCA1000EVM của TI qua đầu nối Samtec 60 chân.



Hình 3-8 mmwaveRadar của hãng Mistral

- Tính năng nổi bật:
 - Mô-đun nhỏ, tối ưu hóa và sẵn sàng sử dụng.
 - Được hỗ trợ bởi USB
 - Hiệu chuẩn và tự kiểm tra tích hợp
 - Đèn flash QSPI và PMIC trên bo mạch
 - Dạng ăng-ten: Ba yếu tố thuôn nhọn
 - Giao tiếp UART qua USB để cấu hình và dữ liệu RADAR

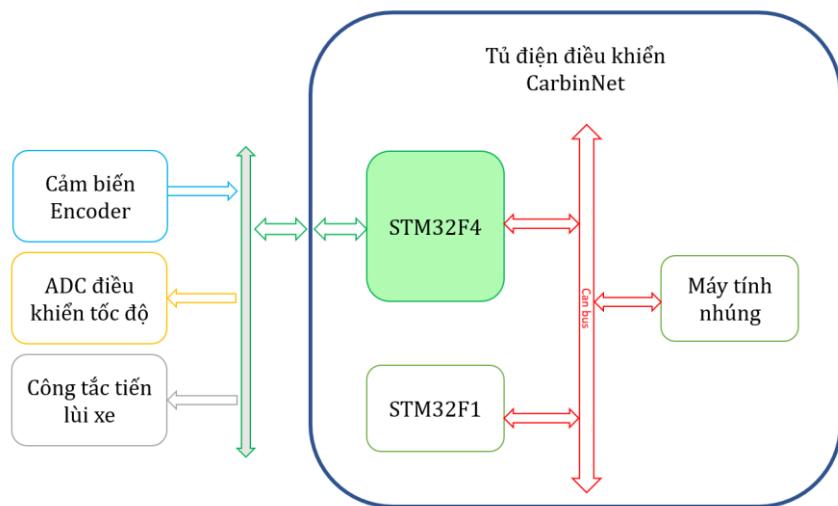
- Kết nối linh hoạt: CAN, UART, SPI, I2C, GPIO, JTAG và LVDS
- Hỗ trợ TI's SDK 3.01
- Hỗ trợ đầu nối tốc độ cao 60 chân cho giao diện điều khiển máy chủ
- Hỗ trợ thu thập dữ liệu thông qua DCA1000 EVM
- Kích thước bo mạch (L x W x H): 54 x 53 x 18 mm
- Nhiệt độ hoạt động công nghiệp: -40⁰C đến 85⁰C

Bảng 3-2 Thông số kỹ thuật cảm biến radar

Thông số kỹ thuật	Mô tả
Nguồn cấp	USB 5V
Range max	Adult – 20m Car – 50m
Azimuth FOV max	150 ⁰
Elevation FOV max	60 ⁰
Chip	IWR6843 Chip Texas Instrument
Memory	1.75MB

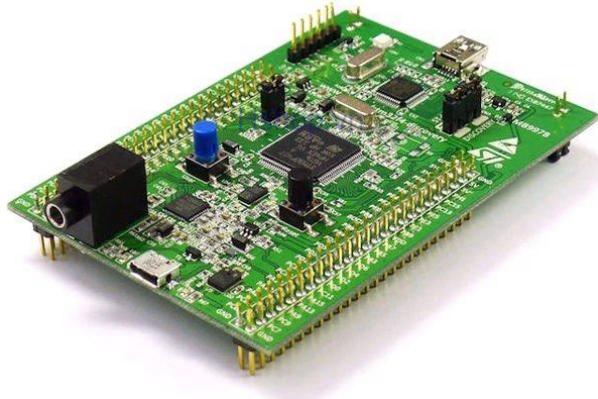
3.5. Kết nối vi xử lý STM32F4

Vi xử lý STM32F4 kết nối với cảm biến encoder, các phần tử chấp hành trên xe để thực hiện đóng ngắt, chuyển trạng thái các công tắc phục vụ cho mục đích tự động hóa xe điện. STM cũng được kết nối với hệ thống điều khiển CANbus giao tiếp được với STM3F1 và máy tính nhúng, gửi nhận lệnh điều khiển trên mạng, kiểm tra trạng thái các thiết bị kết nối...



Hình 3-9 Kết nối vi xử lý STM32F4

Kit STM32F4 DISCOVERY thúc đẩy khả năng của vi điều khiển hiệu suất cao STM32F407 cho phép người dùng dễ dàng phát triển các ứng dụng. Kit bao gồm một bộ công cụ gỡ lỗi nhúng ST-LINK, một giao tốc số ST-MEMS, một micro số, một audio DAC với trình điều khiển loa tích hợp lớp D, đèn LED, nút bấm và cổng micro-AB USB OTG.



Hình 3-10 Board STM32F407 Discovery

Các đặc điểm tính năng chính

- Bộ vi xử lý STM32F407VGT6: 32-bit ARM Cortex - M4 với lõi FPU, bộ nhớ Flash 1MB, RAM 192KB với kiểu chân LQFP100.
- Trình gỡ lỗi ST-LINK/V2 trên STM32F4DISCOVERY.
- USB ST-LINK với khả năng kiểm tra lại và ba giao diện khác nhau:
 - Cổng gỡ lỗi.
 - Cổng COM ảo.
 - Lưu trữ khối lượng.
- Nguồn cung cấp: thông qua USB hoặc nguồn điện bên ngoài 3V – 5V.
- Gia tốc kế 3 trục LIS302DL hoặc LIS3DSH ST MEMS.
- Micro số đa hướng dùng cảm biến âm thanh MP45DT02 ST-MEMS.
- CS43L22 bộ chuyển đổi âm thanh/kỹ thuật số âm thanh công suất thấp (DAC) với trình điều khiển loa tích hợp lớp D.
- Tám đèn LED:
 - LD1 (đỏ/xanh) cho giao tiếp USB.
 - LD2 (đỏ) led báo nguồn 3.3V.
 - Bốn led của người dùng, LD3 (màu cam), LD4 (xanh lục), LD5 (màu đỏ) và LD6 (màu xanh lam).

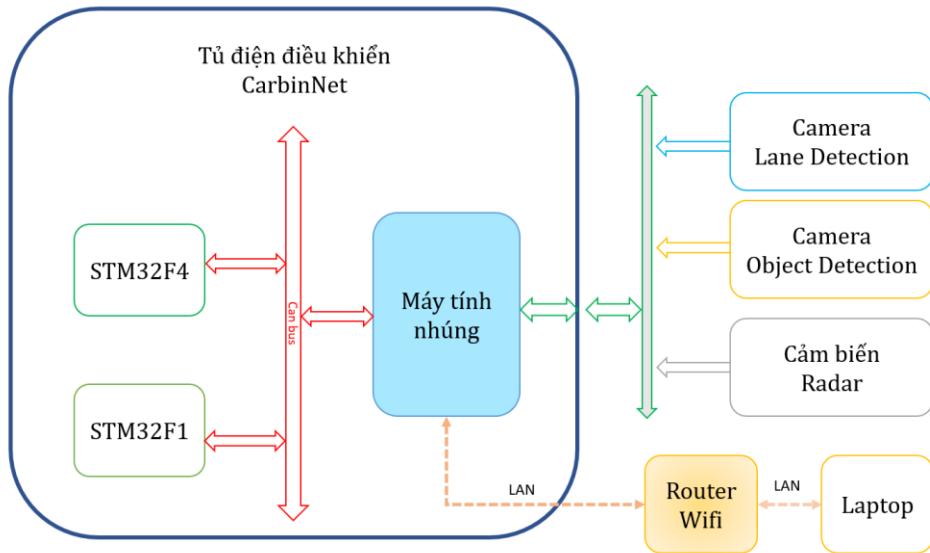
- 2 đèn led OTG USB LD7 (xanh lá cây) VBUS và LD8 (màu đỏ) quá dòng điện.
- Hai nút nhấn (người dùng và reset).
- USB OTG FS có cổng kết nối micro-AB.
- Header mở rộng I/O cho LQPF64 để kết nối nhanh với bo mạch gốc và dễ dàng sử dụng.

Bảng 3-3 Thông số kỹ thuật vi xử lý STM32F407

Thông số kỹ thuật	Mô tả
Nguồn cấp	3V – 5V DC
Bộ nhớ	1MB Flash, 192 KB RAM
Hỗ trợ mạch nạp và debug	ST – LINK/V2
Tần số xung Clock	168MHz

3.6. Máy tính nhúng xử lý trung tâm

Hệ điều hành ROS chạy trên hệ điều hành Ubuntu, do đó Robot cần có một máy tính nhỏ gọn, có thể cài đặt được hệ điều hành này. Thay vì sử dụng laptop cá nhân có cấu hình mạnh nhưng lại có kích thước lớn, chiếm không gian diện tích robot, máy tính nhúng có thể thay thế được máy tính cá nhân vì có kích thước nhỏ gọn, khả năng xử lý đủ nhanh. Jetson Xavier NX là một máy tính có thể đáp ứng mọi yêu cầu đó.



Hình 3-11 Kết nối máy tính nhúng trong hệ thống

Jetson Xavier NX là biến thể mới nhất trong gia đình NVIDIA Jetson, bao gồm Jetson Nano, Jetson AGX Xavier và Jetson TX2, mang hiệu năng Xavier nhưng với dạng thức Nano. Siêu máy tính AI cực nhỏ này có thiết kế module, hướng đến trang bị cho ngành tự động hóa cũng như các thiết bị tính toán nhúng ở biên. Chỉ tiêu thụ 10W năng lượng, Jetson Xavier NX có thể xử lý được những công việc AI hiện đại với hiệu năng ngang tầm Server, lên đến 21 TOPS (Tera Operations Per Second), hay 21 nghìn tỷ phép tính mỗi giây.



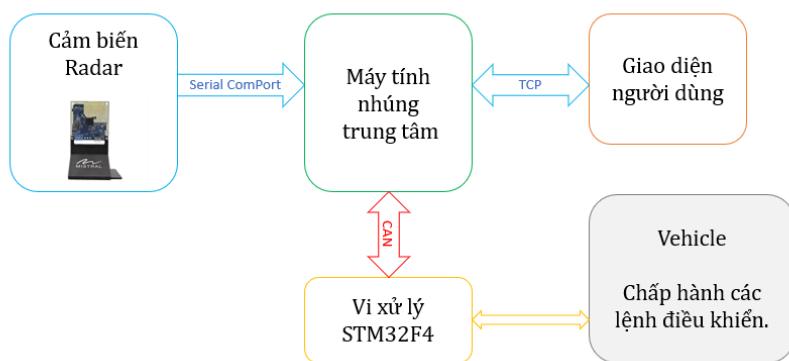
Hình 3-12 Máy tính nhúng Jetson Xavier NX

Bảng 3-4 Thông số kỹ thuật của Jetson Xavier NX:

Thông số kỹ thuật	Mô tả
GPU	512-core Volta GPU with Tensor Cores
CPU	8-core ARM v8.2 64-bit CPU. 8MB L2 + 4MB L3
Memory	16 GB 256-Bit LPDDR4x 137 GB/s
Storage	32GB eMMC 5.1
DL Accelerator	(2x) NVDLA Engines
Vision Accelerator	7-way VLIW Vision Processor
Encoder/Decoder	4Kp60 HEVC/(2x) 4Kp60 12-Bit Support
Size	105 mm x 105 mm

Chương 4. THIẾT KẾ HỆ THỐNG PHẦN MỀM

4.1. Tổng quan kết nối hệ thống phần mềm



Hình 4-1 Tổng quan kết nối hệ thống cho thuật toán TTC

Triển khai thuật toán TTC cho hệ thống theo sơ đồ kết nối như trên. Cảm biến radar hoạt động truyền dữ liệu theo các chuỗi ký tự, số. Dữ liệu sẽ được xử lý chính tại máy tính nhúng trung tâm. Kết quả đạt được sẽ được gửi theo hai hướng: một là gửi đến giao diện người dùng (GUI điều khiển) thông qua giao thức TCP các thông tin mà cảm biến radar xử lý, kết quả thời gian va chạm TTC, kết quả điều khiển... Hai là kết quả sẽ được gửi lên CANbus để vi xử lý STM32F4 nhận và xử lý gói dữ liệu. Kết quả đạt được là tốc độ và góc bánh lái, điều khiển xe chuyển động.

Robot Operating System – ROS là hệ điều hành mã nguồn mở dành cho robot, là một framework được dùng rất rộng rãi trong lĩnh vực robotic với nhiều ưu điểm. Nó tạo ra một nền tảng phần mềm có thể hoạt động trên rất nhiều robot khác nhau mà không cần sự thay đổi quá nhiều trong chương trình phần mềm. ROS cung cấp môi trường phát triển giúp các phần mềm kết nối với nhau một cách dễ dàng, nó giúp cho việc sử dụng lại và tích hợp nhiều mã nguồn mở trở nên đơn giản hơn nhiều



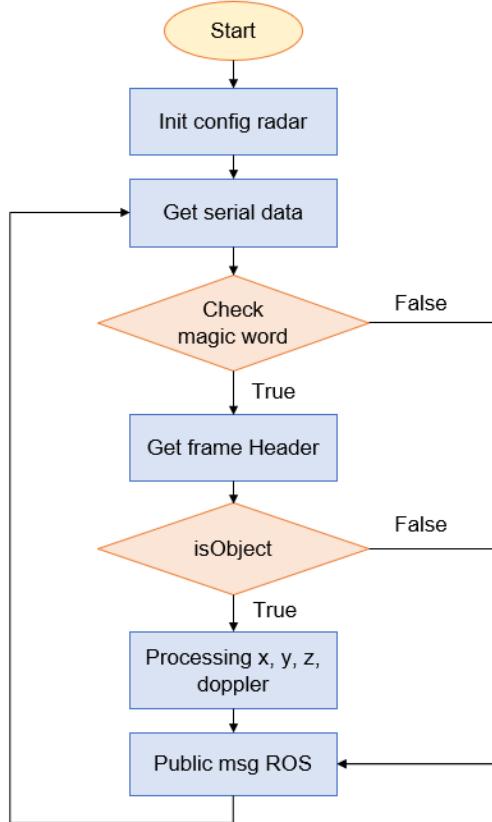
Hình 4-2 Robot Operating System

Kết quả xử lý cũng như hệ thống tránh va chạm ứng dụng được thực hiện trên máy tính nhúng Jetson được cài đặt hệ điều hành Ubuntu 18.04.

4.2. Xử lý dữ liệu cảm biến Radar

Cảm biến radar mmwave IWR6843 sử dụng bộ firmware “Parking Garage Occupancy” cung cấp bởi TI mmWave SDK. Kết nối với phần cứng cảm biến thông qua giao thức serial. Các tính năng chính trong firmware này là thuật toán phát hiện tĩnh và theo dõi nhóm đối tượng:

- Thuật toán phát hiện tĩnh cho phép xác định phản xạ từ các phương tiện đứng yên. Sử dụng thuật toán phát hiện tĩnh cho phép phát hiện các vật thể tĩnh là ô tô, xe tải và các vật thể lớn khác trong khi bỏ qua các vật thể tĩnh vĩnh viễn bao gồm cảnh như cột điện, cột và tường, ...
- Với tính năng theo dõi nhóm, cảm biến có thể đo quỹ đạo và tốc độ của một người, ô tô đang di chuyển, cho phép hệ thống phát hiện vật thể đang di chuyển ra xa hay tiến lại gần.



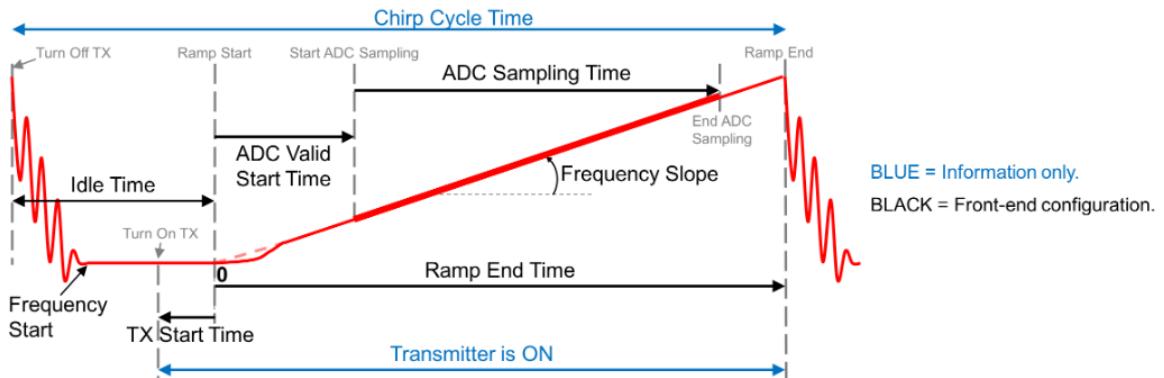
Hình 4-3 Lưu đồ thực hiện xử lý dữ liệu trả về từ radar

Chương trình sẽ được thực hiện trong môi trường ROS. Khi bắt đầu kết nối với cảm biến, lệnh cấu hình (*init config*) sẽ được máy tính nhúng gửi xuống radar thông qua cổng serial. Khi hoàn tất quá trình config, cảm biến sẽ bắt đầu thực hiện gửi lên các gói dữ liệu theo chu kỳ mà đã được cấu hình trước đó. Các gói dữ liệu đều được bắt đầu bởi 1 chuỗi ký tự đặc biệt gọi là “Magic word”. Khối “*Check magic word*” kiểm tra nếu có chuỗi ký tự thì sẽ tiếp tục quá trình xử lý tiếp theo. Tính toán vị trí, vận tốc, hướng di chuyển, cảnh báo... rồi cuối cùng sẽ đóng gói thành một tin nhắn (message) gửi lên ROS-master.

4.2.1. Cấu hình ban đầu cho cảm biến

Cấu hình này cho phép ta chọn các thông số để cấu hình thiết bị mmWave. Theo nguyên tắc, các trường cấu hình sẽ được đặt đầu tiên, sau đó là các tham số phụ thuộc. Ta cần lưu ý rằng một số tham số cấu hình sẽ ảnh hưởng đến giá trị của các trường phụ thuộc khác, vì thế ta có thể phải điều

chỉnh lại các tham số của các trường phụ thuộc để có được lựa chọn mong muốn.



Hình 4-4 Sơ đồ hoạt động của một chu kỳ tín hiệu quét (chirp) radar

Từ sơ đồ trên, ta sẽ thực hiện các cấu hình phù hợp tương ứng với các trường trong bảng dưới.

```

CQSigImgMonitor 0 115 6
analogMonitor 0 0

aoaFovCfg -1 -60 60 0 20
cfarFovCfg -1 0 0 50
cfarFovCfg -1 1 -20 20

% *****TRACKING COMMANDS*****
staticBoundaryBox -1 1 0 1 0 1
boundaryBox -3 1 0 50 0 1.5
gatingParam 50 6 6 6 20
stateParam 3 1 1 1 1
allocationParam 100 50 0.05 5 20 20
maxAcceleration 8 8 8
trackingCfg 1 2 500 25 200 50 100 90

heatmapGenCfg -1 1 0 40 130 60.0 3.0 10
staticDetectionCfg -1 0 -50 50 0 1 0.7 6 0.2 4 20

% *****SENSOR START*****
sensorStart

```

Những trường cần chú ý khi cấu hình hoạt động cho cảm biến:

- *profileCfg*: Trường này bao gồm cấu hình các thông số như tần số hoạt động, thời điểm phát tín hiệu Tx, gian lấy mẫu, số lượng mẫu ADC trong chu kỳ, ... thông số này ảnh hưởng trực tiếp đến kết quả mong muốn và hoạt động bình thường của cảm biến.
- *frameCfg*: Cấu hình *chu kỳ gửi khung dữ liệu* gửi qua serial tính bằng millisecond.
- *aoaFovCfgL*: Thông số cấu hình *độ rộng góc quét* cho ăng-ten TX theo phương ngang và dọc. Góc càng lớn, không gian quan sát được càng rộng và số điểm dữ liệu (point cloud) càng nhiều. Ngoài ra, trường này còn dùng để *giới hạn vận tốc* tối thiểu/tối đa của các vật thể chuyển động, giúp loại bỏ những di chuyển không cần thiết.
- *boundaryBox*: Giới hạn khoảng không *địa lý* cho việc theo dõi các đối tượng. Việc mở rộng góc quét giúp tăng không gian quan sát nhưng vô

tình làm cho dư thừa các vật thể không cần thiết như cây cối, nhà cửa xung quanh. Do đó, ta có thể vẫn nhận biết được môi trường xung quanh nhờ góc quét lớn nhưng vẫn giới hạn được phạm vi cho phép để theo dõi các đối tượng mà không cần thu hẹp góc quét của cảm biến. Các thông số giới hạn bao gồm độ rộng x, chiều sâu y, và độ cao z. (có thể âm hoặc dương).

- *gatingParam*: trong khoảng quan sát của cảm biến có thể xuất hiện cùng lúc nhiều đối tượng theo dõi. Thông số này sẽ giới hạn về *diện tích* của từng vật thể trong khoảng quan sát. Giúp tách biệt được đối tượng ở gần nhau.
- *allocationParam*: ở mục [4.3] ta có đề cập đến thuật toán theo dõi đối tượng gtrack, thông số này giúp cấu hình những yêu cầu của thuật toán như vận tốc tối đa/tối thiểu của các pointCloud, giá trị ngưỡng SNR và số điểm tối thiểu (number of points) để trở thành một đối tượng mới.

4.2.2. Xử lý dữ liệu nhận về từ serial

Các kết quả xử lý từ cảm biến radar chứa thông tin theo dõi của các đối tượng sẽ được đóng gói thành từng gói dữ liệu và gửi thông qua UART. Mỗi gói tin bao gồm một tiêu đề (header) như được hiển thị bên dưới và một số mục TLV có thể thay đổi có chứa các thông tin khác nhau. Mỗi mục TLV bao gồm thông tin về loại (type), chiều dài(length) và giá trị dữ liệu (payload). Vì độ dài của gói dữ liệu phụ thuộc vào số lượng pointCloud và đối tượng được theo dõi nên nó có thể thay đổi theo từng khung. Phần cuối của gói dữ liệu gửi được thêm tự động để tổng chiều dài gói luôn là bội số của 32 Byte.

The screenshot shows a Notepad window with the title 'as_demo_uart_stream.txt - Notepad'. The menu bar includes File, Edit, Format, View, and Help. The main content area displays a large amount of binary data in hex format. The data starts with '02 01 04 03 06 05 08 07 04 00 05 03 40 00 00 00 00 43 68 0a 00 01 00 00 00 ea d9 6e a7 00 00 00 00 00 00 02' and continues for several pages. At the bottom of the Notepad window, there are status bars showing 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

Hình 4-5 Dữ liệu nhận được từ một gói tin radar

- **Header:** Tiêu đề khung được gửi ở đầu mỗi gói dữ liệu. Sử dụng “Magic Word” để tìm phần đầu của mỗi gói. Magic Word = “2 1 4 3 6 5 8 7”

Thông tin nhận được từ header bao gồm chi tiết của các trường sau:

Bảng 4-1 Cấu trúc frameheader

Value	Type	Bytes
Magic Word	uint16_t	8
Version	uint32_t	4
Total Packet Length	uint32_t	4
Platform	uint32_t	4
Frame Number	uint32_t	4
Time [in CPU Cycles]	uint32_t	4
Num Detected Obj	uint32_t	4
Num TLVs	uint32_t	4
Subframe Number	uint32_t	4
Num Static Detected Obj	uint32_t	4

- TLV Header: Đối với mỗi TLV sẽ bao gồm thông tin về type và length tương ứng bên dưới:
 - Type 1: Detected Object (Dynamic Points); Length: 16 Bytes x Num Detected Obj. Bao gồm 16-byte đại diện cho vị trí và vận tốc của từng điểm trong pointCloud. Lưu ý: vị trí của mỗi điểm được cho trong hệ tọa độ cầu.

Bảng 4-2 Cấu trúc frameTLV - type 1

Value	Type	Bytes
range [m]	float	4
angle [rad]	float	4
elev [rad]	float	4
doppler [m/s]	float	4

- Type 7: Detected Object (Dynamic Points) Side Info; Length: 4 Bytes x Num Detected Obj. Bao gồm 4-byte mang giá trị cho SNR và nhiễu của từng điểm trong pointCloud.

Bảng 4-3 Cấu trúc frameTLV - type 7

Value	Type	Bytes
snr	uint16_t	2
noise	uint16_t	2

- Type 10: Tracked Object List; Length: 40 Bytes x Num Tracked Objects. Bao gồm 40-byte cho từng đối tượng được theo dõi, cung cấp các thành phần ID, vị trí, vận tốc và gia tốc của mỗi mục tiêu.

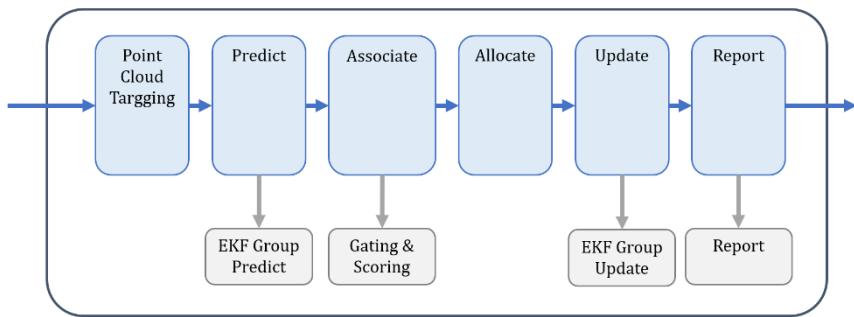
Bảng 4-4 Cấu trúc frameTLV type - 10

Value	Type	Bytes
Target ID	uint32_t	8
pos X [m]	float	4
pos Y [m]	float	4
vel X [m/s]	float	4
vel Y [m/s]	float	4
acc X [m/s^2]	float	4
acc Y [m/s^2]	float	4
pos Z [m]	float	4
vel Z [m/s]	float	4
acc Z [m/s^2]	float	4

Dữ liệu đầu vào quan trọng cho bộ điều khiển ứng dụng thuật toán TTC là “Type 10: Tracked Object List”. Một số type khác không được sử dụng trong phiên bản phần mềm này.

4.3. Thuật toán theo dõi đối tượng – Đầu vào dữ liệu thuật toán TTC

Thuật toán theo dõi đối tượng Group Tracking (*gtrack*) được triển khai như một thư viện trong DPU – cấu trúc nền tảng để phát triển phần mềm xử lý radar được cung cấp bởi SDK. DPU được cấu hình với các thông số mô tả cảm biến, giới hạn môi trường xung quanh và chuyển động của các đối tượng. Hình 4-6 cho thấy các bước thuật toán thực hiện trong mỗi chu kỳ dữ liệu. Thuật toán có ngõ vào dữ liệu theo tọa độ cực (Khoảng cách, phương vị, độ cao, Doppler, SNR) và theo dõi các đối tượng trong không gian Descartes. Do đó, hãy sử dụng quy trình bộ lọc Kalman mở rộng (EKF)



Hình 4-6 Thuật toán theo dõi Group Tracking

Đầu vào của thuật toán *gtrack* là một “đám mây điểm” - tập hợp những điểm dữ liệu trong không gian 3D, những điểm này sẽ được loại bỏ hoặc giữ lại dựa trên cấu hình giới hạn môi trường xung quanh. Nếu được phân loại là nằm ngoài ranh giới thì sẽ bị bỏ qua trong các quy trình “*association*” và “*allocation*”.

Mỗi lần lặp lại của *gtrack* sẽ chạy qua các bước sau:

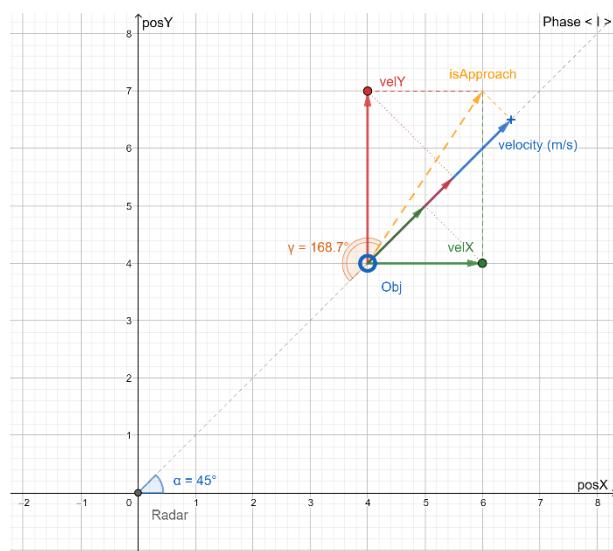
- Nếu tồn tại đối tượng theo dõi, sử dụng bộ lọc Kalman để dự đoán trung tâm (centroid) của nhóm đối tượng theo dõi tại thời điểm n dựa trên ma trận hiệp phương sai trạng thái được ước tính tại thời điểm n-1. Tính toán trạng thái tiên nghiệm và ước tính hiệp phương sai cho từng đối tượng có thể theo dõi.
- Nếu tồn tại đối tượng theo dõi, quy trình kết nối “*association*” cho biết mỗi đối tượng theo dõi liệu mỗi điểm trong “đám mây điểm” đó có đủ gần (gating) với điểm tâm (centroid) hay không và nếu có, điểm sẽ được gán giá trị (scoring) cao nhất.
- Bất kỳ điểm nào không được chỉ định là tâm (centroid) sẽ chuyển sang quy trình phân bổ “*allocation*”. Trong quá trình này, đầu tiên các điểm được nối thành một tập hợp dựa trên sự gần nhau của chúng trong các tọa độ Descartes + Doppler 3D. Mỗi tập hợp phải vượt qua ngưỡng (threshold) SNR tích lũy và số điểm tối thiểu (number of points) để trở

thành một đối tượng mới (Object). Nếu đáp ứng được những điều kiện đó, một đối tượng mới được phân bổ thành công

- Trong bước cập nhật, các đối tượng được cập nhật dựa trên tập hợp các điểm liên quan. Tính toán sự thay đổi, độ lợi Kalman, vectơ trạng thái tiếp theo và hiệp phương sai. Chuẩn bị cho chu kỳ tiếp theo của *gtrack*.
- Cuối cùng báo cáo từng đơn vị theo dõi và tạo ra kết quả thuật toán.

4.4. Tính toán thời gian va chạm TTC, vận tốc, vị trí của các đối tượng

Thực hiện khảo sát một đối tượng chuyển động như hình sau:



Hình 4-7 Khảo sát chuyển động của đối tượng

Dữ liệu từ nhận được từ cảm biến radar cho ta được những giá trị về vị trí và vận tốc của đối tượng. posX, posY cho biết vị trí của đối tượng trên hệ tọa độ descartes. velX, velY cho biết độ lớn và hướng di chuyển của đối tượng. Thể hiện các thông số lên trên đồ thị, chọn hướng dương như các trục tọa độ, ta được các vector:

- velX – màu xanh lá cây
- velY – màu đỏ
- Obj – vòng tròn xanh lớn tại vị trí posX, posY

- *isApproach* – màu cam; được tính bằng tổng vector velX và velY trên hệ tọa độ thể hiện cho hướng di chuyển của đối tượng. Xét góc gamma (γ) như Hình 4-7, độ lớn γ quyết định sự ảnh hưởng của đối tượng đối với xe trong trường hợp $-90^\circ < \gamma < 90^\circ$.

Giả sử xe điện AEV có chiều rộng là 3m, ta lắp đặt cảm biến radar ở chính giữa xe phía trước. Gọi $posX_{Approach}$ là giới hạn chiều rộng của xe tính từ radar. Dựa vào điều kiện của góc γ , theo công thức ta được:

$$posX_{Approach} = posX - \left(\frac{velX \cdot posY}{veclY} \right) \quad (24)$$

Nếu $-1.5m < posX_{Approach} < 1.5m$ thì ta kết luận đối tượng sẽ có nguy cơ va chạm với xe trong tương lai.

- *velocity* – màu xanh dương; được tính bằng tổng vector của hai hình chiếu velX và velY lên đường thẳng đi qua gốc tọa độ (vị trí cảm biến radar) đến đối tượng theo dõi. Ý nghĩa của vector velocity thể hiện vận tốc hướng tâm của đối tượng. Nếu độ lớn mang giá trị dương ta nói rằng đối tượng đang tiến xa cảm biến. Ngược lại, đang có xu hướng tiến lại gần cảm biến.

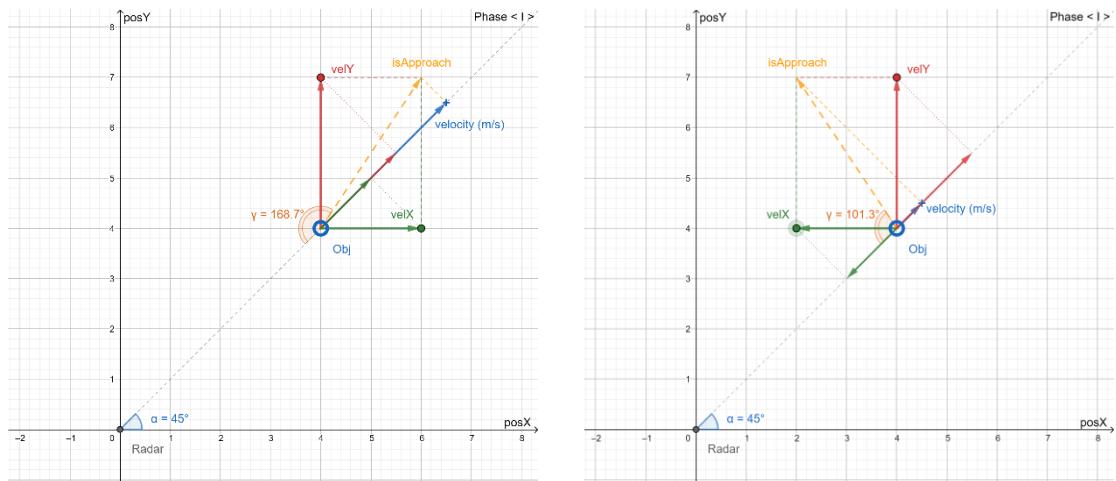
$$\alpha = \cos^{-1} \frac{posX}{\sqrt{posX^2 + posY^2}} \quad (25)$$

$$velocity = vecx + vecy = velX \cos \alpha + velY \cos(90^\circ - \alpha)$$

Tương tự cho tất cả các vị trí trên hệ tọa độ. Ở đây ta chỉ xét góc phần tư thứ I và thứ II vì giới hạn góc quét của cảm biến radar:

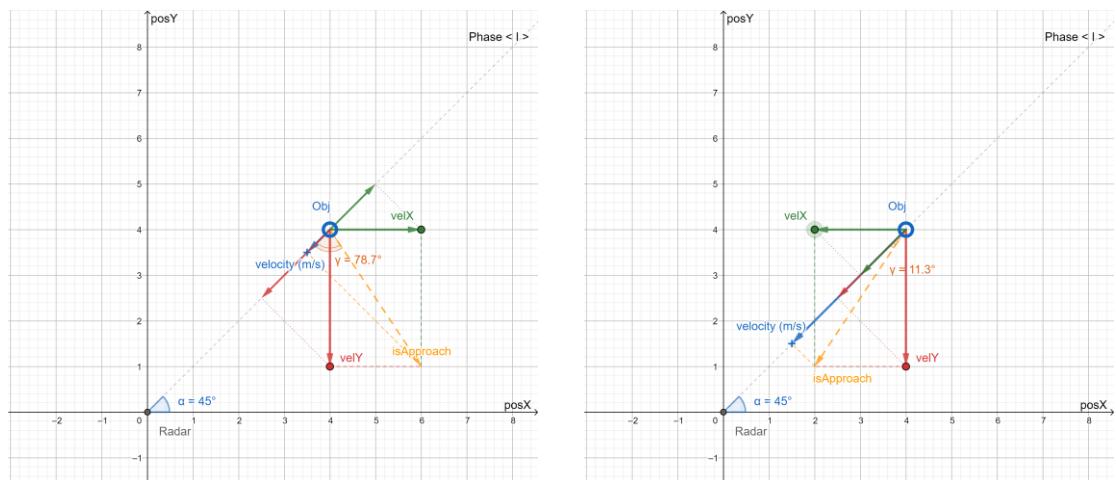
❖ Góc phần tư thứ I

- $velY > 0$: đối tượng có xu hướng tiến ra xa cảm biến.



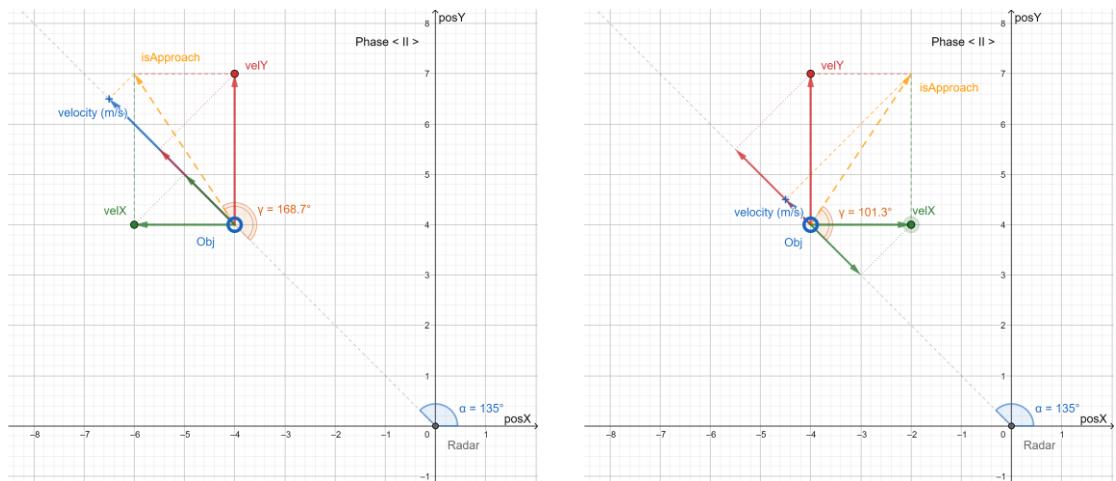
Hình 4-8 Khảo sát chuyển động của đối tượng khi $velY > 0$

- $velY < 0$: độ lớn góc gamma (γ) thể hiện nguy cơ va chạm. (Hình thứ 1 – không có nguy cơ va chạm; Hình thứ 2 – có nguy cơ va chạm)



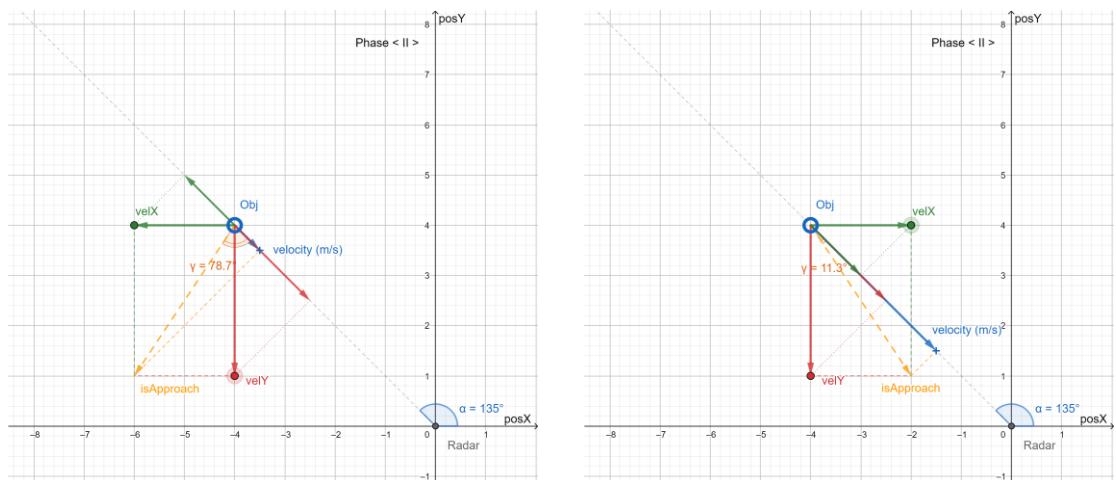
Hình 4-9 Khảo sát chuyển động của đối tượng khi $velY < 0$

- ❖ Góc phần tư thứ II
 - $velY > 0$: đối tượng có xu hướng *tiến ra xa* cảm biến.



Hình 4-10 Khảo sát chuyển động của đối tượng khi $velY > 0$

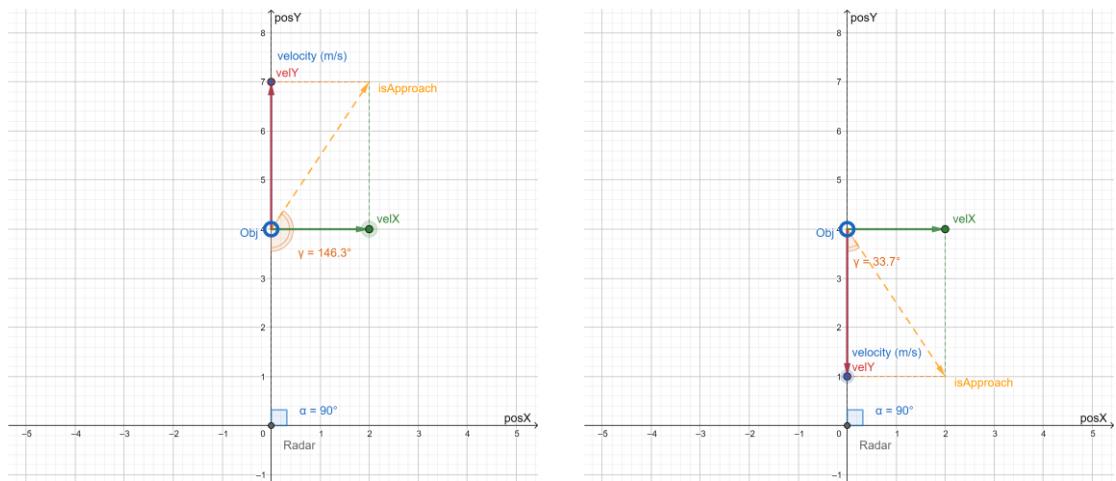
- **$velY < 0$:** độ lớn góc *gamma* (γ) thể hiện nguy cơ va chạm. (Hình thứ 1 – không có nguy cơ va chạm; Hình thứ 2 – có nguy cơ va chạm)



Hình 4-11 Khảo sát chuyển động của đối tượng khi $velY < 0$

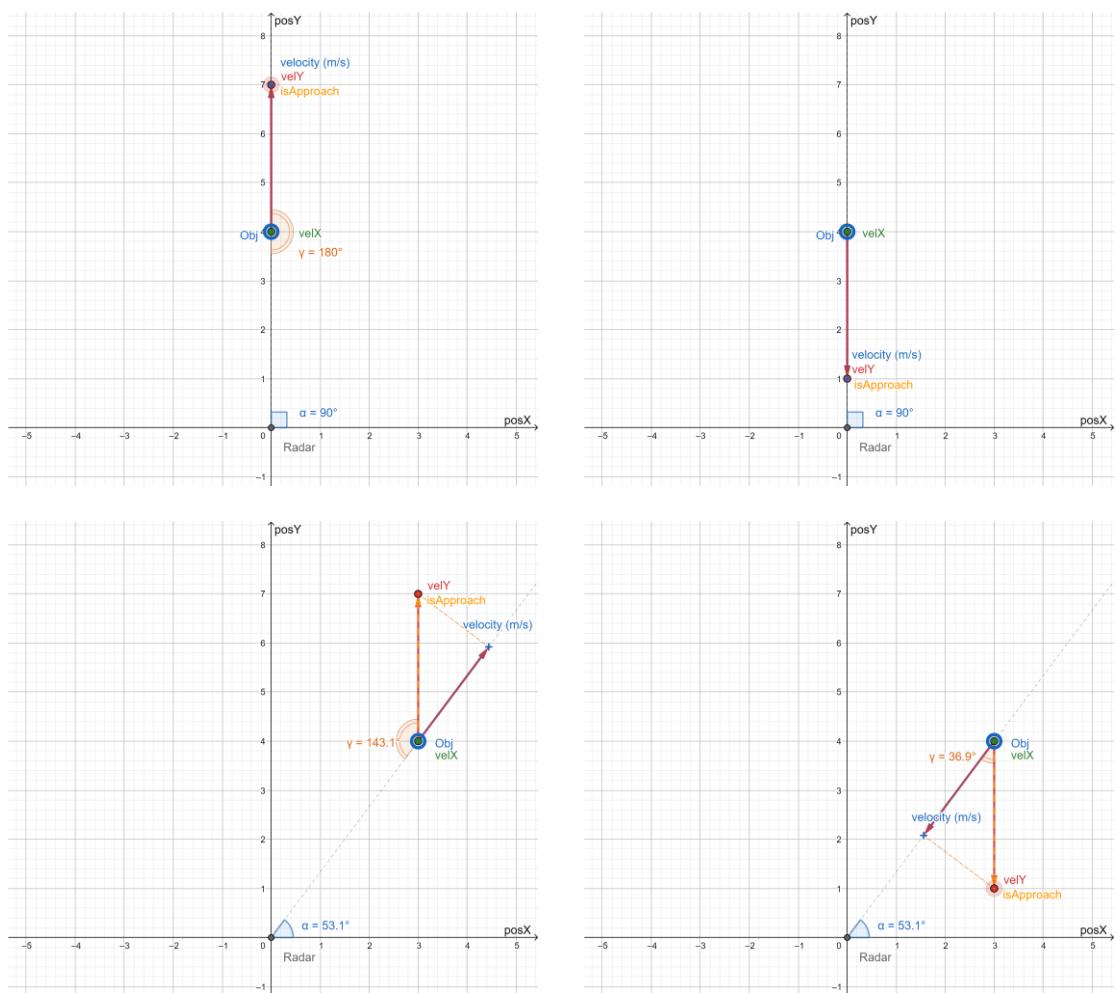
❖ Góc chính diện

- **$VelX > 0$:**



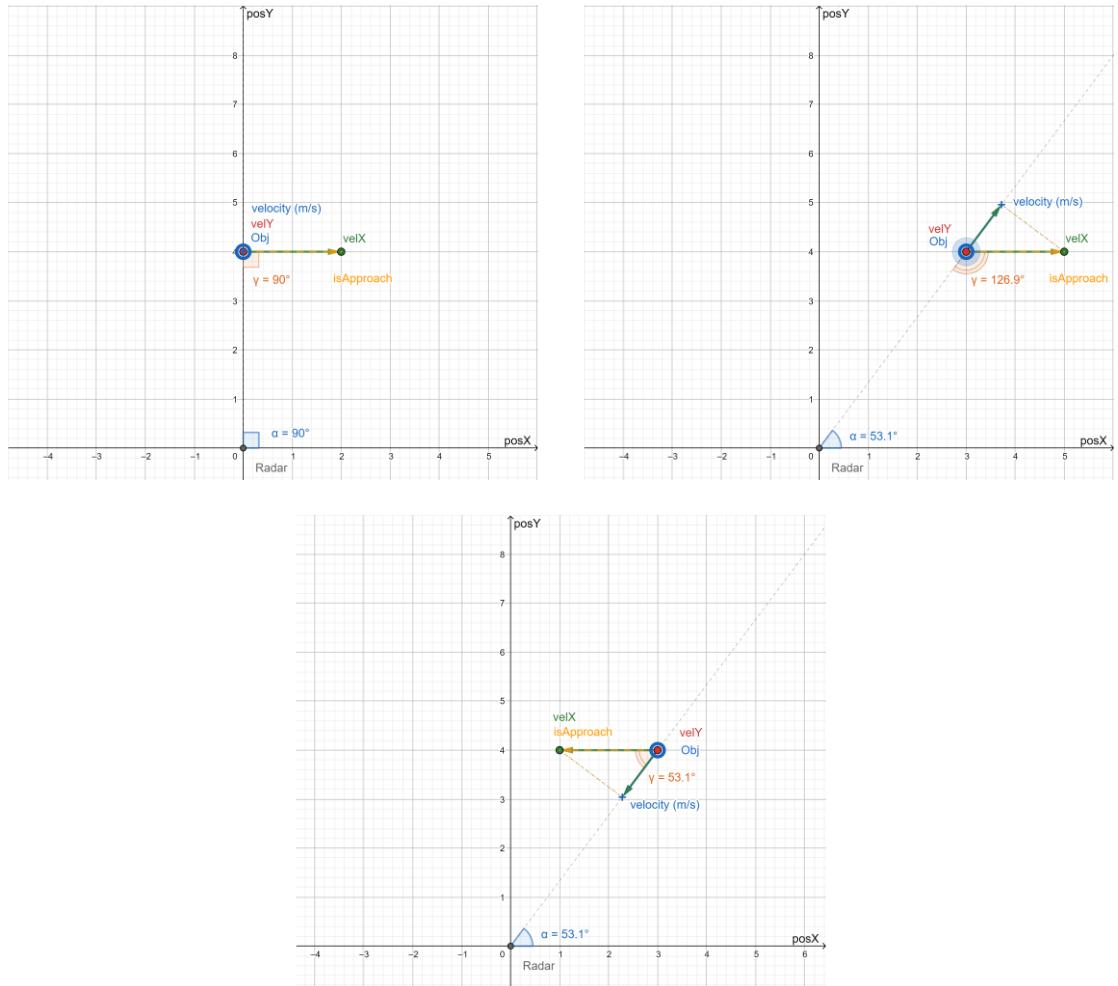
Hình 4-12 Khảo sát chuyển động của đối tượng khi velY trùng phương chuyển động

- $VelX = 0$:



Hình 4-13 Khảo sát chuyển động của đối tượng khi $VelX = 0$

- $VelY = 0$: không có nguy cơ xảy ra va chạm

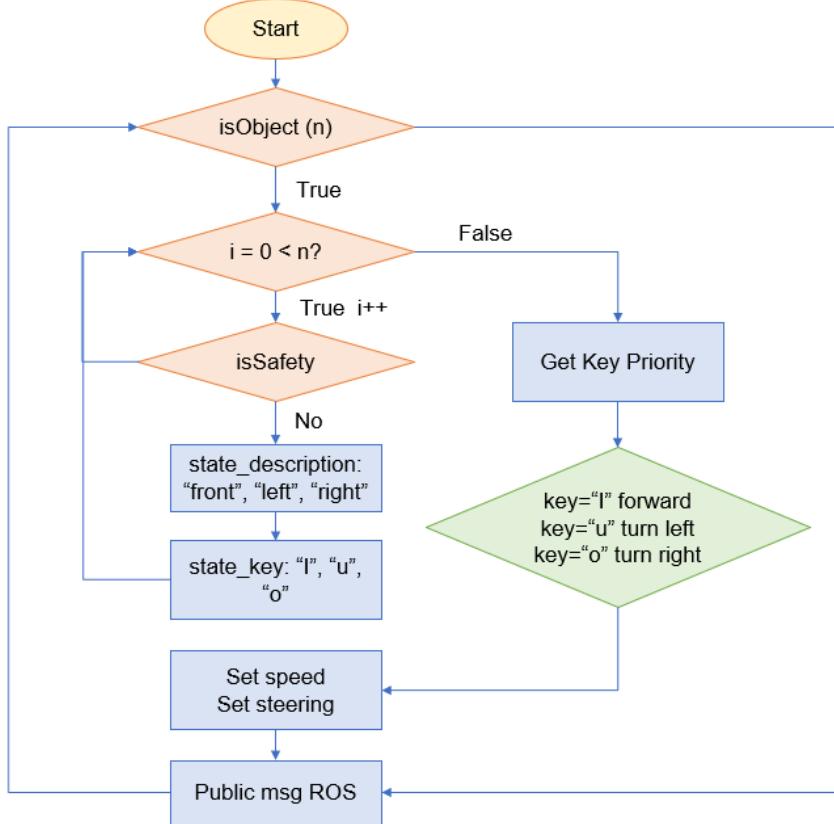


Hình 4-14 Khảo sát chuyển động của đối tượng khi $VelY = 0$

4.5. Xây dựng bộ điều khiển tránh va chạm từ thuật toán TTC

Sau khi xử lý dữ liệu thu thập được từ cảm biến, kết quả ấy được đưa vào quá trình điều khiển chuyển động của xe điện tự lái. Ngõ vào cho hệ thống điều khiển chính là các đối tượng chuyển động đang được cảm biến theo dõi. Kết quả mong muốn là tốc độ di chuyển và hướng chuyển động có thể né tránh va chạm với những đối tượng. Giúp giảm tối đa nguy cơ xảy ra các vụ tai nạn. Bên cạnh đó, hệ thống sẽ phát những cảnh báo nguy hiểm, trạng thái hoạt động của xe... góp phần nâng cao độ tin cậy và chất lượng điều khiển.

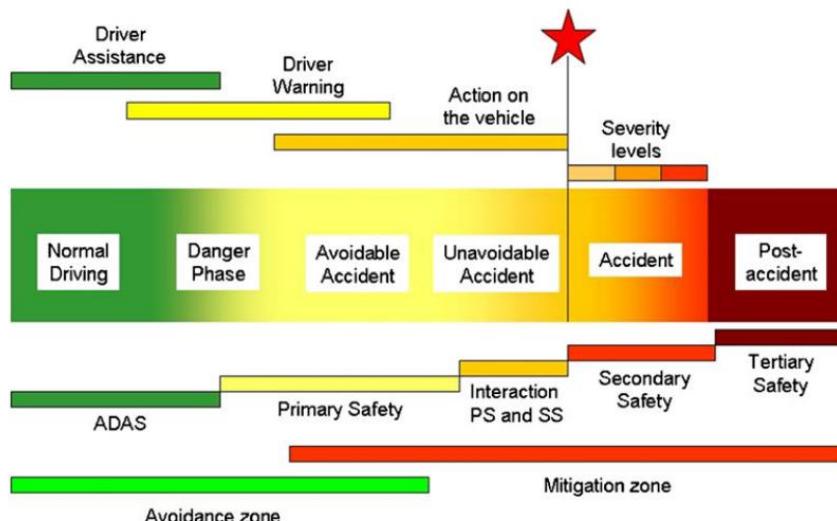
Sơ đồ giải thuật hoạt động đơn giản của hệ thống tránh va chạm trình bày như hình bên dưới:



Hình 4-15 Giải thuật hoạt động của bộ điều khiển

Trong giới hạn quan sát của cảm biến, có thể cùng lúc phải theo dõi nhiều đối tượng trong giới hạn. Mỗi một đối tượng sẽ đặc trưng riêng bởi vị trí tọa độ trong không gian, vận tốc và mức độ nguy hiểm mà đã được đánh giá ở mục [4.3.3]. Bắt đầu chu trình, từng đối tượng sẽ được kiểm tra, đánh giá về mức độ nghiêm trọng dựa vào hai thông số quan trọng đó là độ lớn của góc gamma γ và giá trị thời gian xảy ra va chạm TTC. Tại đây chia làm 4 cấp độ đánh giá nguy hiểm theo thứ tự giảm dần “carAccident”, “carWarning”, “carSafety”, “carNormal” dựa trên mô hình an toàn (Ingeniería del Transporte) [11] (Hình 4-16).

- “carAccident”, “carWarning” là nhóm những đối tượng có khả năng gây ra vụ va chạm với xe tự hành. Chỉ những đối tượng thuộc nhóm nguy hiểm mới tiếp tục các quá trình tiếp theo của bộ điều khiển. $TTC < 1.5s$ đối tượng được đánh giá “carAccident”; nếu thời gian va chạm trong khoảng $1.5s < TTC < 10s$ sẽ phân thành “carWarning”.
- “carSafety”, “carNormal” được đánh giá an toàn vì có những đặc trưng không thỏa các điều kiện phân loại: góc va chạm γ có giá trị lớn hoặc thời gian va chạm TTC đạt ở mức cao hơn ngưỡng nguy hiểm. Những đối tượng này sẽ được bộ trách nhiệm bù trừ qua trong quá trình xử lý. Thời gian TTC trong khoảng $10s < TTC < 30s$ được đánh giá là “carSafety” khi TTC lớn hơn ngưỡng đó, đối tượng không nguy hiểm.



Hình 4-16 Mô hình an toàn tích hợp

- Một vài thông số quan trọng ảnh hưởng chất lượng điều khiển ngõ ra:
 - ratioSlowdown/ ratioSpeedup – tốc độ tăng/giảm vận tốc điều khiển xe điện. Ngõ ra vận tốc điều khiển sẽ tỉ lệ với khoảng cách đối tượng (càng gần đối tượng tốc độ điều khiển giảm càng nhanh).
 - Duty - là khoảng thời gian giảm tốc giữa 2 lần liên tiếp, duty thay đổi tùy thuộc vào khoảng cách đối tượng.

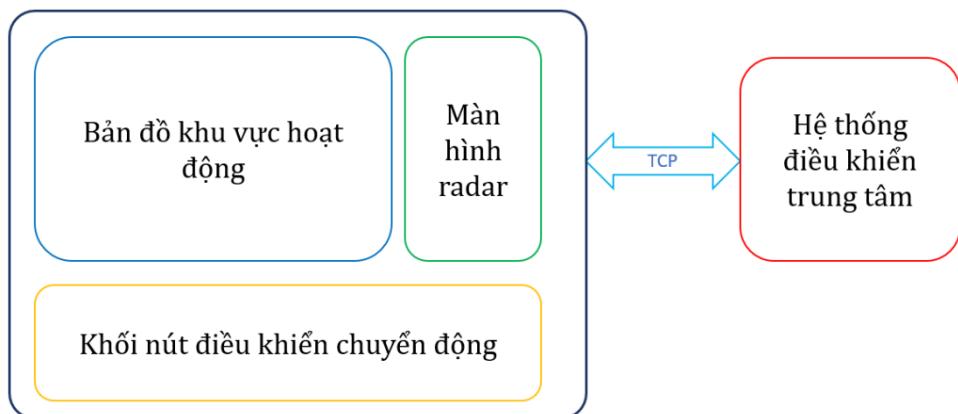
- numframeFilter – mỗi khung dữ liệu cảm biến gửi về có thông tin chuyển động của đối tượng, thời gian TTC sẽ dễ nhiễu loạn nếu có 1 khung bị lỗi. Thông số này giúp giữ cho đối tượng không bị đột ngột biến mất trong bộ nhớ điều khiển.
- speed_{max}/speed_{min} khoảng vận tốc ngõ ra của bộ điều khiển.

Trong tất cả các đối tượng nguy hiểm, những đặc trưng chuyển động có thể khác nhau và do đó, sẽ có thể có nhiều hơn các tác lệnh điều khiển trong toàn hệ thống. Đánh giá mức độ ưu tiên thực chất là đang so sánh mức độ nghiêm trọng của từng đối tượng. Ví dụ đơn giản cho đánh giá ưu tiên, hai đối tượng A và B đang trong giới hạn quan sát được của cảm biến, A ở phía trước và đang tiến gần về phía xe điện, B ở phía bên phải và cũng đang tiến lại gần cùng tốc độ. Tác lệnh đầu ra cho A là giảm tốc để tránh xảy ra va chạm và B sẽ phải rẽ trái để tránh đối tượng. Trong hệ thống đơn giản, môi trường lý tưởng, ta có thể lựa chọn rẽ trái để giảm rủi ro. (Để tối ưu ta cần áp dụng thuật toán tránh vật cản chuyên biệt).

4.6. Thiết kế giao diện người dùng

Giao diện sẽ được kết nối với hệ thống điều khiển trung tâm trong mạng LAN cục bộ thông qua giao thức TCP/IP. Bao gồm các khối chức năng như:

- Bản đồ mà khu vực xe đang hoạt động.
- Vị trí hiện tại, đường đi và trạng thái của xe.
- Các nút nhấn giúp ta có thể tự điều khiển xe trong quá trình thực nghiệm.
- Màn hình quan sát radar, hiển thị kết quả theo dõi các đối tượng phía trước xe.



Hình 4-17 Tổng quan giao diện người dùng

Sử dụng Framework Qt Creator C++ thực hiện thiết kế giao diện. Đây là một framework hỗ trợ mạnh mẽ không chỉ xây dựng giao diện chuyên nghiệp trong ngành công nghiệp thiết kế giao diện xe ô tô nói chung và xe điện tự hành nói riêng. Qt cũng hỗ trợ xây dựng phát triển thuật toán với ngôn ngữ C++, Python, hay các mô-đun giúp truy cập lên server để ứng dụng cho dự án muốn lưu trữ dữ liệu hay quan sát điều khiển từ xa.

Thiết kế các tính năng điều khiển xe

Ngoài việc hiển thị bản đồ giúp quan sát được vị trí hiện tại của xe, giao diện sẽ có các tính năng cơ bản để có con người có thể điều khiển xe khi thực nghiệm. Các nút nhấn bắt đầu chạy và dừng lại, nút nhấn chuyển chế độ chạy tự động hay con người sẽ điều khiển. Các nút dùng để bật tắt đèn hay còi xe, các thanh trượt giúp điều khiển vận tốc xe, điều khiển góc đánh lái của vô-lăng. Thao tác chọn điểm bắt đầu vào điểm kết thúc của quá trình di chuyển...

Như vậy để các tín hiệu đó được gửi đến bộ điều khiển trung tâm Qt có hỗ trợ mô-đun `QtcpSocket`. Khi các máy tính được kết nối chung mạng LAN thông qua router, các tín hiệu điều khiển từ giao diện sẽ được gửi đến trung tâm thông qua giao thức TCP.

Chương 5. KẾT QUẢ THỰC HIỆN

5.1. Kết quả thực hiện phần cứng

Tủ điện điều khiển sẽ đảm nhiệm vai trò là nơi phân phối nguồn cho các thiết bị, cảm biến, xử lý các giải thuật, thu thập dữ liệu, xuất xung cho động cơ, truyền thông và xử lý các tác vụ khẩn cấp. Từ đó, ta chia tủ điện điều khiển thành các khối, mỗi khối đảm nhiệm các chức năng, nhiệm vụ khác nhau và được giao tiếp, thông tin với nhau qua chuẩn giao tiếp CAN.



Hình 5-1 Tủ điện điều khiển đặt trên xe

- Phần nguồn bao gồm 3 bộ giảm áp cấp nguồn cho Vi xử lý, máy tính nhúng, quạt làm mát, driver điều khiển động cơ, ...
- Phần bộ xử lý có microcontroller STM32F1, STM32F4, máy tính nhúng Jetson Xavier Nx.
- Phần công suất có quạt làm mát, driver điều khiển động cơ "step steering"

- Chức năng đảm nhiệm:

- Chức năng của tủ điện là trung tâm điều khiển hoạt động trên xe điện tự lái. Chuyển đổi chức năng tự hành trên xe hoặc người dùng có thể lựa chọn chính bản thân điều khiển.
- Theo dõi nhiệt độ tủ điều khiển bằng cảm biến nhiệt độ LM35. Gửi thông báo cho người dùng khi xảy ra sự cố.
- Đọc và xử lý ngoại vi nút nhấn điều khiển các chế độ, bật/tắt các khối thành phần.
- Đọc và xử lý cảm biến encoder, camera, radar cho quá trình hoạt động tự hành của xe điện.
- Giao tiếp kết nối cùng lúc nhiều thiết bị trong mạng LAN cục bộ. Điều này giúp mở rộng thêm khả năng theo dõi hoạt động và phát triển cho hệ thống.



Hình 5-2 Chi tiết bên trong tủ điều khiển

- Đánh giá thời gian hoạt động liên tục đạt được hơn 10 giờ và đạt được sự ổn định khi xe điện hoạt động, di chuyển. Đáp ứng đầy đủ các chức năng điều khiển.

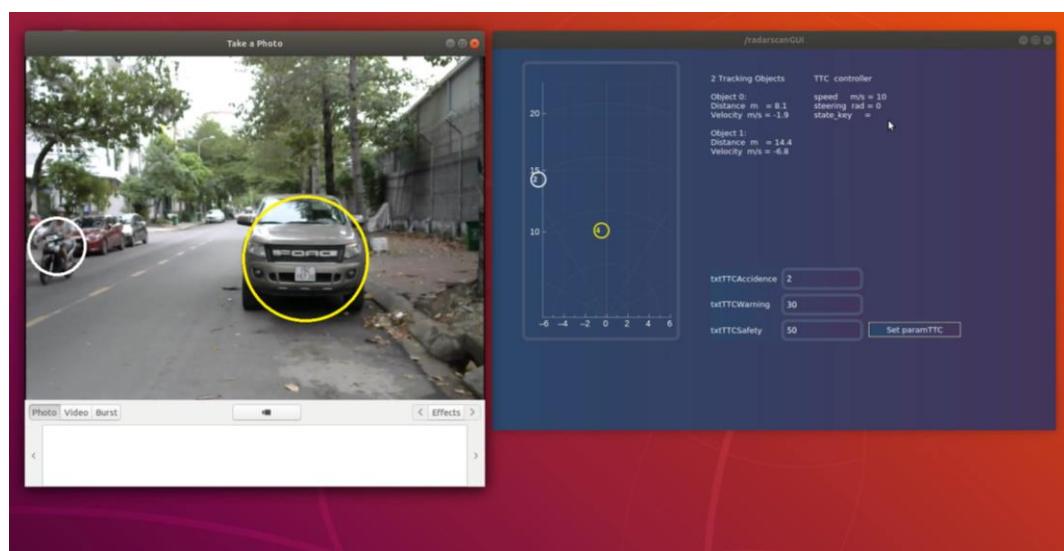
5.2. Kết quả xử lý cảm biến Radar

5.2.1. Kết quả phát hiện, theo dõi đối tượng

Kết quả cho quá trình theo dõi đối tượng xuất hiện phía trước cảm biến bao gồm vị trí tọa độ, khoảng cách của đối tượng đến xe điện, vận tốc tương đối so với xe, thời gian TTC ước tính.



Hình 5-3 Hình ảnh cảm biến radar trên xe điện



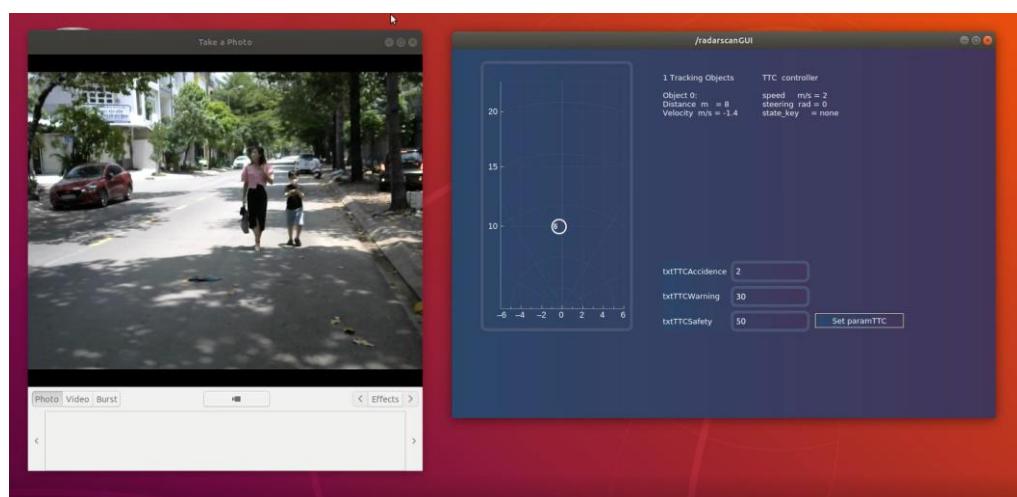
Hình 5-4 Kết quả phát hiện, theo dõi đối tượng thực tế (trái) và GUI (phải)

Từng đối tượng sẽ được định danh bởi 1 vòng tròn tương ứng. Từ giao diện màn hình quan sát radar, ta có được thông tin về vị trí cũng như chuyển động của đối tượng.

- Giá trị hiển thị bên trong những vòng tròn chính là thời gian TTC ước tính theo thời gian thực. Nếu **TTC > 100s** ta không hiển thị giá trị.
- Màu sắc mỗi vòng tròn sẽ tương ứng với ý nghĩa khác nhau:
 - *Màu trắng* thể hiện đối tượng không có nguy hiểm đối với xe điện (do góc gamma lớn hơn giới hạn được đề cập tại mục [4.3]).
 - *Màu xanh* thể hiện đối tượng có nguy cơ va chạm với xe điện nhưng giá trị TTC nằm trong khoảng an toàn.
 - *Màu vàng* thể hiện cảnh báo nguy hiểm, ảnh hưởng đến các tác lệnh của bộ điều khiển tránh va chạm.
 - *Màu đỏ* thể hiện mức nguy hiểm cao nhất, các tác lệnh khẩn cấp được thực thi.
- Ngoài ra còn có chi tiết thông số của các đối tượng phía bên phải màn hình radar: ID Object, Distance, Velocity.

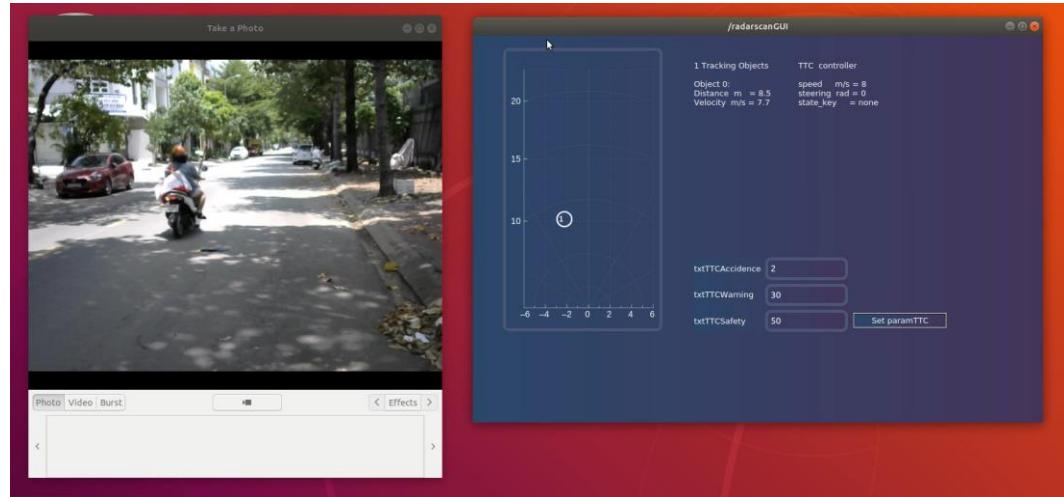
Kết quả phát hiện từng loại đối tượng khác nhau:

- Phát hiện đối tượng nhỏ, người chuyển động:



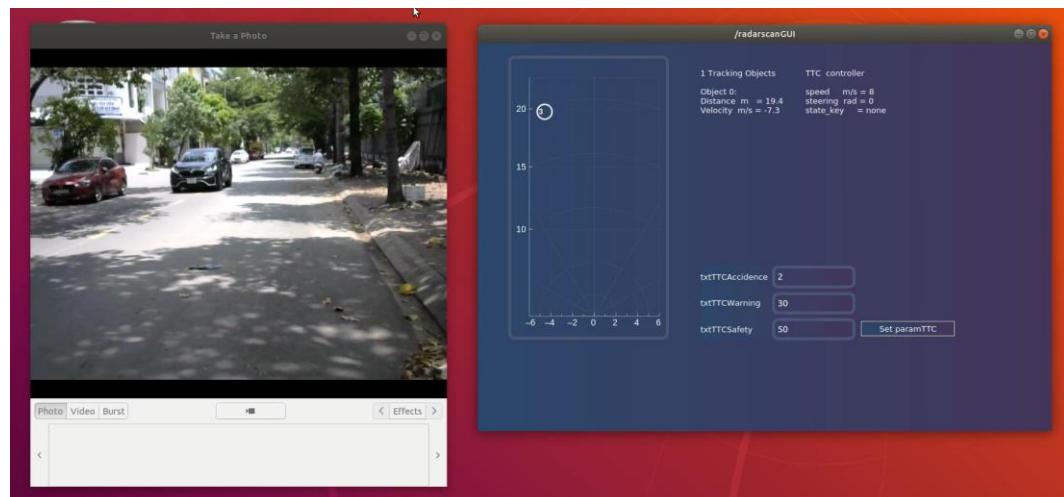
Hình 5-5 Kết quả phát hiện, theo dõi đối tượng người đi bộ

- Phát hiện đối tượng xe máy:



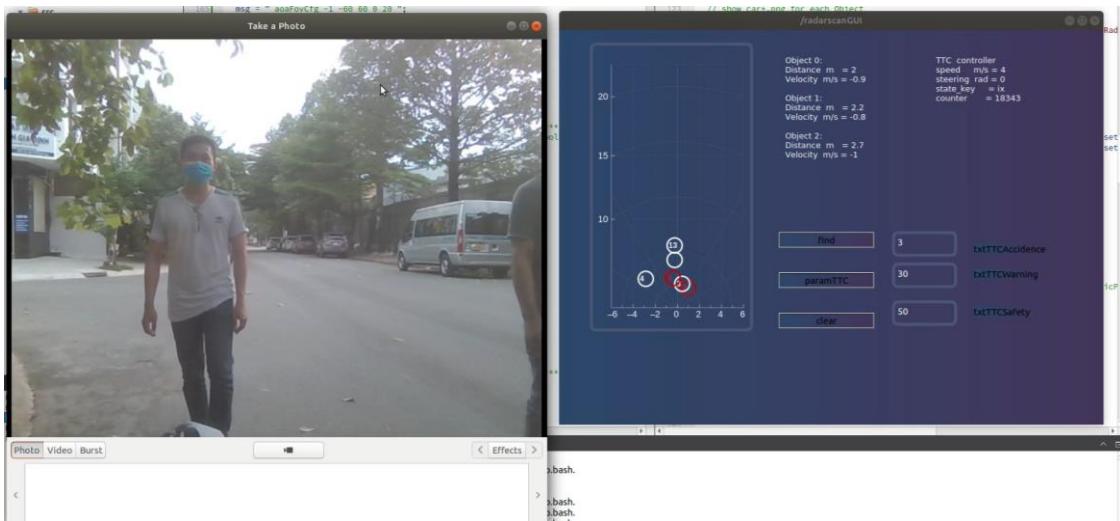
Hình 5-6 Kết quả phát hiện, theo dõi đối tượng xe máy

- Phát hiện đối tượng xe ô tô:



Hình 5-7 Kết quả phát hiện, theo dõi đối tượng xe ô tô

Một số kết quả cho thấy khả năng phát hiện đối tượng khá tốt, tuy nhiên như đã đề cập ở phần [4.2.1], vấn đề nhiều cảm biến cho ra các kết quả không thể sử dụng làm dữ liệu đầu vào bộ điều khiển tránh va chạm. Điều này do phần cứng, và quan trọng thông số cấu hình cảm biến radar chưa được tối ưu.



Hình 5-8 Kết quả phát hiện đối tượng bị nhiễu do cấu hình không tối ưu

- Đánh giá cho chức năng cảm biến là có thể ứng dụng trong cho bộ điều khiển tránh va chạm. Có khả năng phát hiện và theo dõi chuyển động của đối tượng. Một vài thông số đánh giá tiếp theo cho thấy được độ đáng tin cậy của cảm biến và hệ thống.

5.2.2. Kết quả xác định vị trí đối tượng

Tọa độ của vật được xác định trong hệ tọa độ của cảm biến radar, với gốc tọa độ tại điểm chính giữa của radar, chiều dương trục X hướng sang phải và chiều dương trục Y hướng ra xa xe điện. Mỗi giá trị được thực hiện 5 lần đo, giá trị tính toán trung bình được ghi nhận.

- Sai số trung bình được tính bằng công thức: $\overline{\Delta p} = \frac{1}{N} \sum_{i=1}^N \Delta p_i$
- Với N là số lần thực hiện, và: $\Delta p_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}$
- Sai số RMS được tính bằng công thức: (26)

$$\Delta p_{RMS} = \sqrt{\frac{\sum_{i=1}^N (\overline{\Delta p} - \Delta p_i)^2}{N}}$$

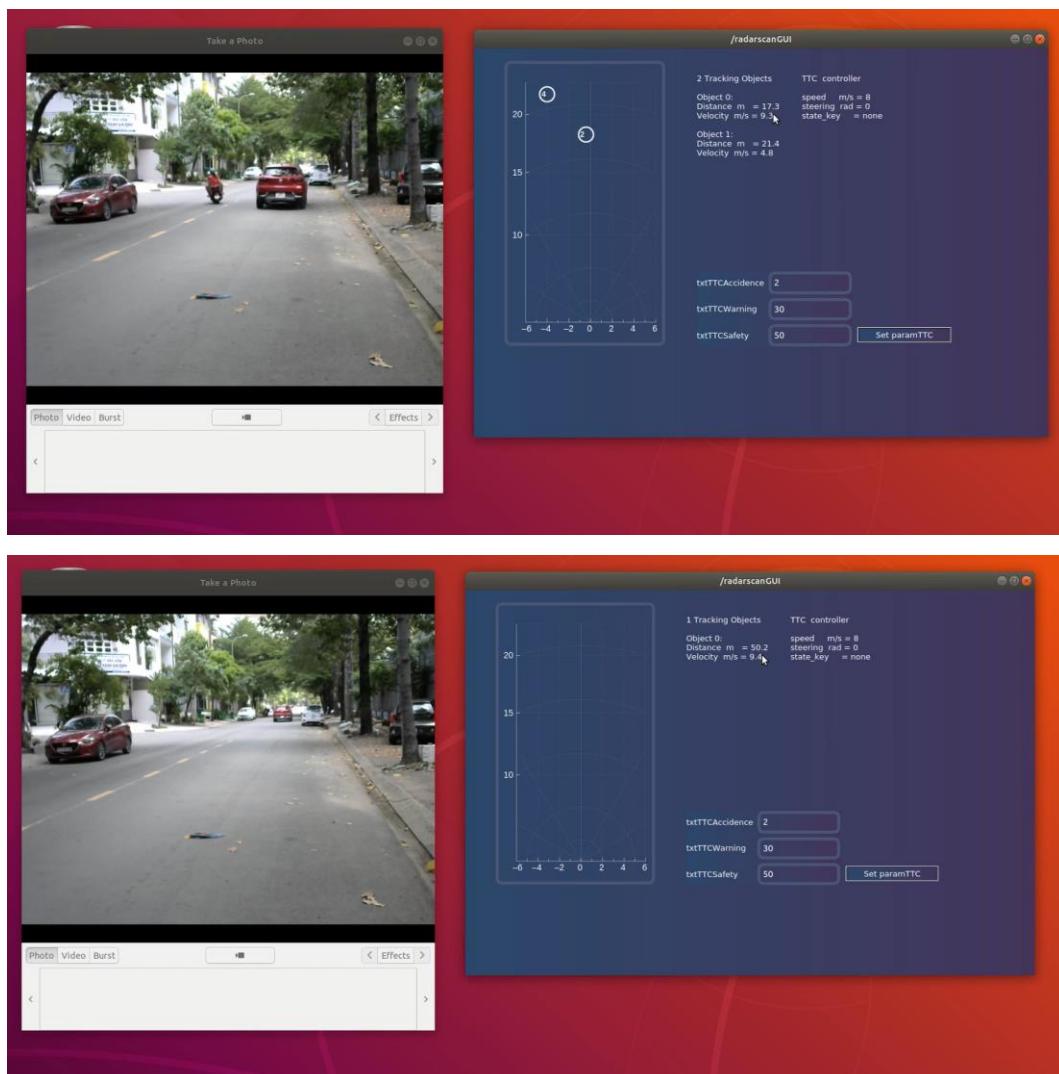
Bảng 5-1 Giá trị khảo sát vị trí vật cản

Số lần	Giá trị thực		Giá trị đo	
	Vị trí X (m)	Khoảng cách Y (m)	Vị trí X (m)	Khoảng cách Y (m)
1	1.2	6	1	5.9
2	1.2	6	1.1	6
3	1.2	6	1.1	6
4	1.2	6	1.2	6.1
5	1.2	6	1	6
6	4	12	4.1	11.5
7	4	12	4	11.7
8	4	12	4	11.6
9	4	12	4.1	12
10	4	12	4	12.3
11	5	24	5.1	24.2
12	5	24	4.8	24.5
13	5	24	4.9	25
14	5	24	4.8	23.5
15	5	24	5	24.2

Bảng 5-2 Kết quả xác định vị trí vật cản

Giá trị thực (m)		Giá trị tính trung bình		Sai số trung bình	Sai số RMS
X	Y	X	Y		
1.2	6	1.08	6	0.145	0.055
4	12	4.04	11.82	0.322	0.135
5	24	4.92	24.28	0.501	0.291

Trên giao diện màn hình radar, khoảng cách hiển thị tối đa là 22[m]. Nhưng trên thực tế, khoảng cách tối đa mà cảm biến radar có thể theo dõi đối tượng phía trước đạt được 50[m], vận tốc tối đa 12,57[m/s].



Hình 5-9 Khoảng cách tối đa cảm biến có thể phát hiện đối tượng.

- **Nhận xét:** Kết quả phát hiện vị trí đối tượng phụ thuộc nhiều vào phần cứng cũng như thông số cấu hình thích hợp cho cảm biến radar. Có thể thay các giá trị như “Range max”, “Velocity max”, “Range resolution”, “Velocity resolution” - đề cập ở phần cấu hình cảm biến mục [4.2.1] - để đáp ứng được yêu cầu ứng dụng. (Xem phụ lục để cấu hình thông số cho cảm biến).

Xe điện tự hành hoạt động trong khu vực có giới hạn tốc độ, lượng phương tiện giao thông vừa phải nên những thông số trên có thể đáp ứng cho việc phát hiện và tránh đối tượng. Có thể giảm mức sai số đến

dưới mức 5[cm] nhưng sẽ bị giới hạn khoảng cách tối đa “Range max”, tương tự cho vận tốc. Ở ứng dụng cho đối tượng chuyển động nhanh và cần theo dõi từ xa, cấu hình cảm biến radar sẽ ưu tiên cho “Range max” thay vì độ phân giải khoảng cách “Range resolution”. Sai số chấp nhận nhỏ hơn 1[m].

5.2.3. Kết quả xác định chuyển động đối tượng

Tốc độ của vật được xác định trong hệ tọa độ của cảm biến radar, với gốc tọa độ tại điểm chính giữa của radar, chiều dương trục X hướng sang phải và chiều dương trục Y hướng ra xa xe điện. Mỗi giá trị được thực hiện 5 lần đo, giá trị tính toán trung bình được ghi nhận

Bảng 5-3 Giá trị khảo sát tốc độ của đối tượng

Số lần	Giá trị thực vận tốc (m/s)	Giá trị đo vận tốc (m/s)
1	2.7	2.9
2	2.7	2.9
3	2.7	3
4	2.7	2.7
5	2.7	2.8
6	5.5	5.2
7	5.5	5.5
8	5.5	5.3
9	5.5	5.3
10	5.5	5.5
11	10	9.1
12	10	9.3
13	10	9.2
14	10	9.2
15	10	9.4

Bảng 5-4 Kết quả xác định tốc độ của đối tượng

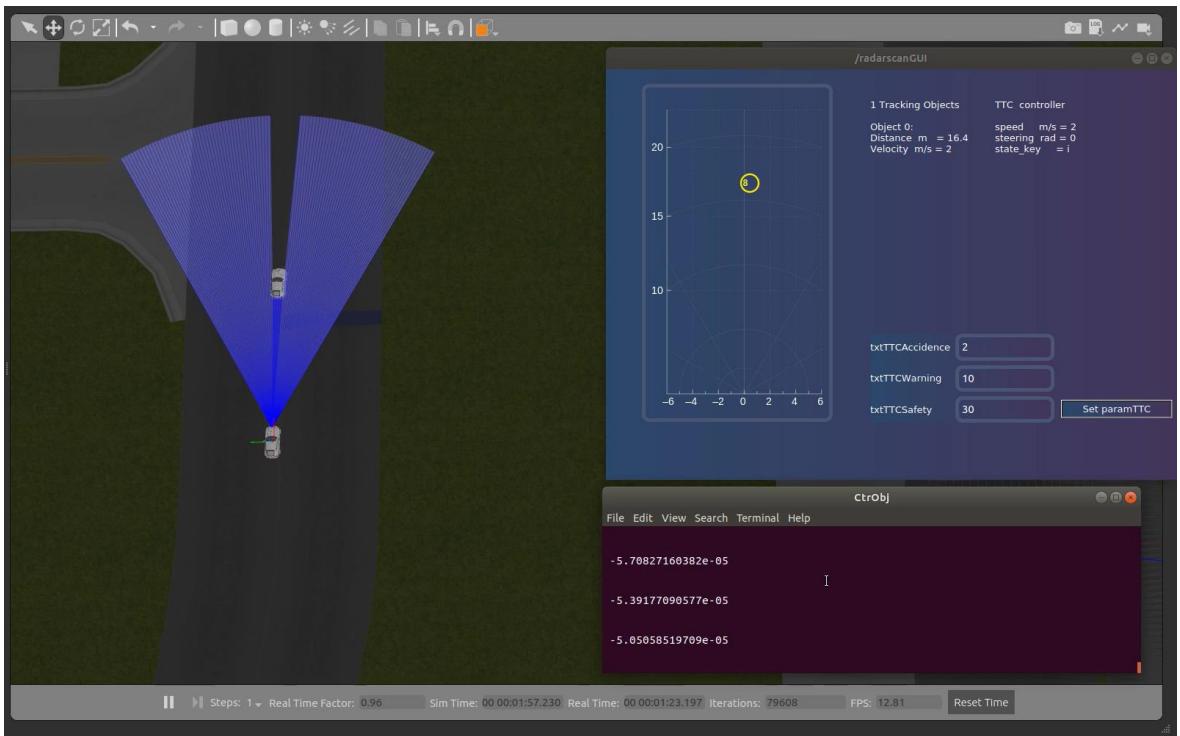
Giá trị thực (m)	Giá trị tính trung bình	Sai số trung bình	Sai số RMS
2.7	2.86	0.16	0.102
5.5	5.36	0.14	0.120
10	9.24	0.76	0.102

- **Nhận xét:** Kết quả đo đạt tốc độ của đối tượng ánh hưởng trực tiếp đến kết quả tính toán thời gian va chạm TTC. Sai số vận tốc trung bình khoảng 1[m/s] là tương đối lớn nhưng có thể ứng dụng vào thuật toán với điều kiện tốc độ các đối tượng chuyển động cần đạt mức nhỏ hơn 10[m/s]. Điều này không bắt buộc cho bộ tránh vật cản nhưng để tăng tính an toàn trong khi thử nghiệm, điều này cần đảm bảo.

5.3. Kết quả điều khiển

5.3.1. Kết quả mô phỏng

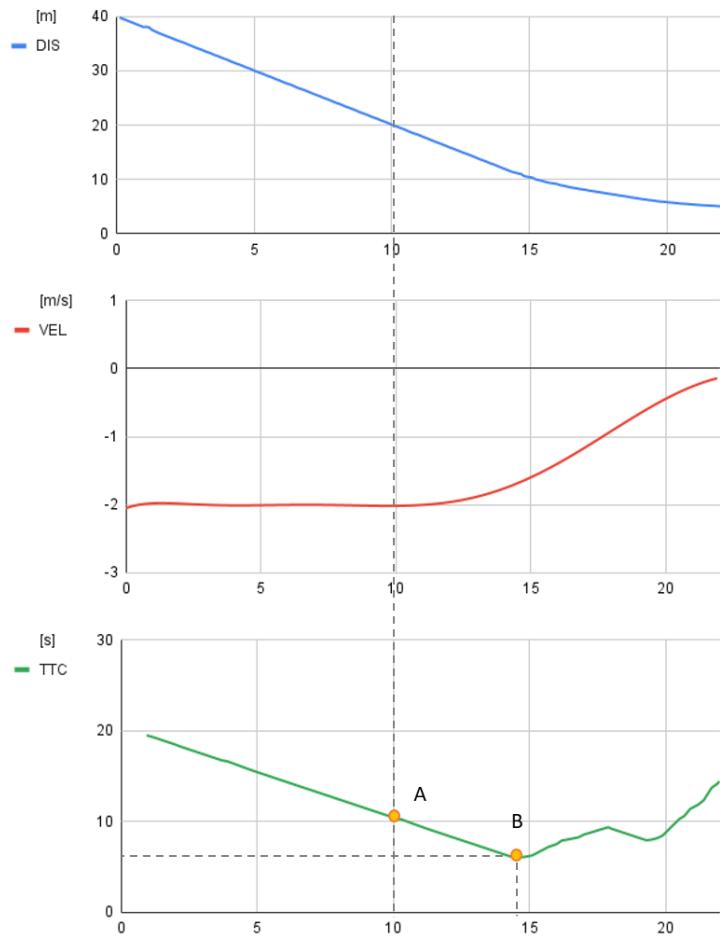
Thực hiện kiểm tra giả lập môi trường vận hành trên Gazebo ROS. Hai phương tiện “Car1” và “Car2” trong cùng một môi trường. Car1 là phương tiện được trang bị cảm biến LaserScan (kết quả ngõ ra tương đồng với cảm biến radar đang sử dụng) giúp phát hiện và theo dõi đối tượng phía trước. Car1 tự lái với tốc độ điều khiển 2[m/s]. Đối tượng tĩnh Car2 ở xa phía trước hướng di chuyển của xe. Cho Car1 chạy tự động, thu thập số liệu như kết quả:



Hình 5-10 Mô phỏng môi trường hoạt động

- Mô phỏng mô hình phương tiện Carsim_gazebo trên môi trường Ubuntu ROS. Có thể tự do di chuyển các hướng hoặc được điều khiển bởi người dùng thông qua các nút nhấn trên bàn phím “keyboard”.
- Có khả năng kết nối với các hệ thống khác trong ROS như kết nối với GUI điều khiển thực tế để hiển thị kết quả từ môi trường mô phỏng.
- Bộ điều khiển tránh va chạm TTC được ứng dụng vào xe có trang bị cảm biến LaserScan. Điều khiển tốc độ và hướng di chuyển của xe.
- Có thể điều khiển xe thứ hai để mô phỏng tránh đối tượng chuyển động. Kết quả đạt được tương tự như với đối tượng tĩnh.

Kết quả mô phỏng điều khiển tránh va chạm:



Hình 5-11 Kết quả mô phỏng điều khiển tránh vật thể

Khảo sát ba đồ thị: khoảng cách (DIS), vận tốc (VEL), thời gian va chạm (TTC) bắt đầu từ khi cảm biến radar phát hiện đối tượng và xe điện đạt vận tốc 2[m/s]. Hai điểm giao trên đồ thị TTC, điểm A tại mốc thời gian $t = 10[\text{s}]$, điểm B tại thời điểm $t = 14.7[\text{s}]$

- Khoảng thời gian xe điện di chuyển từ $t = 0$ đến $t = A$, xe giữ không đổi tốc độ, khoảng cách giảm càng gần đối tượng, thời gian TTC giảm nhưng trong giới hạn an toàn.
- Tại vị trí $t = A$, tác lệnh giảm tốc từ bộ điều khiển tránh va chạm được thực hiện, vận tốc điều khiển giảm tỉ lệ với khoảng cách đối tượng (càng gần đối tượng tốc độ điều khiển giảm càng nhanh).

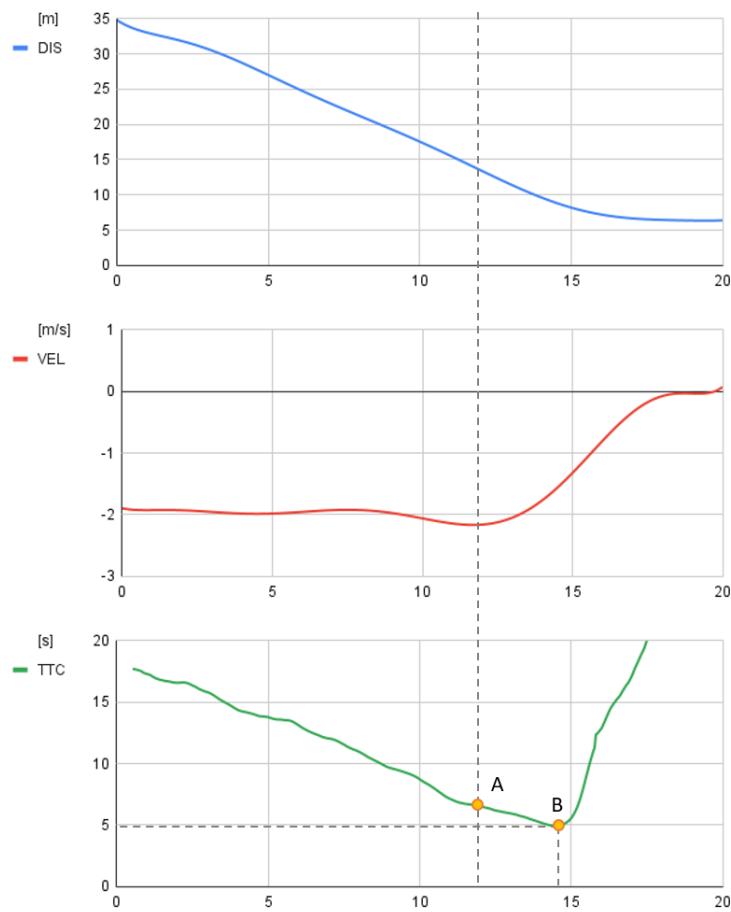
- TTC giảm đến vị trí $t = B$ giá trị $TTC \approx 7$. Vận tốc giảm, do đó thời gian TTC tăng và đến khi xe điện dừng hẳn, TTC đạt vô cùng lớn \rightarrow Không xảy ra va chạm.

5.3.2. Kết quả thực tế thuật toán TTC – Điều khiển tránh va chạm đối tượng tĩnh

Thực hiện kiểm tra trên đoạn đường chạy thực tế. Xe điện tự lái với tốc độ điều khiển 10[km/h]. Đối tượng tĩnh ở xa phía trước hướng di chuyển của xe (bộ tránh va chạm sẽ bỏ qua những vật có hướng di chuyển không hướng vào xe, hay $posX_{Approach}$ vượt ngưỡng cho phép. Mục [4.3]-vector isApproach). Cho xe điện chạy tự động, thu thập số liệu như kết quả:



Hình 5-12 Xe điện tự động giảm tốc khi đối tượng quá gần



Hình 5-13 Kết quả thực tế điều khiển tránh vật thể tĩnh

Khảo sát ba đồ thị: khoảng cách (DIS), vận tốc (VEL), thời gian va chạm (TTC) bắt đầu từ khi cảm biến radar phát hiện đối tượng và xe điện đạt vận tốc $2[m/s]$. Hai điểm giao trên đồ thị TTC, điểm A tại mốc thời gian $t = 12.4[s]$, điểm B tại thời điểm $t = 14.7[s]$

- Khoảng thời gian xe điện di chuyển từ $t = 0$ đến $t = A$, xe giữ không đổi tốc độ, khoảng cách giảm càng gần đối tượng, thời gian TTC giảm nhưng trong giới hạn an toàn.
- Tại vị trí $t = A$, tác lệnh giảm tốc từ bộ điều khiển tránh va chạm được thực hiện, vận tốc điều khiển giảm tỉ lệ với khoảng cách đối tượng (càng gần đối tượng tốc độ điều khiển giảm càng nhanh).

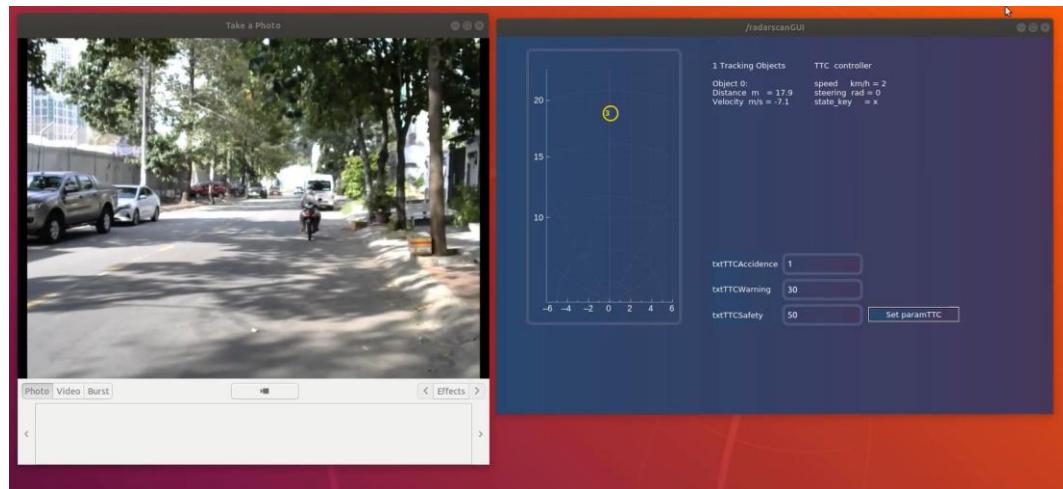
- TTC giảm đến vị trí $t = B$ giá trị $TTC \approx 5$. Vận tốc giảm, do đó thời gian TTC tăng và đến khi xe điện dừng hẳn, TTC đạt vô cùng lớn \rightarrow Không xảy ra va chạm.
- **Nhận xét:** Kết quả điều khiển đạt được như mong muốn. Thời gian va chạm TTC đạt mức thấp nhất $TTC_{min} = 5s$ là cần thiết để tránh xảy ra va chạm hay cần thiết để chuẩn bị các tác lệnh khẩn cấp. Và thời gian trị $TTC < 1.5s$ là tình huống được đánh giá là nguy hiểm xảy ra va chạm. “Tiêu chí TTC trong phạm vi từ 4,5 đến 5 giây để kích hoạt CAS là điều đáng giá” theo [12]. Cần điều chỉnh thông số điều khiển đề cập ở mục [4.5] để đạt tốc độ và thời gian TTC mong muốn.

Ta có thể quan tâm đến khoảng cách gần nhất từ xe điện đến đối tượng sau khi thực hiện xong tác lệnh từ bộ điều khiển. Khoảng cách an toàn đến khi xe dừng hẳn từ 5m đến 8m (phụ thuộc vào thông số duty).

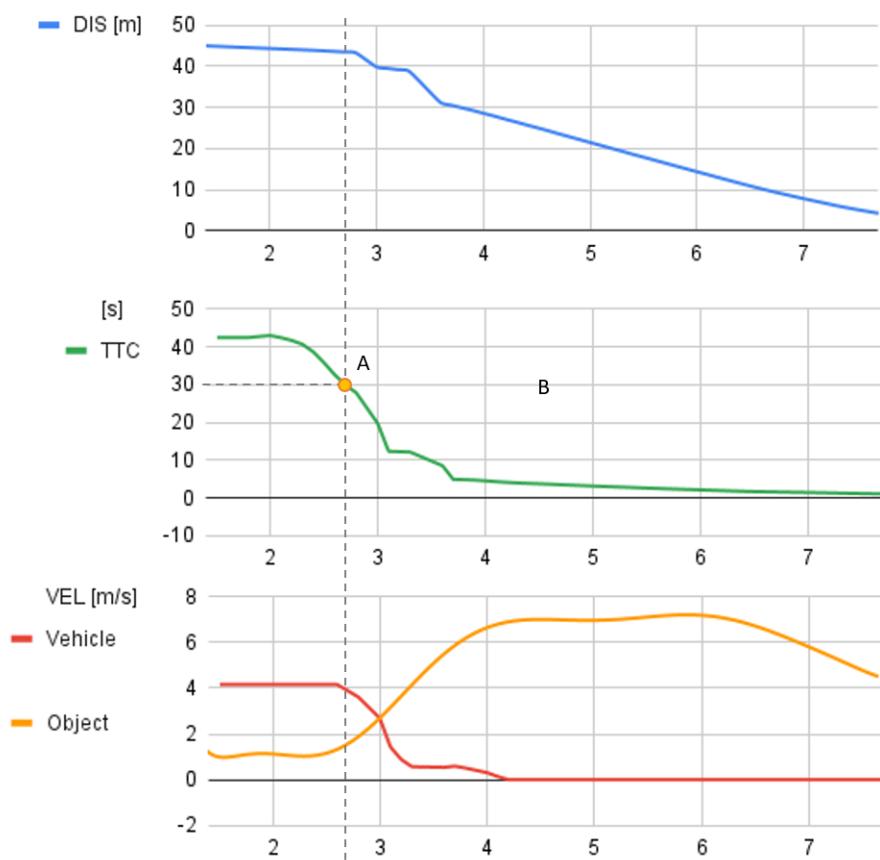
Kết quả mô phỏng và chạy thực tế là tương đồng. Kết quả mô phỏng ổn định hơn thực tế do cảm biến radar khi di chuyển có rung lắc ảnh hưởng kết quả ngõ ra cảm biến.

5.3.3. Kết quả thực tế thuật toán TTC – Điều khiển tránh va chạm đối tượng đang chuyển động

Thực hiện kiểm tra trên đoạn đường chạy thực tế. Xe điện tự lái với tốc độ điều khiển 15[km/h]. Đối tượng chuyển động ở xa phía trước hướng di chuyển của xe, di chuyển đâm trực diện vào xe điện. Cho xe điện chạy tự động, thu thập số liệu như kết quả:



Hình 5-14 Đối tượng đang chuyển động đâm thẳng trực diện vào xe điện.



Hình 5-15 Kết quả thực tế điều khiển tránh vật thể đang chuyển động

Khảo sát ba đồ thị: khoảng cách (DIS), vận tốc (VEL), thời gian va chạm (TTC) bắt đầu từ khi cảm biến radar phát hiện đối tượng và xe điện đạt vận tốc 15[km/h] khoảng 4.2[m/s]. Điểm giao trên đồ thị TTC, điểm A tại mốc thời gian $t = 2.6[s]$

- Tại vị trí $t = A$, thuật toán TTC phát hiện đối tượng thuộc nhóm nguy hiểm có khả năng va chạm với xe điện. Tác lệnh giảm tốc từ bộ điều khiển tránh va chạm được thực hiện, vận tốc điều khiển (đường màu đỏ) giảm tỉ lệ với khoảng cách đối tượng (càng gần đối tượng tốc độ điều khiển giảm càng nhanh).
 - Sau khi đã giảm tốc độ, thời gian TTC có xu hướng đi ngang, có thể đánh giá là đã giảm được nguy hiểm, nhưng do vận tốc của đối tượng (đường màu cam) có xu hướng tiếp tục tăng lên, cảnh báo nguy hiểm lại được thực hiện, xe giảm tốc độ về 0[km/h] để giảm mức nghiêm trọng của tai nạn.
- **Nhận xét:** Thời gian va chạm TTC đạt mức thấp nhất $TTC_{min} = 0.96[s]$ được đánh giá là không đủ thời gian để tránh được va chạm theo [12]. Bộ điều khiển đã phát hiện sớm và thực hiện tác lệnh nhưng trong trường hợp này, đối tượng chuyển động vẫn tiếp tục vận tốc nguy hiểm nên va chạm không thể tránh được.

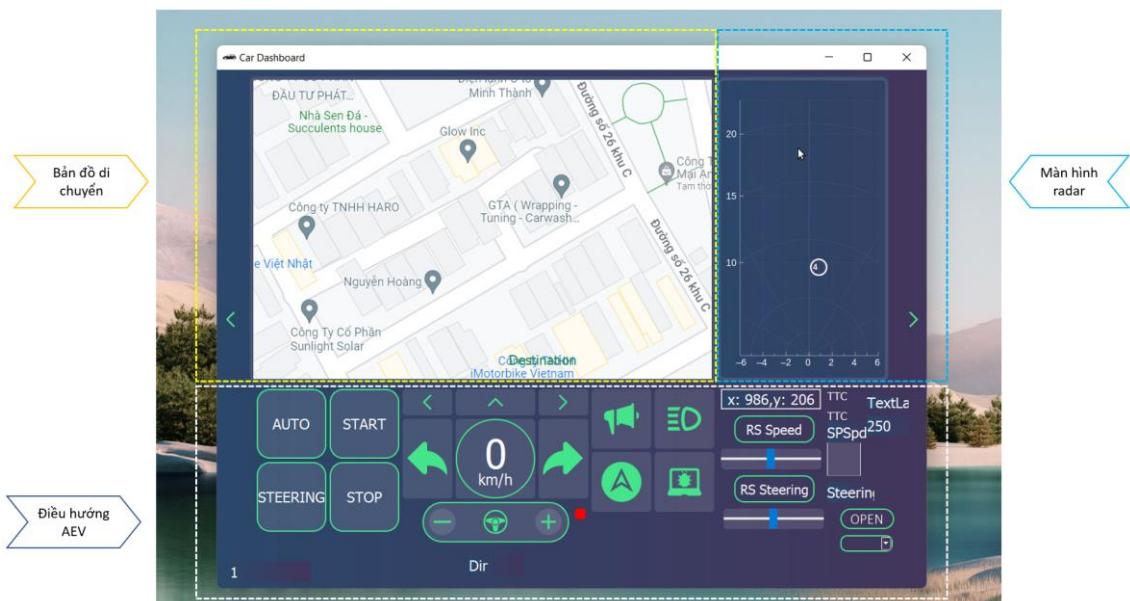
Kết quả cho thấy bộ điều khiển hoạt động đúng như thiết kế, tác động điều khiển giúp giảm được mức độ nguy hiểm va chạm. Nếu cả xe điện và đối tượng cùng duy trì tốc độ trước đó thì mức độ nguy hiểm của tai nạn là rất cao. Điều cần thiết ở đây là phát hiện được nguy hiểm từ xa để thực hiện điều khiển an toàn.

Để có thể tránh được hoàn toàn tai nạn như trường hợp trên, ta cần phát triển, kết hợp thêm các bộ điều khiển tránh vật cản vào hệ thống. Điều

này sẽ giúp vẽ được quỹ đạo di chuyển tốt nhất cho xe điện có thể tránh được đối tượng, tránh được tai nạn xảy ra.

5.3.4. Giao diện người dùng sử dụng

Màn hình quan sát radar hiển thị chi tiết các đối tượng trong phạm vi theo dõi, điều này là cần thiết để có thể giám sát khách quan hoạt của hệ thống. Trong quá trình vận hành, màn hình hiển thị trên xe (giao diện người dùng) cần được tối ưu để tăng khả năng tập trung cho người giám sát, một số chi tiết cần được ẩn bên trong và tích hợp thành một hệ thống nhất. Vì đó, màn hình radar sẽ được tối ưu và kết hợp với giao diện hiện có trên xe điện:



Hình 5-16 Màn hình radar được tích hợp trên GUI hiện tại

Ta có thể xem chi tiết các đối tượng ở các Tab sau của giao diện người dùng. Các chức năng điều khiển sẽ được tích hợp trên một GUI tổng quát. Điều này đáp ứng được nhờ các thiết bị được kết nối với cùng một "master" trong hệ thống ROS. Dữ liệu ngõ ra của hệ thống tránh va chạm gửi dữ liệu và GUI điều khiển nhận kết quả.

- Bản đồ di hiển thị bản đồ giúp quan sát được vị trí hiện tại của xe, giao diện sẽ có các tính năng cơ bản để có con người có thể điều khiển xe khi thực nghiệm.
- Các nút nhấn bắt đầu chạy và dừng lại, nút nhấn chuyển chế độ chạy tự động hay con người sẽ điều khiển. Các nút dùng để bật tắt đèn hay còi xe, các thanh trượt giúp điều khiển vận tốc xe, điều khiển góc đánh lái của vô-lăng. Thao tác chọn điểm bắt đầu vào điểm kết thúc của quá trình di chuyển...
- Màn hình quan sát radar, hiển thị kết quả theo dõi các đối tượng phía trước xe.

Chương 6. ĐÁNH GIÁ, KẾT LUẬN, HƯỚNG PHÁT TRIỂN

6.1. Đánh giá kết quả

Tủ điện phần cứng có thể đánh giá tủ điện điều khiển hoạt động tốt và đáp ứng đầy đủ các chức năng yêu cầu hiện tại. Bộ nguồn ắc quy của tủ điều khiển cung cấp liên tục và phục vụ đủ cho thời gian xe điện hoạt động, di chuyển. Các chức năng điều khiển tiến, lùi, chuyển đổi chế độ Auto/Manual, đọc giá trị cảm biến, kết nối ngoại vi hoạt động đúng thiết kế; kích thước tủ điện 35cm x 25cm x 15cm có thể lắp đặt dễ dàng trên xe. Tối ưu kết nối bằng những đầu nối chuyên dụng, tránh lắp ngược và sai vị trí.

Thiết bị cảm biến radar có ngõ ra phù hợp để thực hiện điều khiển tránh va chạm. Hoạt động ổn định và liên tục trong môi trường thực nghiệm. Kích thước nhỏ gọn dễ dàng tháo lắp vào xe điện.

Cảm biến hoạt động liên tục sẽ có hiện tượng tăng nhiệt độ, nhất là khi hoạt động trực tiếp dưới ánh nắng gắt. Khoảng cách tối đa có thể phát hiện đối tượng là 50[m], khoảng cách này sẽ giúp tăng thêm độ tin cậy cho hệ thống tránh va chạm nếu đạt được ngưỡng 100[m] hoặc hơn. Nhưng với môi trường ứng dụng hiện tại, giới hạn này có thể sử dụng mà vẫn đảm bảo tính an toàn. Do đặc trưng về góc quét nên khi lắp đặt cần cẩn chỉnh góc thẳng, song song với mặt đất để có kết quả tốt nhất. Phần cứng cảm biến tiếp xúc trực tiếp môi trường, đảm bảo không vật cản phía trước vị trí lắp đặt và do đó, cảm biến hoạt động môi trường thời tiết tốt không có mưa hay giọt nước.

Bộ điều khiển tránh va chạm – sử dụng thuật toán TTC được xây dựng và ứng dụng vào thực tế. Kết quả điều khiển tương đồng với kết quả thực hiện mô phỏng. Nhận dữ liệu ngõ vào từ cảm biến radar, xử lý dữ những liệu cần thiết. Theo dõi đối tượng, đánh giá nguy hiểm đối tượng, đưa ra tác lệnh bảo vệ an toàn; Kết nối được với toàn hệ thống trên xe điện, gửi dữ liệu xử lý đến “master”. Có thể ứng dụng điều khiển cho hoạt động tự hành của xe điện.

Qua các khảo sát thực tế, một số trạng thái không mong muốn xảy ra trong quá trình tự hành của xe điện như xe tự động dừng đột ngột khi di chuyển. Điều này là do lỗi vào của hệ thống bị nhiều. Xe dừng đột ngột là do cảm biến radar phát hiện đối tượng “ảo” nhưng không được loại bỏ, vẫn đánh giá là đối tượng nguy hiểm và do đó, bộ điều khiển thực hiện các tác lệnh khẩn cấp khiến xe giảm tốc độ. Có thể khắc phục bằng cách tăng “numframeFilter” cho bộ lọc nhưng sẽ tạo độ trễ cho bộ đánh giá. Hoặc một cách khác là thay đổi thông số cấu hình cho cảm biến để tăng độ chính xác, thêm bộ lọc thông cao “High Pass Filter” mà cảm biến hỗ trợ sẵn. Cần kết hợp điều chỉnh để có kết quả tốt nhất.

Giao diện người dùng, cụ thể là màn hình quan sát radar hiển thị được trực quan đối tượng, giúp quan sát kết quả điều khiển hỗ trợ cho người giám sát. Từ màn hình có thể biết nhanh thông tin của đối tượng đang theo dõi: vị trí, tốc độ tương đối, thời gian va chạm TTC, mức độ nguy hiểm đối với xe... Tuy nhiên, cần tối ưu hơn về khoảng cách hiển thị để trực quan hơn với thực tế. (Giao hiện hiện tại chỉ hiển thị đối tượng từ 0 – 22[m], những đối tượng xa hơn vẫn được theo dõi nhưng không đủ không gian hiển thị).

6.2. Kết luận đề tài

Tuy vẫn còn hạn chế về phần cứng cảm biến radar, nhưng kết quả thực hiện cho thấy phương tiện xe tự hành có thể hoạt động thực tế, tự động điều khiển tránh, làm giảm khả năng xảy ra các tình huống tai nạn va chạm.

Thiết kế hệ thống điều khiển ứng dụng thực tế lý thuyết Time to Collision vào phương tiện xe điện tự hành. Với khả năng tự động theo dõi đối tượng phía trước hướng di chuyển của xe và tự động điều khiển để hạn chế thấp nhất nguy cơ xảy ra va chạm.

6.3. So sánh sản phẩm ứng dụng trên thị trường

Telematics FMC125 của hãng Teltonik [13]



Hình 6-1 Telematics FMC125

Hệ thống cảnh báo va chạm bao gồm camera ADAS được gắn trên cửa sổ phía trước của xe và thiết bị theo dõi GPS tiên tiến Teltonika Telematics FMC125, cả hai đều được kết nối thông qua giao diện RS-232. Camera ADAS phát hiện các sự kiện đáng báo động ở phía trước và gửi tất cả dữ liệu đến máy chủ chuyên dụng để theo dõi và phân tích thêm.Thêm vào đó, nó hiển thị thông báo cho người lái xe về các sự kiện phía trước xe.

Chức năng nổi bật:

- Cảnh báo va chạm với người đi bộ: thông báo cho người lái xe về bất kỳ người đi bộ, xe đạp hoặc xe máy nào phía trước.
- Cảnh báo chêch làn đường: giúp lấy lại hướng nếu người lái xe vô ý chêch làn đường.
- Cảnh báo va chạm phía trước: gửi cảnh báo nếu sắp xảy ra va chạm để hỗ trợ người lái xe duy trì khoảng cách di chuyển an toàn.
- Cảnh báo khoảng cách phía trước: thông báo cho người lái xe nếu xe di chuyển về phía trước trong khi xe khác đang ở trong phạm vi phát hiện, có thể được thiết lập ở 1 m, 2m, 3m.
- Báo động khởi động xe phía trước: thông báo cho người lái xe nếu xe phía trước bắt đầu di chuyển về phía trước từ tốc độ 0 mà xe không chuyển động trong vòng 2s.

- Cảnh báo khoảng cách an toàn: thu hút sự chú ý của người lái xe để giữ khoảng cách an toàn phía trước bằng cách cảnh báo người lái xe (hoạt động từ 30 km/h).
- Nhận dạng giới hạn tốc độ: nhận biết các biển báo giới hạn tốc độ và đưa ra cảnh báo khi vượt quá tốc độ.
- Đầu ghi video kỹ thuật số: ghi lại các cảnh ở độ phân giải HD vào thẻ SD trước và sau khi xảy ra tai nạn (mỗi phút trong một vòng lặp)

So sánh về chức năng, hệ thống hiện tại của đề tài nghiên cứu thực hiện được khả năng tránh va chạm, cảnh báo nguy hiểm mà module Telematics FMC125 đề cập. Nhưng với khả năng tối ưu sâu và nhúng vào bên trong khối điều khiển của xe điện, ta đánh giá bộ điều khiển tránh va chạm có các tác lệnh tự động ngăn chặn những tình huống nguy hiểm, trực tiếp bảo vệ an toàn. Do Telematics sử dụng camera để phát hiện đối tượng, vấn đề này dẫn đến tình trạng cần phải hiệu chuẩn thông số ban đầu (calibration) để hoạt động được chính xác. Điều này cần một số thiết bị chuyên dụng để thực hiện. Một số chức năng khác như phát hiện làn đường đã và đang được phát triển trên xe điện. Kết quả đánh giá của đề tài nghiên cứu cần nhiều thời gian tối ưu để hoàn thiện và sớm ứng dụng rộng rãi trong thực tế.

Một sản phẩm khác Module ADAS tránh va chạm phía trước TREK-130 của hãng Advantech [14] là một mô-đun hỗ trợ người lái nâng cao tiên tiến, đa chức năng kết hợp cảnh báo va chạm phía trước và cảnh báo chệch làn đường.



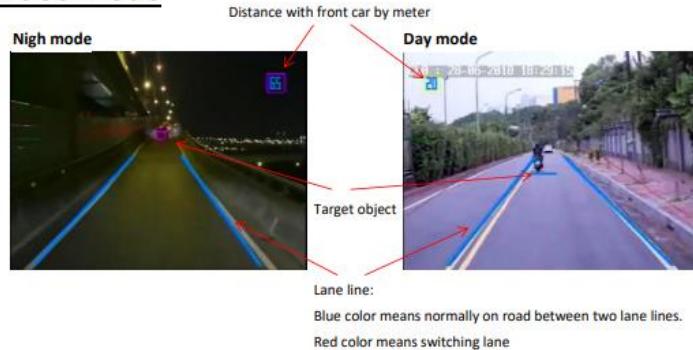
Hình 6-2 TREK - 130 module ADAS

Đây là một giải pháp an toàn chủ động dựa trên tầm nhìn để ngăn ngừa tai nạn và giảm thiểu thương tích bằng cách sử dụng công nghệ nhận dạng video. Module ADAS này có thể phát hiện xung quanh xe và cảnh báo trước cho người lái xe bằng cảnh báo âm thanh nếu xác định được tình huống rủi ro cao. Các chức năng nổi bật của sản phẩm:

- Phân tích video thông minh:

- Cảm biến camera giám sát các vạch kẻ làn đường để phát hiện xem xe có chạy sang đường khác hay không. Nếu hệ thống phát hiện thấy xe đang đi chệch hướng, cảnh báo bằng hình ảnh và âm thanh sẽ được phát ra để cảnh báo người lái.

Video mode



Hình 6-3 Cảnh báo lấn làn đường

- Đối với các ứng dụng tránh va chạm phía trước, cảm biến máy ảnh xử lý các hình ảnh được chụp bởi máy ảnh phía trước để phát hiện xe phía trước và tiềm ẩn nguy cơ va chạm. Nếu một chiếc xe được phát hiện trong phạm vi gần nguy hiểm, cảnh báo hình ảnh và âm thanh được phát ra để cảnh báo người lái xe.

6.4. Hướng phát triển

- Nâng cấp *phần cứng* cảm biến radar đang sử dụng để đạt kết quả tốt hơn: tăng khoảng cách phát hiện vật thể, tốc độ xử lý nhanh hơn (hiện tại 60Ghz), tăng độ phân giải đo đạc, ...
- Sử dụng thêm các bộ lọc tần số cho cảm biến radar.
- Tính toán, chỉnh sửa các thông số chi tiết *cấu hình ban đầu* cho cảm biến radar (rất quan trọng cho chất lượng ngõ ra cảm biến).
- Tăng khả năng *phát hiện đối tượng phía sau*, xử lý lại ngõ vào cho hệ thống tránh va chạm, nâng độ an toàn cho toàn hệ thống, người sử dụng.
- Thêm chức năng tránh vật thể để có thể duy trì chuyển động của xe.
- Kết hợp cảm biến camera để tăng độ chính xác.
- Cải thiện giao diện quan sát người dùng.
- Hoàn thiện hệ thống có thể ứng dụng di động như một module, tương tự như các hệ thống ADAS trên thị trường.

TÀI LIỆU THAM KHẢO

- [1] Hobbs, F.D, "Traffic planning and engineering," Pergamon Press, Oxford, 1989.
- [2] F. H. Administration, "Surrogate Safety Measures From Traffic Simulation Models," 2003.
- [3] Jie Ji, Amir Khajepour, Wael William Melek, Yanjun Huang, "Path Planning and Tracking for Vehicle Collision Avoidance Based on Model Predictive Control With Multiconstraints," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 952 - 964, 2016.
- [4] J. Hayward, "Near-miss determination through use of a scale of danger," The Pennsylvania State University, Pennsylvania, 1972.
- [5] C. HYDEN, "THE DEVELOPMENT OF A METHOD FOR TRAFFIC SAFETY EVALUATION: THE SWEDISH TRAFFIC CONFLICTS TECHNIQUE," *BULLETIN LUND INSTITUTE OF TECHNOLOGY, DEPARTMENT*, no. 70, p. 57, 1987.
- [6] "The subjective evaluation of traffic conflicts based on an internal concept of dangerousness," vol. 23, no. 1, pp. 53-65, 1991.
- [7] Tiwari G., Mohan D., Fazio J, "Conflict analysis for prediction of fatal crash locations in mixed traffic streams," *Accident Analysis & Prevention*, vol. 30, no. 2, pp. 207-215, 1998.
- [8] Douglas Gettman, Larry Head, "Surrogate Safety Measures from Traffic Simulation Models," *Siemens Gardner Transportation Systems, 6375 East Tanque Verde No. 170, Tucson, AZ 85715*, vol. 1840, no. 1, pp. 104-115, 2003.

- [9] HORST, RVD, NORTH, HOLLAND, "TIME-TO-COLLISION AS A CUE FOR DECISION-MAKING IN BRAKING," pp. 19-26, 1991.
- [10] L. Trà, "Tìm hiểu về hệ thống hỗ trợ lái xe ADAS," vinfastauto, 04 06 2021. [Online]. Available: https://vinfastauto.com/vn_vi/tim-hieu-ve-he-thong-ho-tro-lai-xe-adas. [Accessed 01 05 2022].
- [11] Aparicio, F., Arenas, B., Gómez, A., Jiménez, F., López, J.M.,, "Ingeniería del Transporte," Spanish, 2008.
- [12] Richard van der horst, Jeroen Hogema, "TIME-TO-COLLISION AND COLLISION AVOIDANCE SYSTEMS," 1994.
- [13] Teltonika, "FMC125 and a camera solution designed to avoid collisions and accidents," [Online]. Available: <https://teltonika-gps.com/product/adas/#description> . [Accessed 18 05 2022].
- [14] Advantech, "TREK-130," Advantech, [Online]. Available: https://www.advantech.com/products/1-2jsj5t/trek-130/mod_4662a759-4323-4ba2-a347-d31f34a5aff4. [Accessed 2022].
- [15] Elvik, R., Vaa,, " The handbook of road safety measures.," Elsevier, UK, 2004.
- [16] Allen, B. L., Shin, B. T. and Cooper, D. J, "Analysis of traffic conflicts and," Washington, D.C., 1978.
- [17] Jiwoong Choi, Dayoung Chun, Hyuk-Jae Lee, Hyun Kim, "Uncertainty-based Object Detector for Autonomous Driving Embedded Platforms," *2020 2nd IEEE International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2020.

PHỤ LỤC

Cấu hình phần cứng cho cảm biến radar: bộ thông số hoạt động tốt trong khoảng cách 0 đến 50[m]:

```
profileCfg 0 60.25 8 7 28 0 0 20 1 256 12500 0 0 48  
frameCfg 0 0 140 0 100 1 0  
cfarCfg -1 0 2 8 4 3 0 10 0  
cfarCfg -1 1 0 4 2 3 1 20 0  
aoaFovCfg -1 -60 60 0 20  
cfarFovCfg -1 0 2 75  
cfarFovCfg -1 1 -20 20  
boundaryBox -8 1 5 50 0 1  
allocationParam 200 100 0.05 5 16 20  
trackingCfg 1 2 500 25 200 50 100 90
```