การพัฒนาระบบการตัดสินใจและการวางแผนเส้นทางของรถอัตโนมัติเพื่อหลีกเลี่ยงการชนในสถานการณ์ที่มีคนเดินข้ามถนน

น.ส.วรดา ขวัญเจริญ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2565
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Development of Decision-making and Local Planning for Autonomous Vehicle to Mitigate Pedestrian Crash

Miss Worada Khwanjaroen

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Mechanical Engineering
Department of Mechanical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2022

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาระบบการตัดสินใจและการวางแผนเส้นทางของรถอัตโนมัติเพื่อหลีกเลี่ยงการชนในสถานการณ์ที่มีคนเดินข้ามถนน
โดย	น.ส.วรดา ขวัญเจริญ
สาขาวิชา	วิศวกรรมเครื่องกล
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.นักสิทธ์ นุ่มวงษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้นับวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(ผู้ช่วยศาสตราจารย์ ดร.สัณหพศ จันทรานุวัฒน์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(ผู้ช่วยศาสตราจารย์ ดร.นักสิทธ์ นุ่มวงษ์)

..... กรรมการ
(รองศาสตราจารย์ ดร.รัชทิน จันทร์เจริญ)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.ปาษาณ กุลวานิช)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วรดา ขวัญเจริญ : การพัฒนาระบบการตัดสินใจและการวางแผนเส้นทางของรถอัตโนมัติเพื่อหลีกเลี่ยงการชนในสถานการณ์ที่มีคนเดินข้ามถนน. (Development of Decision-making and Local Planning for Autonomous Vehicle to Mitigate Pedestrian Crash) อ.ที่ปรึกษาหลัก : ผศ. ดร.นักสิทธ์ นุ่มวงษ์

หากกล่าวถึงอุบัติเหตุบนท้องถนนที่เกิดขึ้นในประเทศไทยในปัจจุบัน ทั้งอุบัติเหตุระหว่างรถยนต์กับรถยนต์ รถยนต์กับรถจักรยานยนต์ และแม้กระทั่งอุบัติเหตุระหว่างคนเดินเท้ากับรถยนต์ ซึ่งอุบัติเหตุระหว่างคนเดินเท้าและรถยนต์ส่วนมากจะเกิดในกรณีที่ไม่ใช่ทางม้าลาย และมีความรุนแรงถึงชีวิต ในงานวิจัยนี้จึงเลือกเหตุการณ์นี้มาพิจารณา ซึ่งรถอัตโนมัติได้เข้ามามีบทบาทในการช่วยลดปัญหาของอุบัติเหตุที่เกิดขึ้นในหลายๆ แง่ ซึ่งรถอัตโนมัติกำลังเป็นที่สนใจของนักพัฒนาและนักวิจัยที่กำลังพัฒนากันอยู่อย่างแพร่หลาย ทั้งพัฒนาด้านการตัดสินใจของรถ ด้านการรับรู้ของรถ โดยในงานวิจัยนี้สนใจในส่วนของการตัดสินใจของรถอัตโนมัติในสถานการณ์การข้ามถนนระหว่างคนกับรถอัตโนมัติที่มีรถจอดบังอยู่ข้างทาง โดยใช้ค่า Time-To-Collision(TTC) และ Time-To-Brake(TTB) เป็นเกณฑ์ที่ช่วยตัดสินใจให้กับรถอัตโนมัติ หากค่า TTB น้อยกว่า TTC รถตัดสินใจที่จะเบรค ถ้าหาก TTB มากกว่า TTC รถตัดสินใจที่จะหักเลี้ยวหลบ โดยการพัฒนานี้จะทำการจำลองเหตุการณ์ใน Simulation และทำการทดลองจริงเพื่อยืนยันค่า Simulation ว่า Algorithm ที่เราคิดสามารถลดจำนวนการเกิดอุบัติเหตุและลดความเสี่ยงที่จะเกิดอุบัติเหตุในเหตุการณ์การข้ามถนนของคนกับรถอัตโนมัติ ที่คำนึงถึงผู้โดยสารที่อยู่บนรถอัตโนมัติด้วยได้จริง

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิศวกรรมเครื่องกล ลายมือชื่อนิสิต .....

ปีการศึกษา 2565 ลายมือชื่อ อ.ที่ปรึกษาหลัก .....

6372100421 : MAJOR MECHANICAL ENGINEERING

KEYWORD: Autonomous Vehicles, Pedestrian, Decision-making, Obstacle vehicle

Worada Khwanjaroen : Development of Decision-making and Local Planning for Autonomous Vehicle to Mitigate Pedestrian Crash. Advisor: Asst. Prof. NUKSIT NOOMWONGS

Nowadays, accidents that occur in Thailand include vehicle to vehicle, vehicle to motorcycle and pedestrian to vehicle. And accidents between pedestrians and vehicle can be life-threatening violence, most of such accidents occur on Jaywalk. Autonomous vehicle(AV) has come to reduce the accidents that occur. Many developer and researcher are interested to develop Autonomous vehicle in decision-making, perception or path planning too. In this paper we are interested to develop decision-making system of autonomous vehicle in scenario of crossing on jaywalk of pedestrian while there was an obstacle vehicle on the shoulder of the road, by using Time-To-Collision(TTC) and Time-To-Brake(TTB) are criteria of decision. If TTB is less than TTC, autonomous vehicle decides to brake, but TTB is more than TTC, autonomous vehicle decides to avoidance. In the results section, we show the simulation results that our algorithm can actually reduce the collision and the risk of such scenario. In the last section is the experiments section, part to confirm the results from simulation.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Field of Study: Mechanical Engineering Student's Signature .....

Academic Year: 2022 Advisor's Signature .....

กิตติกรรมประกาศ

ขอขอบคุณอาจารย์ที่ปรึกษา ผศ.ดร.นักสิทธ์ นุ่มวงษ์ ที่ให้คำปรึกษาเกี่ยวกับงานวิจัย และใช้รถอัตโนมัติระดับที่ 3 (Turing) ขอขอบคุณคณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ที่ให้สถานที่ทำการศึกษางานวิจัยจนสำเร็จลุล่วง

วรดา ขวัญเจริญ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ค

บทคัดย่อภาษาอังกฤษ ..... ง

กิตติกรรมประกาศ ..... จ

สารบัญ ..... ฉ

สารบัญตาราง ..... ฌ

สารบัญรูปภาพ ..... ญ

บทที่ 1 ที่มาและความสำคัญ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการวิจัย 2

1.3 ขอบเขตของการวิจัย 2

1.4 ระเบียบวิธีการวิจัย 3

1.5 แผนการดำเนินงานและผลลัพธ์แต่ละขั้นตอน 3

1.6 ประโยชน์ที่คาดว่าจะได้รับ 4

บทที่ 2 ปริทัศน์วรรณกรรม 5

2.1 การรับรู้ภายนอก (Sensing) 5

2.1.1 LiDAR (Laser Imaging Detection and Ranging) 5

2.1.2 Radar (Radio Detection and Ranging) 6

2.2 การคำนวณต่างๆ (Computing) 8

2.2.1 ระบบการรับรู้ (Perception system) 9

2.2.1.1 การติดตามตำแหน่ง (Localization) 9

2.2.1.2 การตรวจจับวัตถุ (Detection) 9

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2.1.3 การคาดการณ์เหตุการณ์ (Prediction) 10
2.1.2 ระบบการตัดสินใจ (Decision-making system) 10
2.1.2.1 การตัดสินใจโดยใช้ Machine Learning 11
2.1.2.2 การตัดสินใจโดยใช้ Rule-based 11
2.1.3 ระบบการวางแผน (Planning system) 18
2.1.3.1 การวางแผนภารกิจ (Mission planning) 18
2.1.3.2 การวางแผนการเคลื่อนที่ (Motion Planning) 18
2.3 การกระทำ (Actuation) 21
บทที่ 3 แนวคิดและการออกแบบยานพาหนะอัตโนมัติ 23
3.1 การออกแบบระบบการตัดสินใจ (Decision-making system) 23
3.1.1 TTC (Time-To-Collision) 24
3.1.1.1 $TTC_{av,ped}$ 24
3.1.1.2 $TTC_{av,hv1}$ 27
3.1.1.3 $TTC_{av,hv2}$ 27
3.1.2 TTB (Time-To-Brake) 28
3.2 การออกแบบ Motion planning 28
3.2.1 การชะลอความเร็ว (Slow down) 29
3.2.2 การเบรก (Braking) 30
3.2.3 การหักเลี้ยวหลบ (Steering) 31
3.3 ขั้นตอนการทำการทดลอง 32
3.2.1 การทำการทดลองใน Simulation 34
3.2.2 การทำการทดลองจริง (Experiment) 34
บทที่ 4 ผลการทดลอง 36
4.1 ผลการทดลองจาก Simulation 36

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.1.1 Baseline Simulation 36
4.1.2 With Algorithm (Without Steering) 38
4.1.3 With Algorithm (With Steering) 46
4.2 ผลการทดลองจากการทดลองจริง (Experiment) 57
บทที่ 5 สรุปผล และข้อเสนอแนะ 62
5.1 สรุปผลงานวิจัย 62
5.2 ข้อเสนอแนะ 63
ภาคผนวก 65
บรรณานุกรม 2
ประวัติผู้เขียน 5

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญตาราง

หน้า

ตารางที่ 1 ความแตกต่างของ Sensor ที่ใช้ตรวจจับวัตถุแต่ละชนิด 8
ตารางที่ 2 จำนวนเหตุการณ์ที่พิจารณาความเร็วและเวลาของคนและยานพาหนะ 12
ตารางที่ 3 จำนวนเหตุการณ์การใช้และไม่ใช้ AEB control 12
ตารางที่ 4 กรณีการชนระหว่างยานพาหนะ 2 คัน ที่ทำมุม Cross section น้อยกว่า 90 องศา [8] 15
ตารางที่ 5 อัตราหน่วงที่ใช้ในการชะลอความเร็ว 30
ตารางที่ 6 พารามิเตอร์ที่ใช้ในงานวิจัย 33
ตารางที่ 7 จำนวนเหตุการณ์ของแต่ละสถานการณ์ 34
ตารางที่ 8 จำนวนการเกิด Emergency braking และเกิดการชน (collision) ใน Baseline 36
ตารางที่ 9 จำนวนการเกิด Emergency braking และเกิดการชน (collision) ที่ใช้ Algorithm 39
ตารางที่ 10 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 30 กม/ชม 46
ตารางที่ 11 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 25 กม/ชม 50
ตารางที่ 12 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 20 กม/ชม 53
ตารางที่ 13 คุณสมบัติคอมพิวเตอร์ที่ใช้ 67

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

สารบัญรูปภาพ

หน้า

รูปที่ 1 ข้อมูลสัดส่วนผู้เสียชีวิตจากอุบัติเหตุบนท้องถนนจากองค์กรอนามัยโลก (WHO) 1

รูปที่ 2 แผนภาพการทำงานพื้นฐานของรถอัตโนมัติ 5

รูปที่ 3 ตัวอย่างการตรวจจับวัตถุจาก LiDAR sensor 6

รูปที่ 4 ตัวอย่างการตรวจจับวัตถุจาก Radar sensor 7

รูปที่ 5 ตัวอย่างการตรวจจับวัตถุกล้อง RGB sensor 8

รูปที่ 6 วิธีการติดตามวัตถุ (Track object) 10

รูปที่ 7 เหตุการณ์ที่พิจารณาในงานวิจัย [7] 12

รูปที่ 8 โอกาสการเกิดการชนของทั้ง 3 สถานการณ์ในงานวิจัย [7] 13

รูปที่ 9 ยานพาหนะ2 คัน เคลื่อนที่ทางตรงที่ทำมุม Cross section < 90 องศา 14

รูปที่ 10 ทิศทางการชนในแต่ละกรณีของยานพาหนะ 2 คัน 16

รูปที่ 11 เวลาที่ยานพาหนะใช้เพื่อเข้าและออกจาก Conflict area ในกรณี B 17

รูปที่ 12 เกณฑ์การขับขี่ของรถรูปแบบ Shuttle Bus 19

รูปที่ 13 การระบุตำแหน่งใน $s-\rho$ coordinate 20

รูปที่ 14 การสร้างเส้นทางใหม่ที่เป็นไปได้ (Generate path candidate) 20

รูปที่ 15 รูปทางเรขาคณิตที่ใช้ในการคำนวณของวิธี Pure Pursuit 21

รูปที่ 16 รูปทางเรขาคณิตที่ใช้ในการคำนวณของวิธี Stanley 22

รูปที่ 18 แผนภาพ Algorithm ที่งานวิจัยนี้นำเสนอ 23

รูปที่ 19 ยานพาหนะทั้งหมดที่งานวิจัยนี้พิจารณา 24

รูปที่ 20 การเรียกชื่อ Boundary Box ของคนเดินเท้า, รถอัตโนมัติ และ Conflict area 25

รูปที่ 21 กรณีการเกิดการชนในกรณีที่รถอัตโนมัติชนคนเดินเท้า และคนเดินเท้าชนรถอัตโนมัติ 25

รูปที่ 22 การชนกรณีรถอัตโนมัติชนคนเดินเท้า 26

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 23 กรณีที่คนเดินเท้าชนรถอัตโนมัติ 27
รูปที่ 24 Geometry ของรถอัตโนมัติขณะตรวจเจอวัตถุอยู่ข้างทาง (Obstacle object) 30
รูปที่ 25 พารามิเตอร์เกี่ยวกับการเปลี่ยนเลน 31
รูปที่ 26 สถานการณ์ที่งานวิจัยนี้นำมาพิจารณา 32
รูปที่ 27 รถอัตโนมัติที่ใช้ในการทดลอง (Turing) 35
รูปที่ 29 วัตถุบดบัง (Obstacle object) 35
รูปที่ 30 จำนวนการเกิด Emergency braking และการชน (Collision) 37
รูปที่ 31 ความเร็วขณะเกิดการชนของรถอัตโนมัติและคนเดินเท้า 38
รูปที่ 32 จำนวนเหตุการณ์ที่เกิด Emergency braking ของการใช้ Algorithm เปรียบเทียบกับ Baseline 39
รูปที่ 33 จำนวนเหตุการณ์ที่เกิดการชน (Collision) ของการใช้ Algorithm เปรียบเทียบกับ Baseline 40
รูปที่ 34 เหตุการณ์การชนในสถานการณ์ที่ III-1 41
รูปที่ 35 เหตุการณ์การชนในสถานการณ์ที่ III-2 42
รูปที่ 36 เหตุการณ์การชนในสถานการณ์ที่ III-3 44
รูปที่ 37 อัตราหน่วงที่ยานพาหนะ HV1 ใช้เพื่อรักษา Time Headway 2 วินาที 45
รูปที่ 38 ค่า $TTC_{AV,HV1}$ ที่รักษา Time Headway เป็น 2 วินาที 45
รูปที่ 39 ความเร็วของยานพาหนะ HV1 และ AV ที่รักษา Time Headway 2 วินาที 45
รูปที่ 40 เส้นทางการเปลี่ยนเลนของรถอัตโนมัติ 48
รูปที่ 41 Heading angle ของเส้นทางที่เปลี่ยนเลน 48
รูปที่ 42 ค่าอัตราเร่งและอัตราหน่วงในแนว Longitudinal 49
รูปที่ 43 ค่าอัตราเร่งและอัตราหน่วงในแนว Lateral 49
รูปที่ 44 เส้นทางการเปลี่ยนเลนของรถอัตโนมัติ 51
รูปที่ 45 Heading angle ของเส้นทางที่เปลี่ยนเลน 52

รูปที่ 46 ค่าอัตราเร่งและอัตราหน่วงในแนว Longitudinal 52

รูปที่ 47 ค่าอัตราเร่งและอัตราหน่วงในแนว Lateral 53

รูปที่ 48 การเบรกและหักเลี้ยวเพื่อเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติ 25 km/h ในสถานการณ์ที่ I 55

รูปที่ 49 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1 m/s ในสถานการณ์ที่ III-1 55

รูปที่ 50 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1.2 m/s ในสถานการณ์ที่ III-1 56

รูปที่ 51 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1.4 m/s ในสถานการณ์ที่ III-1 56

รูปที่ 52 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 2 m/s ในสถานการณ์ที่ III-2 57

รูปที่ 53 จำลองเหตุการณ์ที่ใช้ทำการทดลอง 57

รูปที่ 54 การระบุตำแหน่งและเตรียมเส้นทางบนรถอัตโนมัติ 58

รูปที่ 55 มุมมองบนรถอัตโนมัติที่เจอคนเดินเท้า (Pedestrian robot) 58

รูปที่ 56 การเคลื่อนที่ในแนว Longitudinal และ Lateral ของรถอัตโนมัติและคนเดินเท้าในสถานการณ์ที่ III-3 59

รูปที่ 57 ความเร็วที่กำหนดในระบบการตัดสินใจที่ออกแบบ และความเร็วที่รถอัตโนมัติใช้วิ่งจริง 60

รูปที่ 58 การหักเลี้ยวเพื่อเปลี่ยนเลนที่ความเร็วต้นของรถอัตโนมัติเป็น 25 km/h และคนเดินเท้าเป็น 1 m/s ในสถานการณ์ที่ III-3 61

รูปที่ 59 คอมพิวเตอร์เครื่องที่ 1 66

รูปที่ 60 คอมพิวเตอร์เครื่องที่ 2 66

รูปที่ 61 Code Python ที่ใช้จำลองเหตุการณ์ของ Algorithm 74

รูปที่ 62 Code Python ที่ใช้จำลองเหตุการณ์ระหว่างรถอัตโนมัติและยานพาหนะคันหลัง 76

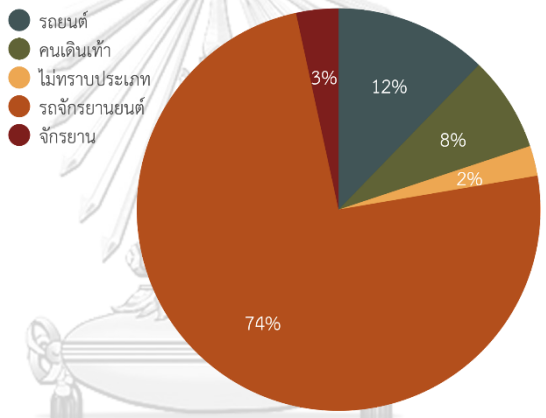
รูปที่ 63 Code Python ที่ใช้ TCP โปรโตคอลในการเชื่อมต่อ Pedestrian และรถอัตโนมัติ 79

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บทที่ 1 ที่มาและความสำคัญ

1.1 ความเป็นมาและความสำคัญของปัญหา

การใช้รถใช้ถนนในปัจจุบันมีผู้ใช้หลายประเภท ทั้งรถยนต์ส่วนบุคคล รถสาธารณะ และคนเดินเท้า ด้วยความหลากหลายของผู้ใช้รถใช้ถนนเหล่านี้ทำให้เกิดอุบัติเหตุบนท้องถนนในหลายรูปแบบ ข้อมูลจากองกรณ์อนามัยโลก (World Health Organization) [1] ปี 2018 พบว่าประเทศไทยมีอัตราการเสียชีวิตบนท้องถนนเป็นลำดับที่ 9 ของโลก มีอัตราการเสียชีวิต 32.7 ต่อประชากร 100,000 ราย ซึ่งการเสียชีวิตที่เป็นคนเดินเท้าเป็น 2.5 ต่อประชากร 100,000 ราย กราฟรูปที่ 1 แสดงถึงสัดส่วนผู้ใช้รถใช้ถนนที่เสียชีวิต ซึ่งอุบัติเหตุจากคนเดินเท้าคิดเป็นประมาณ 8% จากอัตราการเสียชีวิต เป็นประเภทผู้ใช้รถใช้ถนนที่เสียชีวิตเป็นลำดับที่ 3 รองจากรถจักรยานยนต์และรถยนต์



รูปที่ 1 ข้อมูลสัดส่วนผู้เสียชีวิตจากอุบัติเหตุบนท้องถนนจากองค์กรอนามัยโลก (WHO)

ซึ่งในงานวิจัยนี้สนใจเหตุการณ์อุบัติเหตุของคนเดินเท้า โดย [2] ได้สำรวจสาเหตุที่ทำให้เกิดอุบัติเหตุจากการข้ามถนนพบว่าการเกิดการชนคนเดินเท้าเกิดมากที่สุดในถนนทางตรงมากกว่าทางแยก ซึ่งเป็นการข้ามถนนของคนเดินเท้าที่ไม่ใช้ทางเท้า (Jaywaking) ซึ่งเกิดมาจากผู้ขับขี่ไม่ได้ตระหนักว่าจะมีคนเดินเท้าข้ามถนนเนื่องจากไม่ใช่ทางม้าลาย (Zebra-crossing) และผู้ขับขี่อาจสนใจอย่างอื่นอยู่อย่างเช่น หน้าจอวิทยุบนยานพาหนะ ดังนั้นเหตุการณ์การข้ามถนนบน Jaywalk ควรได้รับการแก้ไข

รถอัตโนมัติเริ่มเข้ามามีบทบาทในการช่วยลดอุบัติเหตุบนท้องถนนมากขึ้น ปัจจุบันมีนักพัฒนา และนักวิชาการร่วมมือกันเพื่อพัฒนาอย่างแพร่หลาย สมาคมวิศวกรรมยานยนต์ หรือ SAE (Society of Automotive Engineers) [3] ได้จัดระดับความสามารถในการขับขี่ไว้ 6 ระดับ ดังต่อไปนี้ ระดับ 0 ยานพาหนะที่ไม่มีระบบช่วยเหลือการขับขี่ใดๆ, ระดับที่ 1 ยานพาหนะมีระบบช่วยขับขี่บางอย่าง เช่น ระบบเตือนการชน, ระดับที่ 2 ยานพาหนะมีระบบช่วยเหลือการขับขี่มากขึ้น

เช่น ระบบ Adaptive Cruise Control ทำงานร่วมกับระบบ Lane Keeping Assistance ที่ทำงานร่วมกันอย่างน้อยสองระบบ, ระดับที่ 3 เป็นระดับที่ยานพาหนะสามารถขับขี่ได้เอง ในบางสภาวะตามที่มีการออกแบบไว้ (Conditional Automate Driving) ซึ่งยังต้องการมนุษย์คอยควบคุมอยู่ แต่สามารถทำกิจกรรมอื่นๆ ในระหว่างการขับขี่ได้ เนื่องจากหากรถอัตโนมัติเจอเหตุการณ์ที่ไม่สามารถตัดสินใจได้ หรือเหตุการณ์ที่ไม่เคยเจอมาก่อน ทำให้ไม่สามารถขับขี่ต่อไปได้ จึงจำเป็นต้องมีมนุษย์ช่วยตัดสินใจเมื่อเจอสถานการณ์นั้นๆ, ระดับที่ 4 ยานพาหนะมีความสามารถในการควบคุมการขับขี่เองได้ทั้งหมดแต่ยังต้องมีมนุษย์อยู่ภายในรถ แต่ไม่มีหน้าที่ในการควบคุมรถแต่อย่างใด และระดับที่ 5 ยานพาหนะสามารขับขี่อัตโนมัติโดยอิสระได้โดยไม่ต้องมีมนุษย์ควบคุมการขับขี่อยู่เลย ดังนั้นระบบการตัดสินใจ (Decision-making system) จึงเป็นส่วนสำคัญในการยกระดับความปลอดภัยให้กับรถอัตโนมัติ

ในงานวิจัยนี้จึงสนใจที่จะพัฒนาระบบช่วยการขับขี่ในส่วนของระบบการตัดสินใจเมื่อรถอัตโนมัติเจอคนเดินเท้าข้ามถนนหน้ายานพาหนะที่จอดอยู่ข้างทาง โดยเป็นการพัฒนารถอัตโนมัติระดับที่ 3 ในรูปแบบของรถรับส่งขับขี่อัตโนมัติความเร็วต่ำ (Level-3 Low-speed Autonomous Shuttle) โดยจะคำนึงถึงยานพาหนะที่ขับตามหลังรถอัตโนมัติและผู้โดยสารบนรถอัตโนมัติด้วย

1.2 วัตถุประสงค์ของการวิจัย

1. พัฒนาระบบการตัดสินใจของรถอัตโนมัติเมื่อมีความเสี่ยงที่จะเกิดอุบัติเหตุระหว่าง คนเดินเท้าที่จะข้ามถนนและรถอัตโนมัติ

2. หาเส้นทางที่เหมาะสมเพื่อหักเลี้ยวหลบในกรณีที่การเบรกไม่เพียงพอ เพื่อป้องกันการเกิดอุบัติเหตุโดยคำนึงถึงผู้โดยสารที่นั่งอยู่บนรถ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.3 ขอบเขตของการวิจัย

1. การทดลองเหตุการณ์จริงใช้รถ Turing เป็นยานพาหนะต้นแบบ โดยพัฒนาในด้าน Software เท่านั้น

2. การพัฒนาระบบการตัดสินใจของรถอัตโนมัติใช้ ไม่ได้ใช้ระบบการตรวจจับวัตถุจริง (Detection object) เพียงแต่เผื่อค่า Time delay ให้ระบบเท่านั้น

3. การหาเส้นทางที่เหมาะสมของรถอัตโนมัติในกรณีที่การเบรกไม่เพียงพอ สนใจในส่วนการเลือกเส้นทางเท่านั้น แต่ไม่ได้สนใจวิธีการสร้างเส้นทางใหม่

4. ทำการทดสอบในพื้นที่โล่ง และเป็นพื้นที่ปิด ในถนนทางตรงโดยใช้ Pedestrian Robot

แทนคนเดินเท้า รถอัตโนมัติมีความเร็วสูงสุด 30 กิโลเมตร/ชั่วโมง และ คนเดินเท้ามีความเร็วสูงสุด 2 เมตร/วินาที

5. พิจารณาความเร็วของคนเดินเท้าเป็นความเร็วคงที่เท่านั้น

1.4 ระเบียบวิธีการวิจัย

1. ศึกษางานวิจัยที่เกี่ยวข้องกับงานวิจัยนี้
2. ศึกษางานวิจัยที่เกี่ยวกับระบบการตัดสินใจ (Decision-making)
3. ออกแบบระบบ Decision-making ในการตัดสินใจเมื่อเจอคนเดินเท้าข้ามถนน
4. ศึกษางานวิจัยที่เกี่ยวข้องกับการหาเส้นทางใหม่เพื่อหลีกเลี่ยงการชน (Path Avoidance)
5. ออกแบบการเลือกเส้นทางการหลีกเลี่ยงการชน (Path Avoidance) ที่เหมาะสมในกรณีหักเลี้ยวหลบ
6. ศึกษางานวิจัยที่เกี่ยวข้องกับการติดตามเส้นทาง (Track Path) ที่จะนำมาปรับใช้กับรถอัตโนมัติให้เหมาะสม
7. ทดสอบระบบทั้งหมดที่ออกแบบใน Simulation และเลือกสถานการณ์ที่จะนำไปทำการทดลอง
8. ออกแบบการทดลองและทำการทดลองจริงในสถานที่ปิด
9. วิเคราะห์และสรุปผลการทดลอง

1.5 แผนการดำเนินงานและผลลัพธ์แต่ละขั้นตอน

ขั้นตอน	2565				2566					
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
รวบรวมข้อมูล และศึกษา งานวิจัยที่เกี่ยวข้อง										
ศึกษาวิธีใช้ Autoware, ROS										
พัฒนาระบบ Decision-making ในการตัดสินใจ เบรค และหักเลี้ยวหลบ										
ศึกษางานวิจัยที่เกี่ยวข้องกับการหาเส้นทางใหม่เพื่อหลีกเลี่ยงการชน										

ขั้นตอน	2565				2566					
	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.
(Path Avoidance)										
หาเส้นทางการหลีกเลี่ยงการชน (Path Avoidance) เมื่อระบบตัดสินใจหักเลี้ยวหลบ										
ศึกษางานวิจัยที่เกี่ยวข้องกับการติดตามเส้นทาง (Track Path)										
จำลองเหตุการณ์บน Simulation										
เตรียมการทดลอง ทดสอบระบบบน Autoware										
ทำการทดลองระบบที่ออกแบบทั้งหมด										
สรุปผล และทำรายงาน										

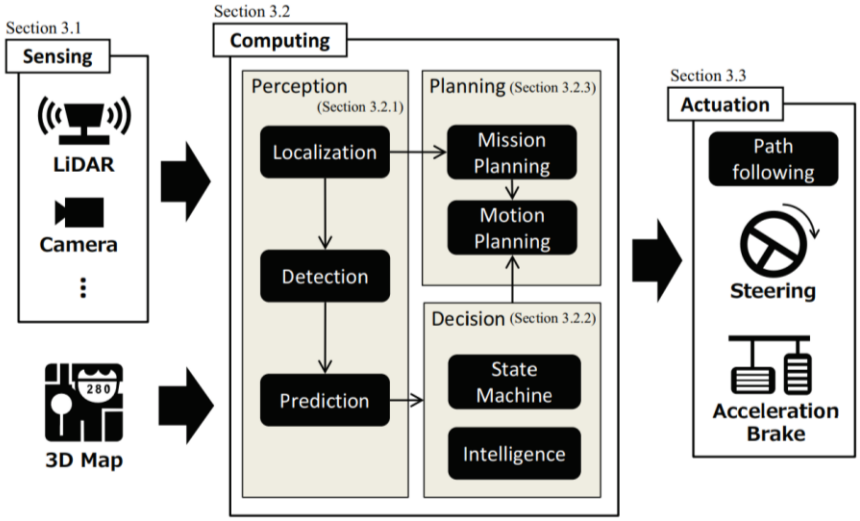
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1.6 ประโยชน์ที่คาดว่าจะได้รับ

1. ช่วยลดอุบัติเหตุระหว่างการข้ามถนนของคนเดินเท้าและรถอัตโนมัติได้ ในอนาคตเมื่อมีการใช้รถอัตโนมัติเพื่อรับส่งมนุษย์ หรือรับส่งวัสดุ ในมหาวิทยาลัย
2. ช่วยยกระดับความปลอดภัยในรถอัตโนมัติให้เป็น High level automation
3. ผู้คนที่ใช้ถนนรู้สึกปลอดภัยมากขึ้นเมื่อเจอกับรถอัตโนมัติ

บทที่ 2 ปริทัศน์วรรณกรรม

ระบบการทำงานพื้นฐานบนรถอัตโนมัติเป็นดังรูปที่ 2 ที่ [4] ได้สรุปไว้มี 3 ส่วนนั่นคือ 1.การรับรู้ภายนอก (Sensing) 2.การคำนวณต่างๆ (Computing) และ 3.การกระทำ (Actuation) ซึ่งมีรายละเอียดดังต่อไปนี้



รูปที่ 2 แผนภาพการทำงานพื้นฐานของรถอัตโนมัติ

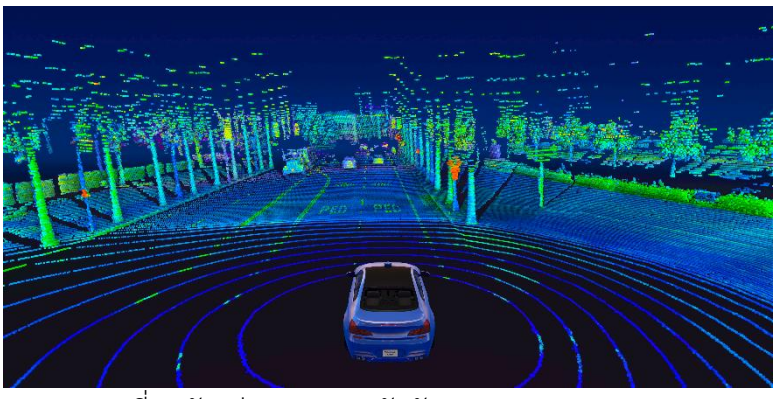
2.1 การรับรู้ภายนอก (Sensing)

รถอัตโนมัติใช้ Sensor ชนิดต่างๆ เพื่อรับข้อมูลเกี่ยวกับสภาพแวดล้อมภายนอก ช่วยให้รถอัตโนมัติสามารถรับรู้และเข้าใจสภาพแวดล้อมโดยรอบ ชนิดของ Sensor ที่ใช้ในการรับรู้ (Sensing) ในงานวิจัยนี้ จะยกตัวอย่างดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.1.1 LiDAR (Laser Imaging Detection and Ranging)

เป็น Sensor ที่ใช้แสง Laser เพื่อวัดระยะทางจากวัตถุที่อยู่ในระยะใกล้ถึงระยะไกล หลักการทำงานคือจะส่งสัญญาณ Laser ออกไปยังสิ่งแวดล้อมหลายทิศทางเพื่อเก็บข้อมูลในพื้นที่รอบตัวของรถอัตโนมัติ สัญญาณจะสะท้อนกลับมาจากวัตถุและวัดเวลาที่ใช้ในการเดินทางไปยังวัตถุและกลับมายัง Sensor ซึ่งข้อมูลที่ได้รับจากการส่งและรับสัญญาณจะถูกนำไปสู่กระบวนการประมวลผล และข้อมูลที่ได้จากการประมวลผลจะถูกนำมาสร้างเป็นภาพสามมิติที่แสดงตำแหน่งและระยะทางของวัตถุที่ตรวจจับได้ ภาพสามมิตินี้จะเป็นชุดของจุด 3 มิติ เรียกว่า Point cloud ดังรูปที่ 3



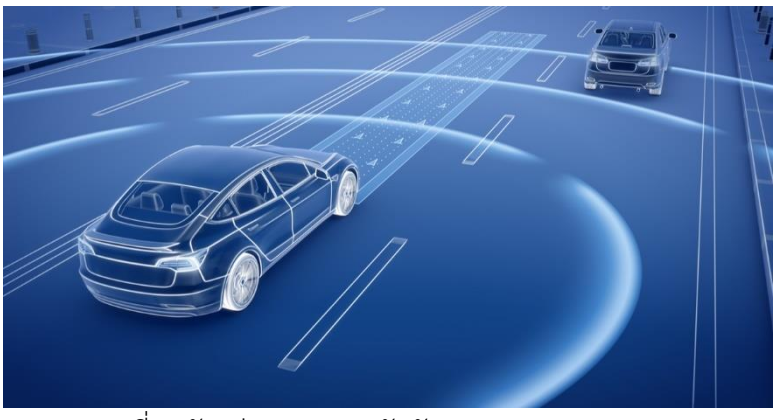
รูปที่ 3 ตัวอย่างการตรวจจับวัตถุจาก LiDAR sensor

ข้อดีของการใช้ LiDAR sensor คือสามารถตรวจจับวัตถุที่อยู่ในระยะใกล้และระยะไกลได้อย่างแม่นยำ และสามารถทำงานได้ภายในสภาพแวดล้อมที่ซับซ้อน เช่น สภาวะที่มีฝนตกหรือหมอกหนา แต่อาจมีราคาสูง และมีขนาดและน้ำหนักที่มากเมื่อเทียบกับ Sensor ชนิดอื่น โดย Lidar sensor แบ่งเป็น 4 ประเภท ตามหลักการทำงาน มีรายละเอียดดังต่อไปนี้

- Mechanical Scanning LiDAR: ใช้หลักการหมุนหรือการเคลื่อนที่ของอุปกรณ์ส่งสัญญาณแสง laser และตัวรับสัญญาณแสง laser เพื่อสแกนวัดระยะทาง ตัวอย่าง LiDAR ประเภทนี้ เช่น Velodyne VLP LiDAR หรือ Velodyne HDL LiDAR
- Solid-State LiDAR: ไม่ต้องการการเคลื่อนที่ แต่จะใช้ Shift sensor เพื่อส่งและรับสัญญาณแสง laser มีความแม่นยำต่ำกว่า Mechanical Scanning LiDAR
- Flash LiDAR: ใช้หลักการส่งสัญญาณแสงทั้งหมดไปในเวลาเดียวกันและวัดระยะทางด้วยการวัดเวลาที่สัญญาณสะท้อนกลับมา รูปแบบนี้มีความเร็วในการวัดระยะทางสูงและมีความแม่นยำสูง แต่ราคาที่สูงมาก
- Hybrid LiDAR: เป็นการรวมเอา LiDAR sensor ประเภทต่างๆ เข้าด้วยกัน เพื่อให้ได้ผลลัพธ์ที่เหมาะสมในการรับรู้และวัดระยะทาง รูปแบบนี้สามารถรวมคุณสมบัติเพื่อให้ได้ผลลัพธ์ที่แม่นยำและครอบคลุมกว่า

2.1.2 Radar (Radio Detection and Ranging)

เป็น Sensor ที่ใช้สัญญาณคลื่นในการตรวจจับวัตถุ หลักการคือ Radar จะส่งสัญญาณคลื่นและรับสัญญาณที่สะท้อนกลับมาจากวัตถุเพื่อวัดระยะทาง ข้อมูลที่ได้จาก Radar sensor จะเป็นรูปแบบ Point cloud เช่นเดียวกับ LiDAR รูปที่ 4 แสดงตัวอย่างจำลองการใช้ Radar sensor บนยานพาหนะ



รูปที่ 4 ตัวอย่างการตรวจจับวัตถุจาก Radar sensor

ข้อดีของ Radar sensor คือมีความแม่นยำสูงในการตรวจจับและวัดระยะทางของวัตถุ โดยสามารถระบุตำแหน่งและระยะทางได้อย่างแม่นยำได้ในระยะไกล สามารถใช้ได้ในสภาพแวดล้อมที่ซับซ้อน เช่น สภาพอากาศที่มีเมฆหรือฝนตก สามารถระบุความเร็วและทิศทางการเคลื่อนไหวของวัตถุได้ นิยมใช้ Radar sensor ทั้งในการบิน อวกาศ การจราจร แต่ตามมาด้วยราคาที่สูงมากเมื่อเทียบกับ Sensor ชนิดอื่นมากๆ

2.1.3 กล้อง (Camera)

กล้อง หรือ Camera sensor ที่ใช้ในการตรวจจับวัตถุมักเป็นกล้องดิจิตอลที่มีความละเอียดสูง เพื่อรับภาพและแปลงเป็นสัญญาณดิจิตอล ที่สามารถนำไปประมวลผลต่อไปเพื่อตรวจจับวัตถุ เซ็นเซอร์ของกล้องสามารถแบ่งออกเป็นหลายประเภทตามความสามารถและการใช้งานดังต่อไปนี้

- RGB Camera: เป็นกล้องที่สามารถรับภาพสีและสามารถแสดงผลเป็นรูปภาพสีได้ ใช้ในการตรวจจับวัตถุและการระบุคุณสมบัติต่างๆ ของวัตถุ
- Thermal Camera: เป็นกล้องที่ใช้ในการตรวจจับความร้อนของวัตถุ มีความสามารถในการระบุวัตถุที่มีอุณหภูมิสูงหรือต่างจากสภาพแวดล้อม
- Depth Camera: เป็นกล้องที่สามารถรับภาพสีพร้อมข้อมูลระยะทาง (depth) ของวัตถุ ใช้เทคโนโลยีเชิงลึก (depth sensing) เพื่อระบุระยะทางของวัตถุและสร้างภาพสามมิติ เป็นที่นิยมใช้บนรถอัตโนมัติ
- 3D Camera: เป็นกล้องที่สามารถรับภาพสามมิติและสร้างโมเดลสามมิติของวัตถุ ใช้ในการตรวจจับวัตถุและวัดความลึกในพื้นที่ต่าง ๆ เช่นการสแกนวัตถุเพื่อใช้ในงานอุตสาหกรรมหรือสร้างภาพสามมิติสำหรับการจัดแสดงผล นิยมใช้ในหุ่นยนต์ (Robotic) หรือรถอัตโนมัติ

การเลือกใช้ camera sensor ที่เหมาะสมขึ้นอยู่กับการใช้งานและความต้องการของระบบ เช่น ใช้ RGB camera เพื่อใช้เป็น Sensor ท้ายรถ ดังรูปที่ 5 ความละเอียดที่ต้องการ ระยะทางที่ต้องการตรวจจับ และภาพรวดเร็วหรือชัดเจนที่ต้องการ ซึ่งความสามารถของ Camera sensor ที่

มากกว่า LiDAR และ Radar sensor คือ Camera sensor สามารถจำแนกวัตถุ (Classified object) ได้ว่าวัตถุที่ตรวจจับเป็นวัตถุประเภทใด เช่น คน สัตว์ สิ่งของ เป็นต้น

รูปที่ 5 ตัวอย่างการตรวจจับวัตถุกล้อง RGB sensor

จากข้อมูล Sensor ทั้ง 3 ประเภท ตารางที่ 1 เป็นตารางที่สรุปความแตกต่างของ Sensor 3 ชนิด ในด้านความแม่นยำ Radar sensor LiDAR sensor และ Camera sensor มีความแม่นยำสูงสุดไปต่ำสุดตามลำดับ และความสามารถที่ Sensor แต่ละชนิดสามารถตรวจจับได้ พบว่า LiDAR sensor กับ Camera sensor มีระยะการตรวจจับได้ไกลเท่าๆ กัน ขึ้นกับชนิดของ LiDAR และ Camera sensor แต่ทั้ง 2 sensor มีระยะการตรวจจับที่ไกลกว่า Radar sensor ส่วนราคาของ Sensor LiDAR Radar และ Camera sensor ราคาสูงสุดไปต่ำสุดตามลำดับ ซึ่งการเลือกใช้ Sensor แต่ละชนิดควรพิจารณาให้เหมาะสมกับการใช้งาน

ตารางที่ 1 ความแตกต่างของ Sensor ที่ใช้ตรวจจับวัตถุแต่ละชนิด

Sensor	ความแม่นยำ	ระยะที่ตรวจจับได้	ราคา
Radar			
Lidar			
Camera			

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 การคำนวณต่างๆ (Computing)

ระบบการคำนวณ (Computing) ดังแผนภาพในรูปที่ 2 มีระบบย่อย 3 ระบบด้วยกันคือ 1. ระบบการรับรู้ (Perception) 2.ระบบการวางแผน (Planning) และ 3.ระบบการตัดสินใจ (Decision) โดยจะอธิบายรายละเอียดแต่ละระบบดังต่อไปนี้

2.2.1 ระบบการรับรู้ (Perception system)

เป็นระบบที่คำนวนตำแหน่งรถอัตโนมัติ ด้วยการติดตามตำแหน่ง (Localization) และเข้าใจวัตถุโดยรอบ ด้วยการตรวจจับวัตถุ (Detection) เพื่อนำไปคาดการณ์สถานการณ์ในอนาคตที่กำลังจะเกิดขึ้น (Prediction)

2.2.1.1 การติดตามตำแหน่ง (Localization)

เป็นกระบวนการในการระบุตำแหน่งของรถอัตโนมัติ โดยใช้ข้อมูลที่มีอยู่ เช่น ข้อมูลจากเซ็นเซอร์ หรือระบบนำทาง มีหลายวิธีที่ใช้ในการติดตามตำแหน่ง ตัวอย่างเช่น

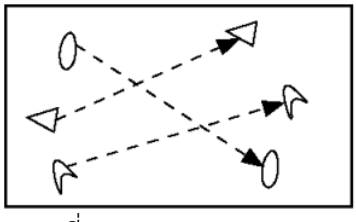
- วิธีการใช้ GNSS (Global Navigation Satellite System) เป็นการใช้สัญญาณจากดาวเทียมเพื่อระบุตำแหน่งทางภูมิศาสตร์ของรถอัตโนมัติ วิธีนี้มักใช้ในระบบนำทางในการติดตามยานพาหนะ
- วิธีการใช้ IMU (Inertial Measurement Unit) เป็นการใช้ IMU sensor เพื่อวัดและติดตามความเร่ง ความเร็ว และมุมเลี้ยวของรถอัตโนมัติในการเลี้ยวที่มี Roll และ Yaw angle

2.2.1.2 การตรวจจับวัตถุ (Detection)

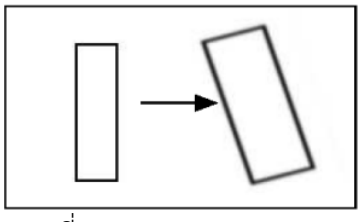
เป็นส่วนสำคัญในการคำนวณเนื่องจากเมื่อรถอัตโนมัติทราบข้อมูลจากการรับรู้ภายนอก (Sensing) ซึ่งการตรวจจับวัตถุจะมี 1.กระบวนการจำแนกวัตถุ (Classified object) ว่าเป็นวัตถุชนิดใด 2.กระบวนการติดตามวัตถุ (Track object) เพื่อให้ทราบว่าวัตถุนั้นมีความเร็วและระยะห่างจากรถอัตโนมัติเท่าใด โดยจะส่งข้อมูลเหล่านี้ให้กระบวนการคาดการณ์ (Prediction) เพื่อคาดการณ์ทิศทางการเลื่อนที่ของวัตถุนั้นๆ

การติดตามวัตถุมีด้วยกันหลายวิธี งานวิจัยของ [5] ได้รวบรวมวิธีการติดตามวัตถุ (Track object) ไว้เป็น 3 วิธี คือ 1.Point Tracking 2.Kernel Tracking และ 3.Silhouette Tracking โดยแต่ละวิธีมีรายละเอียดดังต่อไปนี้

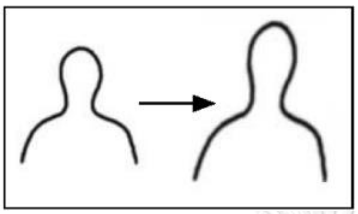
- Point Tracking เป็นการติดตามวัตถุแบบติดตามจุด จุดที่เลือกในภาพหนึ่งๆ จะถูกติดตามตามเวลาเพื่อระบุตำแหน่งของวัตถุในเฟรมต่อไป ดังรูปที่ 6a
- Kernel Tracking เป็นการติดตามวัตถุโดยใช้การคำนวณความคล้ายคลึงของภาพระหว่างภาพต้นฉบับและภาพปัจจุบัน ส่วนที่มีความคล้ายคลึงสูงสุดจะถูกใช้เป็นตำแหน่งของวัตถุในภาพปัจจุบัน ดังรูปที่ 6b
- Silhouette Tracking เป็นการประมาณค่าพื้นที่ของวัตถุในแต่ละเฟรม ถูกติดตามโดยการจับคู่รูปร่าง (shape matching) โดยใช้ค่าที่ได้จากเฟรมก่อนหน้านี้เป็นพื้นฐาน ดังรูปที่ 6c และ 6d



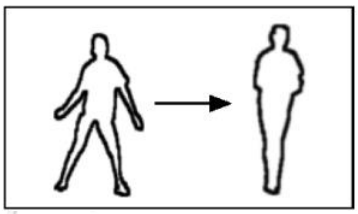
รูปที่ 6a Point Tracking



รูปที่ 6b Kernel Tracking



รูปที่ 6c Silhouette Tracking



รูปที่ 6d Silhouette Tracking

รูปที่ 6 วิธีการติดตามวัตถุ (Track object)

งานวิจัยในปี 2008 ของ [6] เสนอการติดตามวัตถุในวิธี Point Tracking โดยใช้ Velodyne HDL-64E LiDAR sensor ที่สามารถสแกนสิ่งแวดล้อมภายนอกที่ Frame rate 10 ครั้ง/วินาที อ่านข้อมูลได้ 1,000,000 ตำแหน่งต่อวินาที ในการติดตามวัตถุ (Track object) โดยใช้การประเมินค่าสถานะของยานพาหนะด้วย Bayes filter เพียงตัวเดียวต่อยานพาหนะแต่ละคัน พบว่าเวลาที่ใช้ในการจำแนกวัตถุ (Classified object) เป็น 25 มิลลิวินาที และเวลาที่ใช้ในการติดตามวัตถุ (Track object) เป็น 2 มิลลิวินาที โดยผลลัพธ์ของการติดตามวัตถุแม่นยำในระดับ 96-98% ในงานวิจัยนี้จึงนำค่าเวลาทั้ง 2 ค่านี้มาพิจารณาในการเลือก Time delay ของระบบการตรวจจับวัตถุ (Detection system)

2.2.1.3 การคาดการณ์เหตุการณ์ (Prediction)

การคาดการณ์เหตุการณ์ในอนาคต ขึ้นอยู่กับสถานการณ์นั้นๆ ว่ารถอัตโนมัติเจอเหตุการณ์อะไร เช่น เจอคนเดินเท้าที่อยู่บนถนน ต้องการการคาดการณ์ (Prediction) ว่าคนเดินเท้านั้นจะข้ามถนนหรือไม่ หากคาดการณ์ว่าคนเดินเท้าจะข้ามถนน ผลลัพธ์ของการคาดการณ์จะส่งข้อมูลไปบอกระบบการตัดสินใจ (Decision-making system) เพื่อให้ระบบการตัดสินในประมวลผลต่อว่ารถอัตโนมัติควรจะกระทำอย่างไรต่อ (Action)

2.1.2 ระบบการตัดสินใจ (Decision-making system)

เป็นระบบสำคัญ เนื่องจากทำหน้าที่ตัดสินใจเมื่อเจอเหตุการณ์ต่างๆ หรือกล่าวได้ว่าเป็นเสมือนสมองของผู้ขับขี่ เมื่อระบบการรับรู้ (Perception) ส่งข้อมูลมาให้ระบบการตัดสินใจ (Decision-making) เพื่อตัดสินใจที่จะทำบางอย่างบนรถอัตโนมัติ อย่างเช่น เบรกเมื่อเจอวัตถุที่กำลังเคลื่อนที่เข้ามาในถนน เปลี่ยนเลนเมื่อเจอการทำถนนขณะที่รถอัตโนมัติกำลังเคลื่อนที่ไป โดยวิธีการ

ตัดสินใจ (Decision) แบ่งออกเป็น 2 วิธีคือ 1.วิธีการตัดสินใจโดยใช้ Machine Learning และ 2. วิธีการตัดสินใจโดยใช้ Rule-Based ซึ่งแต่ละวิธีมีรายละเอียดดังต่อไปนี้

2.1.2.1 การตัดสินใจโดยใช้ Machine Learning

การตัดสินใจโดยใช้ Machine Learning มีขั้นตอนหลักที่ประกอบไปด้วยการเตรียมข้อมูล (Data Preparation) โดยสำรวจข้อมูลเพื่อให้เข้าใจคุณลักษณะของข้อมูลที่ใช้ในการตัดสินใจ จัดรูปแบบข้อมูลเพื่อปรับปรุงและจัดการข้อมูลให้เหมาะสมเพื่อนำไปใช้ต่อ และจัดแบ่งข้อมูลเพื่อเป็นข้อมูลชุดที่จะนำไปฝึก (Training set) ชุดข้อมูลนำไปทดสอบ (Test set) เพื่อประเมินประสิทธิภาพของโมเดลที่ใช้ฝึก ขั้นตอนต่อมาคือการเลือกและเทรนโมเดล (Model selection and Training) มีโมเดลที่ใช้เพื่อเทรนข้อมูลอยู่หลายโมเดล การเลือกและกำหนดโมเดลการเรียนรู้ต้องเลือกให้เหมาะสมกับปัญหา เช่น อัลกอริทึมที่ใช้ในการจัดกลุ่ม (Clustering) การจำแนก (Classification) หรือการทำนาย (Prediction) และขั้นตอนสุดท้ายคือการใช้ชุดข้อมูลทดสอบ เพื่อวัดประสิทธิภาพของโมเดลที่นำมาเทรน และความแม่นยำในการตัดสินใจ เพื่อนำไปปรับปรุงโมเดลให้มีประสิทธิภาพมากขึ้น เช่น ปรับโครงสร้างโมเดล ปรับพารามิเตอร์ เป็นต้น

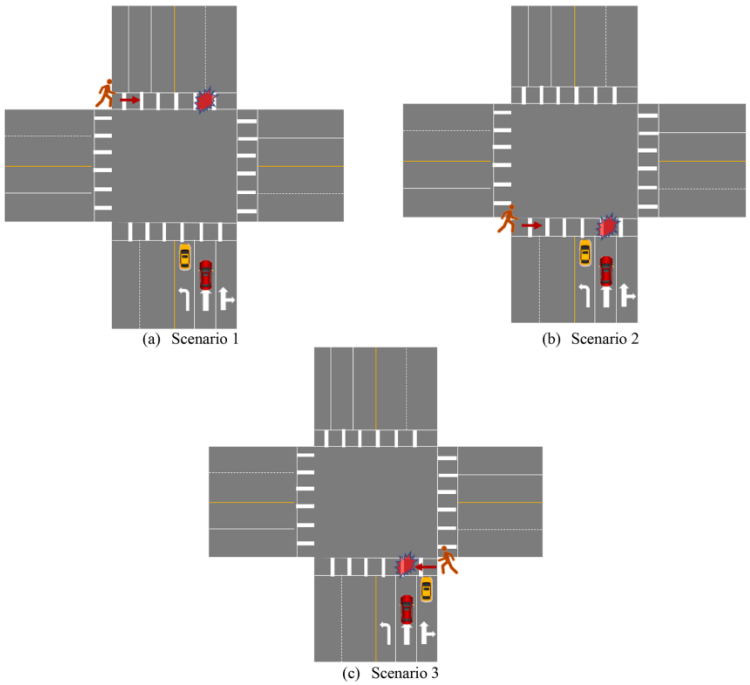
ในงานวิจัยนี้ ไม่ได้ใช้วิธีการตัดสินใจโดยใช้ Machine Learning เนื่องจากระยะเวลาในการเทรนข้อมูล ต้องใช้ระยะเวลาพอสมควร และไม่สามารถใช้รถอัตโนมัติไปวิ่งเพื่อเทรนข้อมูลในช่วงเวลาที่มีคนเดินเท้าข้ามถนนได้ เนื่องด้วยการคำนึงถึงความปลอดภัย ยานพาหนะยังไม่ได้มีการเทสระบบความปลอดภัยที่เพียงพอ

2.1.2.2 การตัดสินใจโดยใช้ Rule-based

การตัดสินใจโดยใช้กฎ (Rule-Based) คือมีการกำหนดกฎหรือเงื่อนไขที่ถูกต้องตามหลักการในระบบ และระบุการกระทำที่เกี่ยวข้องกับกฎแต่ละกฎ ตัวอย่างเช่น มีกฎว่าเจอไฟแดงให้เบรก เจอไฟเขียวให้ไป เมื่อรถอัตโนมัติได้รับข้อมูลจาก Sensor ว่าเจอสัญญาณไฟแดง ก็จะเบรก และไปต่อเมื่อเจอสัญญาณไฟเขียวตามกฎ (Rule-Based) ที่ตั้งไว้ โดยจะยกตัวอย่างงานวิจัยที่ใช้กฎ (Rule-Based) ในการตัดสินใจ ดังต่อไปนี้

งานวิจัยของ [7] นำเสนอการใช้ AEB control โดยเพิ่มค่า TTC threshold ในการตัดสินใจใช้ Automated Emergency Braking (AEB) เพื่อลดอุบัติเหตุระหว่างยานพาหนะและคนเดินเท้าในสถานการณ์ที่มีวัตถุบดบังดังรูปที่ 7 โดยรูปที่ 7a และ 7b เป็นสถานการณ์ที่มีวัตถุบดบังด้านซ้ายของยานพาหนะ แต่ตำแหน่งการข้ามถนนของคนเดินเท้าต่างกัน รูปที่ 7c วัตถุบดบังอยู่ด้านขวาของยานพาหนะ โดยใช้ LiDAR sensor ที่มีระยะตรวจจับ 300 ฟุต หรือประมาณ 91 เมตร จำนวนการทำการทดลองเป็นดังตารางที่ 2 แสดงจำนวนเหตุการณ์ทั้งหมดที่พิจารณาความเร็วคนเดินเท้า ความเร็วยานพาหนะ และเวลาที่คนเดินเท้าจะข้ามหลังที่ยานพาหนะเคลื่อนที่มาถึงจุดที่สนใจ ทั้งหมดเหตุการณ์ละ 216 เหตุการณ์ และตารางที่ 3 แสดงจำนวนเหตุการณ์ทั้งหมดที่พิจารณาเหตุการณ์ที่ใช้

AEB control และไม่ใช้ AEB control ทั้งหมด 16 เหตุการณ์ รวมทั้งหมดทำการทดลอง 10,368 เหตุการณ์ (3 สถานการณ์ x 216 เหตุการณ์ที่พิจารณาความเร็ว x 16 เหตุการณ์ที่พิจารณา AEB control)



รูปที่ 7 เหตุการณ์ที่พิจารณาในงานวิจัย [7]

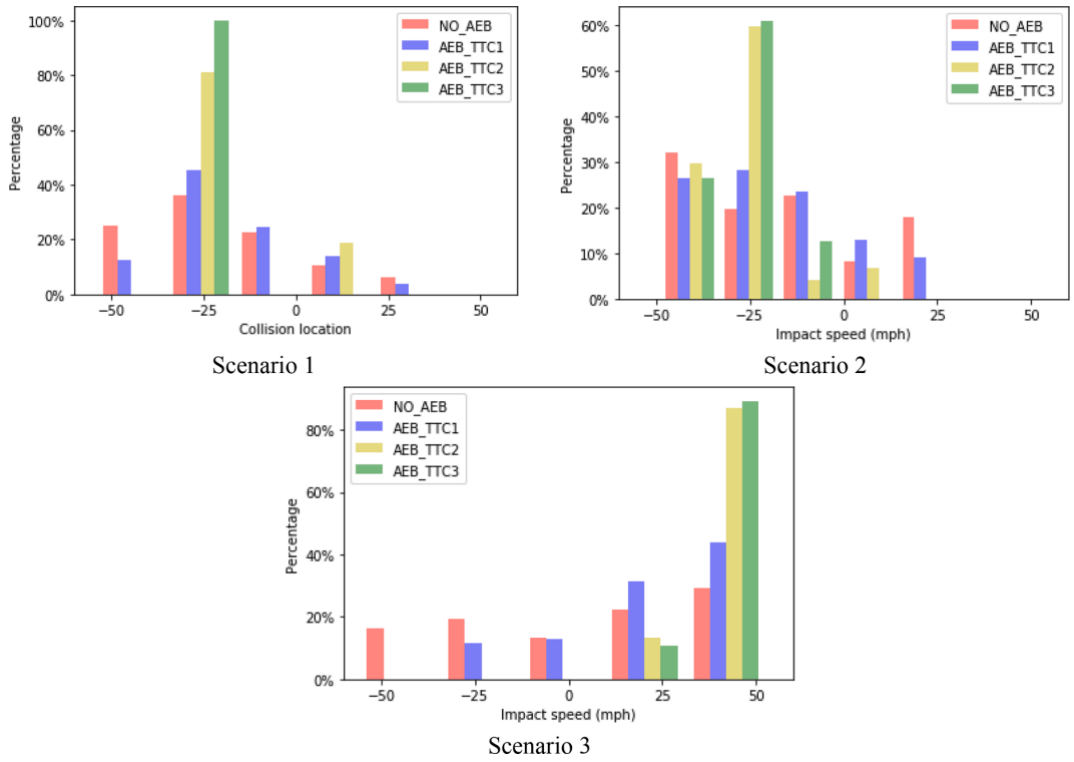
ตารางที่ 2 จำนวนเหตุการณ์ที่พิจารณาความเร็วและเวลาของคนและยานพาหนะ

พารามิเตอร์	Value	Step size	จำนวน
ความเร็วคนเดินเท้า (เมตร/วินาที)	0.6-3.6	0.6	6
ความเร็วยานพาหนะที่สนใจ (กม/ชม)	40-80	1.4	6
เวลาคนเดินเท้าเดินหลังจากยานพาหนะถึงจุดสนใจ	1-6	1	6
รวม	-	-	216

ตารางที่ 3 จำนวนเหตุการณ์การใช้และไม่ใช้ AEB control

พารามิเตอร์		Value	Step size	จำนวน
AEB control	FOV sensor (องศา)	60-180	30	5
	TTC threshold	1-3	1	3
ไม่มี AEB control		-	-	1
รวม		-	-	16

ผลการทดลองของงานวิจัย [7] พบว่าการใช้ AEB control สามารถลดจำนวนการเกิดการชนได้ (Collision) และยิ่งใช้ค่า TTC threshold เพื่อให้ AEB ทำงานเร็ว อย่างการใช้ค่า TTC threshold เป็น 3 วินาที (หมายความว่าเมื่อใดก็ตามที่ยานพาหนะคำนวณค่า TTC ระหว่างยานพาหนะกับคนเดินเท้าได้น้อยกว่า 3 วินาที AEB จะทำงานทันที) ช่วยลดการชนได้มากกว่าการใช้ TTC threshold เป็น 1 วินาที ดังกราฟรูปที่ 8



รูปที่ 8 โอกาสการเกิดการชนของทั้ง 3 สถานการณ์ในงานวิจัย [7]

CHULALONGKORN UNIVERSITY

ตัวแปรที่ในงานวิจัยส่วนใหญ่นำมาพิจารณาในการตัดสินใจคือค่า Time-To-Collision (TTC) และค่า Time-To-Brake (TTB) ซึ่งมีรายละเอียดดังต่อไปนี้

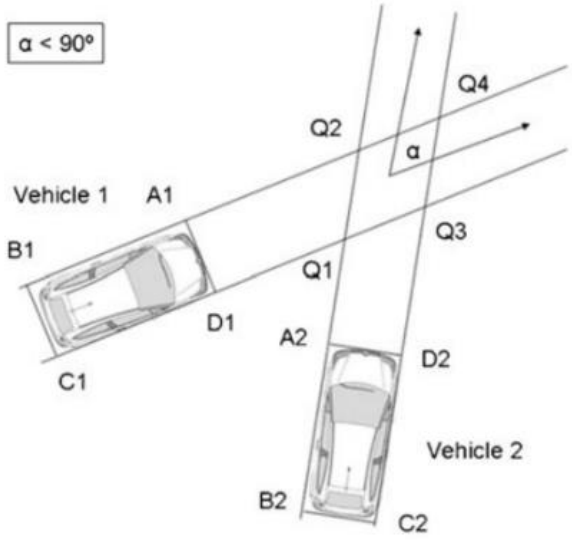
TTC (Time-To-Collision) หมายถึงเวลาที่เหลืออยู่ก่อนที่วัตถุจะชนหรือเกิดการชน หากค่า TTC ยิ่งมากสถานการณ์จะยิ่งปลอดภัย ในการเคลื่อนที่ทิศทางเดียวกันสามารถคำนวณค่า TTC ได้จากความเร็วและระยะห่างระหว่างยานพาหนะที่สนใจและยานพาหนะคันหน้าได้จากสมการที่ 1

$$\mathrm{TTC_i} = \frac{\mathrm{X_{i-1}(t) - X_i(t) - l_{i-1}}}{\mathrm{V_i(t) - V_{i-1}(t)}} \qquad \forall X_i(t) > X_{i-1}(t) \qquad (1)$$

โดยที่ i-1 เป็นข้อมูลของยานพาหนะคันหน้าที่เวลาใดๆ, i เป็นข้อมูลของยานพาหนะที่เราสนใจที่เวลาใดๆ, X คือตำแหน่งของยานพาหนะ, V คือความเร็วของยานพาหะนะ และ l คือความยาวของยานพาหนะ แต่ในสถานการณ์ทางร่วมทางแยกไม่สามารถใช้สมการที่ 1 ในการหา Time-To-Collision ได้เนื่องจากไม่ได้เคลื่อนที่ไปในทิศทางเดียวกัน

งานวิจัยของ [8] เสนอวิธีการหาค่า TTC ระหว่างยานพาหนะ 2 คันที่เคลื่อนที่ทำมุม Cross section น้อยกว่า 90 องศาดังรูปที่ 9 มีการกำหนดชื่อมุมของยานพหานะทั้ง 2 คัน คนที่1 กำหนดเป็นมุม A1 B1 C1 และ D1 คันที่ 2 กำหนดเป็น A2 B2 C2 และ D2 เมื่อลากเส้นตรงของ Boundary ของยานพาหนะแต่ละคันจะเกิดพื้นที่สี่เหลี่ยมตัดกันนั่นคือพื้นที่เกิดการชน (Conflict area) โดยตั้งชื่อมุมแต่ละมุมว่า Q1 Q2 Q3 และ Q4

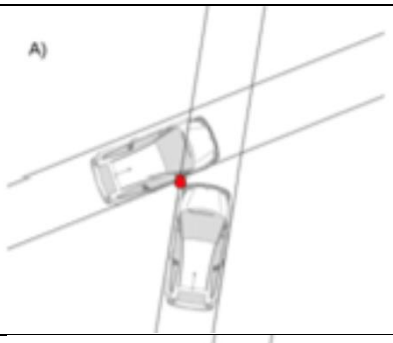
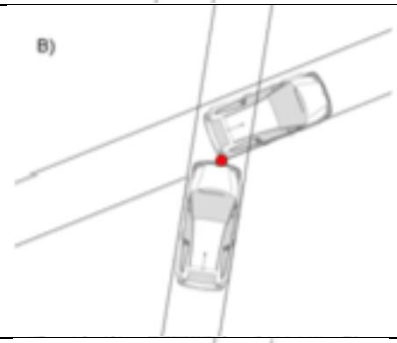
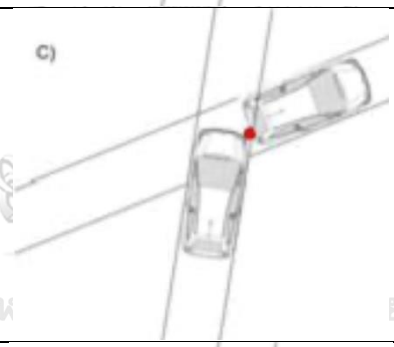
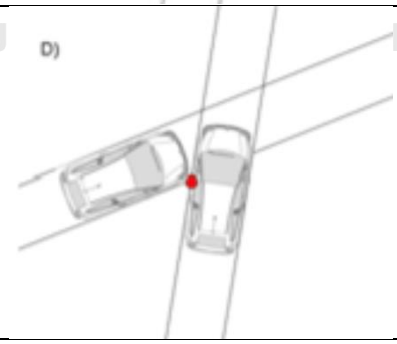
งานวิจัย [8] นำเวลาที่แต่ละมุมใช้เข้าและออกจาก Conflict area มาพิจารณาเพื่อหาเงื่อนไขการชนที่เกิดขึ้น และนำไปหาค่า TTC ที่ทำให้เกิดการชน โดยกำหนดการเรียกชื่อเวลาไว้ยกตัวอย่างดังนี้ TA21 คือเวลาที่มุม A ของยานพาหนะคันที่ 2 ใช้เพื่อเข้า Conflict area ที่ Q1 หรือ TC13 คือเวลาที่มุม C ของยานพาหนะคันที่ 1 ใช้เพื่อออกจาก Conflict area ที่ Q3 เป็นต้น

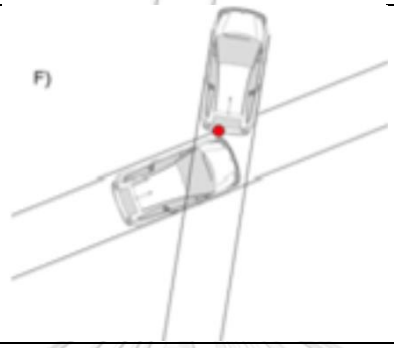


รูปที่ 9 ยานพาหนะ2 คัน เคลื่อนที่ทางตรงที่ทำมุม Cross section < 90 องศา

จากรูปที่ 9 การชนทั้งหมดที่เกิดขึ้นได้มีทั้งหมด 16 กรณี (คือมุม 4 มุมของยานพหานะทั้ง 2 คันคูณกันได้ 16 กรณี) แต่ในความเป็นจริงการชนที่สามารถเกิดขึ้นได้มีทั้งหมด 6 กรณี ดังตารางที่ 4 ที่แสดงถึงภาพจำลองในการเกิดอุบัติเหตุ และบอกถึงเงื่อนไขของการเกิดอุบัติเหตุของแต่ละกรณี

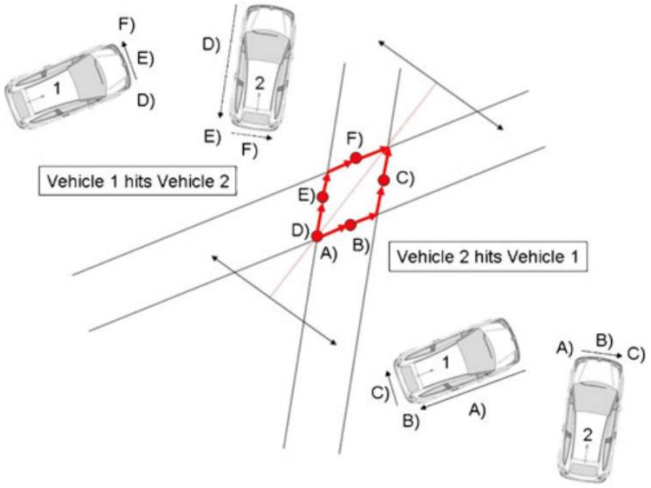
ตารางที่ 4 กรณีการชนระหว่างยานพาหนะ 2 คัน ที่ทำมุม Cross section น้อยกว่า 90 องศา [8]

	เหตุการณ์	ภาพจำลอง	เงื่อนไขการเกิดการชน
A)	มุมซ้ายของยานพาหนะคันที่ 2 ชนด้านข้างยานพาหนะคันที่ 1		TC11 > TA21 > TD11
B)	ด้านหน้ายานพาหนะคันที่ 2 ชนมุมท้ายรถยานพาหนะคันที่ 1		TC13 > TD23 > TA21 > TC11
C)	มุมขวายานพาหนะคันที่ 2 ชนท้ายยานพาหนะคันที่ 1		TB14 > TD24 > TD23 > TC13
D)	มุมขวายานพาหนะคันที่ 1 ชนด้านข้างยานพาหนะคันที่ 2		TB21 > TD11 > TA21

E)	มุมด้านซ้าย ยานพาหนะคันที่ 1 ชนด้านท้าย ยานพาหนะคันที่ 2		TB22 > TA12 > TD11 > TB21
F)	มุมด้านซ้ายของ ยานพาหนะคันที่ 1 ชนด้านท้าย ยานพาหนะคันที่ 2		TC24 > TA14 > TA12 > TB22

การชนในแต่ละเหตุการณ์จะเกิดบน Conflict area ทั้ง 4 ด้านตามลักษณะของการชน ยกตัวอย่างการชนในรูปแบบ B ด้านหน้ายานพาหนะคันที่ 1 ไปชนท้ายยานพาหนะคันที่ 2 ดังนั้น บริเวณที่สามารถเกิดการชนในรูปแบบ B คือเวกเตอร์ B ในรูปที่ 10 ซึ่งจะนำเวลาแรกที่ยานพาหนะถึง Conflict area (s=0) และเวลาที่ด้านในการเกิดอุบัติเหตุของยานพาหนะแต่ละคันออกจาก Conflict area (s=D) ในตัวอย่าง B บริเวณที่เกิดการชนอยู่ระหว่าง Q1-Q3 ดังนั้นจะพิจารณาเวลาที่ยานพาหนะใช้เพื่อไปถึง Q1 และออกจาก Q3 ได้ดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย

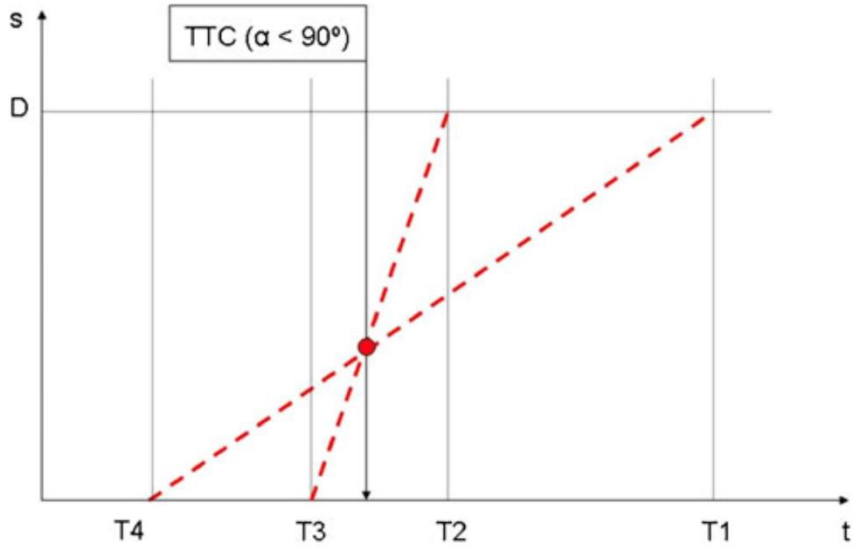


รูปที่ 10 ทิศทางการชนในแต่ละกรณีของยานพาหนะ 2 คัน

- ยานพาหนะคันที่ 1 : ถูกชนที่ด้านท้ายคือมุม C1 ดังนั้นจึงพิจารณาเฉพาะมุม C1 เวลาที่มุม C1 ใช้เพื่อเข้า Conflict area Q1 (s=0) คือ TC11 และเวลาที่มุม C1 ใช้เพื่อออก Conflict area Q3 (s=D) คือ TC13
- ยานพาหนะคันที่ 2 : ชนคันที่ 1 ที่ด้านหน้า ดังนั้นจะพิจารณาเฉพาะด้านหน้าคือมุม A2 และ B2 เวลาที่มุม A2 ใช้เพื่อเข้า Conflict area Q1 (s=0) คือ TA21 และเวลาที่มุม B2 ใช้เพื่อเข้า Conflict area (s=D) คือ TD23

ซึ่งหากเงื่อนไข TC13 > TD23 > TA21 > TC11 เป็นจริง นั่นคือขณะที่ท้ายของยานพาหนะคันที่ 1 อยู่ใน Conflict area ด้านหน้าของยานพาหนะคันที่ 2 ก็อยู่ใน Conflict area ด้วยเช่นกัน ดังนั้นจึงทำให้เกิดการชนในรูปแบบ B เมื่อนำเวลาทั้ง 4 ค่าไปพลอตกราฟโดยที่ T1, T2, T3 และ T4 มีค่าเรียงลำดับจากมากไปน้อย จะได้ดังรูปที่ 11 จะพบจุดตัดของเส้นตรงทั้งสองเส้น นั่นคือค่า TTC ที่ทำให้เกิดการชน ดังนั้นสมการการหาค่า TTC ในถนนที่มี Cross section น้อยกว่า 90 องศา สามารถคำนวณจากสมการเส้นตรง 2 เส้นที่ตัดกัน แล้วหาพิกัด t ของจุดตัดที่เกิดขึ้นจะได้ค่า TTC ดังสมการที่ 2

$$\mathrm{TTC} = \frac{\mathrm{T1T3 - T2T4}}{\mathrm{T1 + T3 - T2 - T4}} \qquad (2)$$



รูปที่ 11 เวลาที่ยานพาหนะใช้เพื่อเข้าและออกจาก Conflict area ในกรณี B

TTB (Time-To-Brake) เป็นค่าเวลาที่เหลืออยู่ก่อนที่ยานพาหนะจะหยุดสนิท หรือหยุดการเคลื่อนที่จากความเร็วปัจจุบัน ในระบบการตัดสินใจที่ใช้ระบบเบรก ค่า TTB จะถูกใช้เพื่อประมาณเวลาที่รถจะใช้เพื่อหยุดหรือลดความเร็วก่อนที่จะชนกับวัตถุหรือสิ่งกีดขวางใดๆ ในการขับขี่อัตโนมัติ โดยสามารถคำนวณได้จากสมการที่ 3 โดยที่ v(t) เป็นความเร็วขณะนั้นของยานพาหนะ และ a คืออัตราหน่วงที่ใช้ในการเบรก

$$\mathrm{TTB} = -\frac{\mathrm{v(t)}}{\mathrm{a}} \qquad (3)$$

หากนำค่า TTC และ TTB มาใช้ตัดสินใจ กล่าวได้ว่าหากค่า TTB < TTC นั่นคือยานพาหนะสามารถเบรกได้สนิท (ในเงื่อนไขอัตราหน่วงที่ยานพหานะทำได้จริง) ก่อนที่จะชน แต่ถ้าหาก TTB > TTC นั่นคือยานพหานะเบรกไม่อยู่ และเกิดการชนเกิดขึ้น

2.1.3 ระบบการวางแผน (Planning system)

เป็นระบบในการวางแผนเส้นทางเมื่อรถอัตโนมัติทราบตำแหน่งเริ่มต้น และตำแหน่งปลายทาง ในระบบการวางแผนเส้นทางจะแบ่งออกเป็น 2 ระบบการวางแผน ดังต่อไปนี้

2.1.3.1 การวางแผนภารกิจ (Mission planning)

เป็นระบบที่รถอัตโนมัติใช้เพื่อเลือกเส้นทางในการเดินทางไปยังจุดเป้าหมาย ปัจจัยในการเลือกเส้นทางประกอบไปด้วย เวลาในการเดินทาง ความปลอดภัยในแต่ละเส้นทาง กฎหมายหรือกฎจราจรในแต่ละเส้นทาง เช่น ถนนที่เป็นทางเดินรถทางเดียว (One way) เป็นต้น

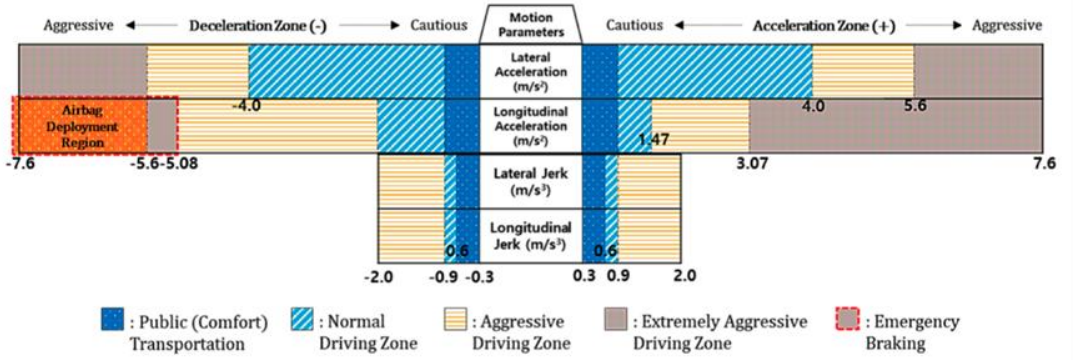
2.1.3.2 การวางแผนการเคลื่อนที่ (Motion Planning)

เมื่อทราบเส้นทางที่ต้องเดินทางไปยังจุดหมายจากระบบการวางแผนภารกิจ (Mission planning) แล้ว และรับข้อมูลจากระบบการตัดสินใจ (Decision-making system) หากต้องมีการเปลี่ยนเส้นทางเพื่อหลีกเลี่ยงการชน เช่น การเปลี่ยนเลน การเบรกให้วัตถุเคลื่อนที่ผ่านหน้ารถอัตโนมัติไปก่อนแล้วค่อยเดินทางต่อ ซึ่งเส้นทางที่ออกแบบมาใหม่ต้องคำนึงถึงความปลอดภัยเป็นอันดับแรก ยานพาหนะรูปแบบ Bus ที่ใช้รับ-ส่ง ผู้โดยสารจากจุดหนึ่งเพื่อไปส่งที่หมายนั้นเรียกว่ารูปแบบ Shuttle Bus ซึ่งในระหว่างการขับขี่ตลอดเส้นทาง การขับขี่ของยานพาหนะส่งผลต่อความรู้สึกของผู้โดยสารตลอดเวลา กล่าวคือหากยานพาหนะขับด้วยการขับขี่ที่นุ่มนวลผู้โดยสารจะไม่รู้สึกกังวลใดๆ แต่หากเมื่อใดก็ตามที่ยานพาหนะเบรกด้วยอัตราหน่วงสูงๆ จะทำให้ผู้โดยสารบนรถเกิดความกังวลใจขึ้น ดังนั้นความสำคัญของเกณฑ์การขับขี่สำหรับรถอัตโนมัติที่มีผลต่อผู้โดยสารบนรถมีมุมมองของความปลอดภัยและความสะดวกสบายของผู้โดยสาร มีรายละเอียดดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

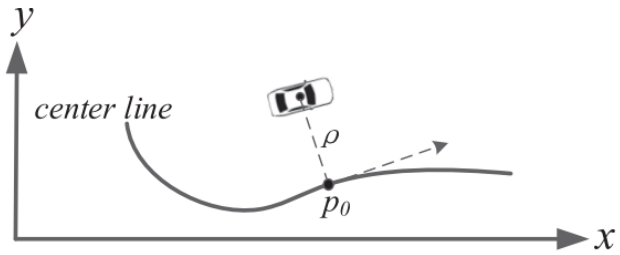
- ความปลอดภัย : ระบบการขับขี่ควรมีความแม่นยำในการตรวจจับแหละหลีกเลี่ยงสิ่งกีดขวาง รวมถึงการรักษาระยะห่างที่เหมาะสมกับยานพาหนะและวัตถุในสภาพแวดล้อม เพื่อลดความเสี่ยงในการเกิดอุบัติเหตุ
- ความสะดวกสบาย : ควรเน้นความสะดวกสบายของผู้โดยสาร ระบบควบคุมความเร็วและการเบรกควรทำงานอย่างราบรื่นและไม่ก่อให้เกิดการเคลื่อนที่ที่รุนแรง ซึ่งอาจส่งผลให้ผู้โดยสารรู้สึกไม่สบายหรือไม่สะดวกในระหว่างการเดินทาง

งานวิจัยของ [9] ได้สรุปเกณฑ์การขับขี่ของยานพาหนะรูปแบบ Shuttle Bus ไว้ในช่วงการขับขี่ Comfort, Normal, Aggressive, Extremely aggressive และ Emergency braking ดังรูปที่ 12 โดยใช้ความเร่งด้านข้าง (Lateral Acceleration) ความเร่งด้านหน้า (Longitudinal Acceleration) การเปลี่ยนแปลงของอัตราเร่ง (Jerk) ทั้งในแนว Lateral และ Longitudinal มาเป็นเกณฑ์ในการเบรก เปลี่ยนเลน หรือการวางแผนเส้นทางเส้นใหม่



รูปที่ 12 เกณฑ์การขับขี่ของรถรูปแบบ Shuttle Bus

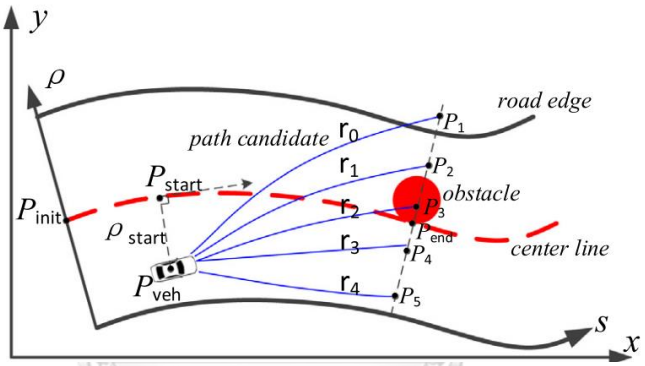
การสร้างเส้นทางใหม่ (Generate new path) มีด้วยกันหลายวิธีขึ้นอยู่กับวัตถุประสงค์ของการสร้างเส้นทาง เช่น การสร้างเส้นทางใหม่เพื่อหลีกเลี่ยงการชนวัตถุ (Generate new path) ในงานวิจัยของ [10] เสนอการสร้างเส้นทางเพื่อหักเลี้ยวหลบวัตถุ (Path avoidance) โดยมีขั้นตอนในการสร้างเส้นทางใหม่คือ 1.การระบุตำแหน่งบน Center line ของเส้นทางดังรูปที่ 13 ใช้การระบุตำแหน่งโดยการแปลงตำแหน่งจาก Cartesian coordinate ไปเป็น $s-\rho$ coordinate โดย Center line คือเส้นแบ่งเลนของถนน และ ρ คือระยะที่ใกล้ที่สุดระหว่างรถอัตโนมัติและ Center line ส่วน ρ_0 คือจุดบน Center line ที่ใกล้รถอัตโนมัติมากที่สุด



รูปที่ 13 การระบุตำแหน่งใน $s-\rho$ coordinate

เมื่อระบุตำแหน่งได้แล้ว จะสร้างเส้นทางใหม่ขึ้นมาเพื่อหาเส้นทางที่เหมาะสมที่สุด (Generate path candidate) โดยคำนวณเส้นทางใน $s-\rho$ coordinate ได้ดังสมการที่ 4 โดยที่ค่า s_{start} s_{end} ρ_{start} และ ρ_{end} เป็นระยะดังรูปที่ 14

$$\rho(s) = \begin{cases} a(s-s_{start})^3 + b(s-s_{start})^2 + c(s-s_{start}) + \rho_{start} & s \in [s_{start}, s_{end}] \\ \rho_{end} & others \end{cases} \quad (4)$$



รูปที่ 14 การสร้างเส้นทางใหม่ที่เป็นไปได้ (Generate path candidate)

เมื่อสร้างเส้นทางได้แล้ว ขั้นสุดท้ายคือการเลือกเส้นทางให้เหมาะสม (Selection path candidate) โดยคำนึงถึงค่าความปลอดภัย (Cost Safety) ความเรียบของเส้นทาง (Smoothness) และความต่อเนื่อง (Consistency) สุดท้ายการเลือกเส้นทางที่เหมาะสมจะเลือกจากสมการที่ 5 เลือกเส้นทางที่มีค่า Cost น้อยที่สุด

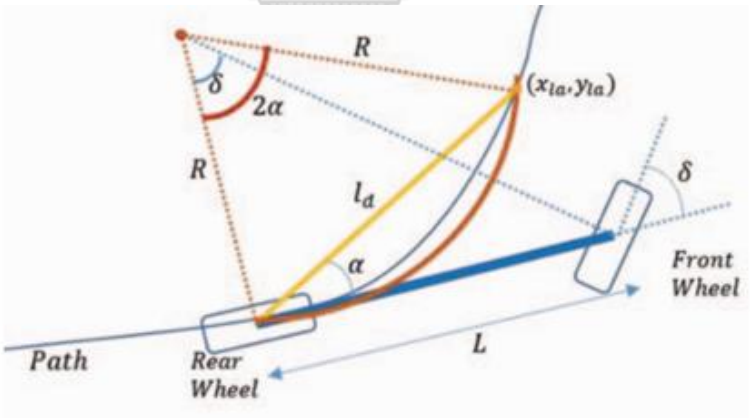
$$J(i) = \omega_S C_S(i) + \omega_K C_K(i) + \omega_C C_C(i) \quad (5)$$

โดยที่ C_s คือ Cost สำหรับความปลอดภัย, C_k คือ Cost สำหรับความนุ่มนวล, C_c คือ Cost สำหรับความต่อเนื่องของเส้นทาง, ω_S คือ Weight factor ด้านความปลอดภัย , ω_K คือ Weight factor การคำนวณด้านความนุ่มนวล และ ω_C คือ Weight factor การคำนวณด้านความต่อเนื่องของเส้นทาง

2.3 การกระทำ (Actuation)

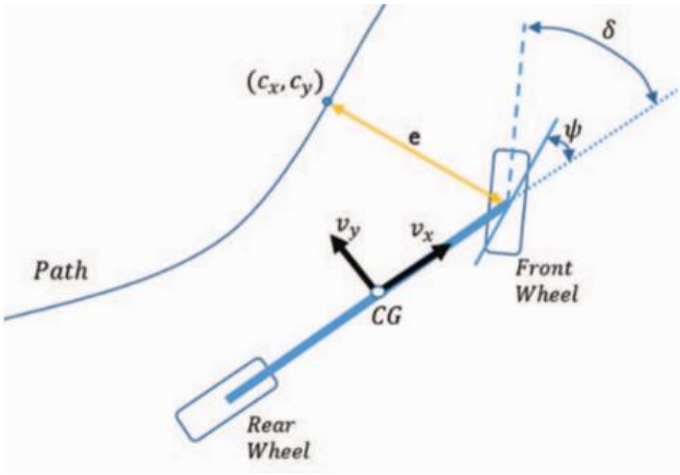
เมื่อรถอัตโนมัติสร้างเส้นทางใหม่ในการเคลื่อนที่จะเรียกเส้นทางที่สร้างใหม่ว่า Desire trajectory ซึ่งเมื่อรถอัตโนมัติเคลื่อนที่ตามเส้นทางของ Desire trajectory ที่ออกแบบจะเรียกว่า Actual path ซึ่งจะเกิด Lateral error คือ Error ด้านข้างระหว่าง Desire trajectory และ Actual path ที่ส่งผลให้เกิดเหตุอันตรายได้ เช่นหาก Lateral error มาก แล้วรถกินเลนซ้ายจะมีความเสี่ยงต่อยานพาหนะที่ขับตามมาในเลนซ้าย ซึ่งสาเหตุของการเกิด Lateral error พบว่ามาได้จากหลายสาเหตุ ทั้ง Dynamic model ของรถ หรือการระบุตำแหน่ง (Localization) ปัจจุบันนักวิจัยหาแนวทางการแก้ปัญหา Lateral error มีหลายวิธี โดยจะยกวิธีที่นิยมใช้กันอย่างแพร่หลายคือ Pure Pursuit Stanley มาอธิบายดังต่อไปนี้

- Pure Pursuit วิธีนี้ใช้ระยะมองหน้า (Looking Forward) และการหาจุดเป้าหมาย (Target point) ในการคำนวณเพื่อหามุมเลี้ยวของพวงมาลัยที่จะหักเลี้ยวให้เข้าเส้นทางที่กำหนดได้ดังรูปที่ 15 ขั้นตอนของวิธีนี้คือ ทำการค้นหาจุดเป้าหมาย (Target point) เพื่อทราบมุมระหว่าง Vehicle heading และจุดเป้าหมาย (Target point) แล้วจะสามารถหามุมเลี้ยว (Steering angle) ได้ วิธีนี้เป็นวิธีที่นิยมใช้มากที่สุดในการติดตามเส้นทาง (Track Path) เนื่องจากว่าเป็นวิธีที่ง่ายและมีประสิทธิภาพ [11]



รูปที่ 15 รูปทางเรขาคณิตที่ใช้ในการคำนวณของวิธี Pure Pursuit

- Stanley วิธีนี้ใช้ระยะทางที่ใกล้ที่สุดระหว่างรถอัตโนมัติและ Trajectory path เพื่อหาระยะ Cross track error เพื่อคำนวณค่ามุมหักเลี้ยว (Steering angle) ที่ต้องการเพื่อให้เข้าใกล้ Trajectory path ที่สุด คำนวณมุมหักเลี้ยวได้ ดังรูปที่ 16



รูปที่ 16 รูปทางเรขาคณิตที่ใช้ในการคำนวณของวิธี Stanley

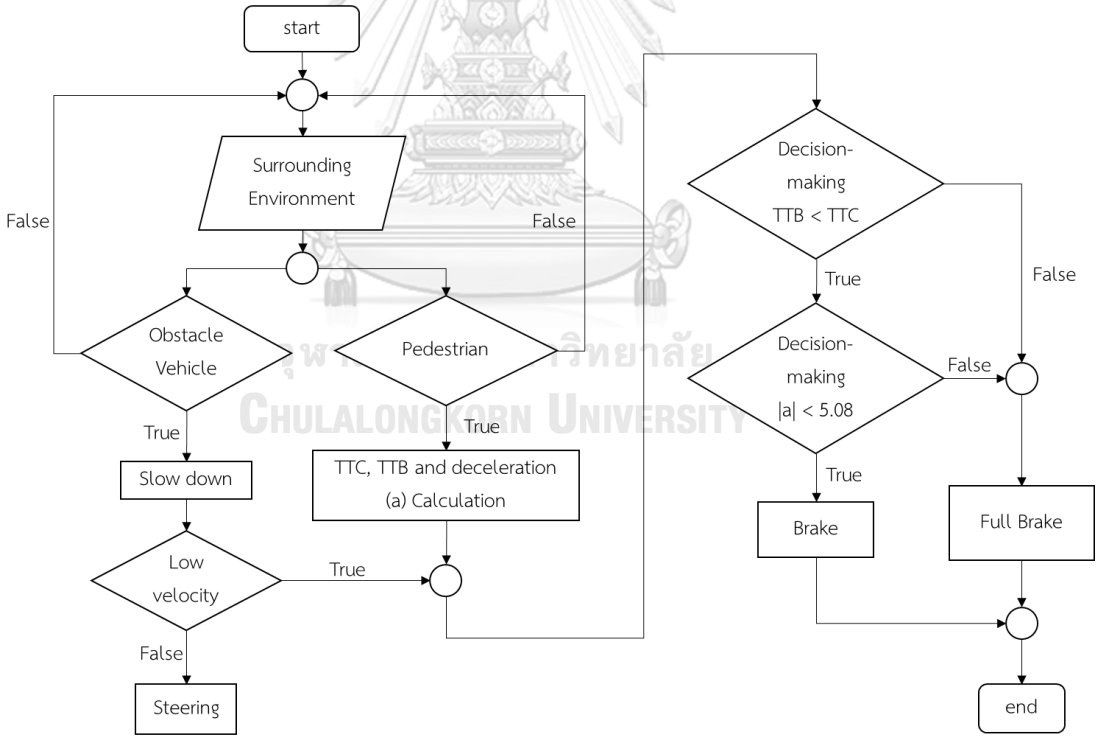
โดยสรุปทั้ง 2 วิธีที่นำมาเสนอวิธี Pure Pursuit เหมาะกับการควบคุมมุมเลี้ยวที่ความเร็วต่ำ ขณะที่ Stanley เหมาะกับการควบคุมมุมเลี้ยวที่ความเร็วสูงมากกว่า ซึ่งความสามารถในการลดค่า Lateral error ขึ้นอยู่กับการใช้งาน ต้องพิจารณาหลายส่วนประกอบ ทั้งความเร็วในการเคลื่อนที่ของยานพาหนะ การระบุตำแหน่งที่แม่นยำของยานพาหนะ (Localization) เป็นต้น [11]

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

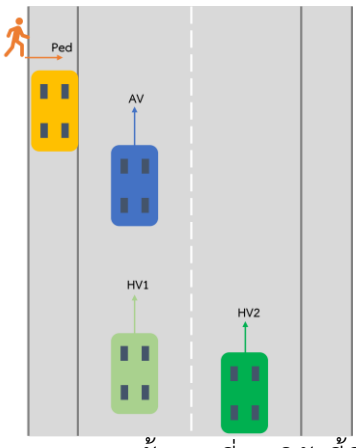
บทที่ 3 แนวคิดและการออกแบบยานพาหนะอัตโนมัติ

3.1 การออกแบบระบบการตัดสินใจ (Decision-making system)

แผนภาพการทำงานของระบบการตัดสินใจที่งานวิจัยนี้เสนอเป็นดังรูปที่ 18 เมื่อรถอัตโนมัติเจอวัตถุบดบังที่อยู่ในถนน (Obstacle object) ดังรูปที่ 19 คือยานพาหนะคันสีเหลือง จะทำการชะลอความเร็วทันที ซึ่งการชะลอความเร็วจะใช้อัตราหน่วงภายในช่วง Comfort zone ในความเร็วต่ำ (Low level) คือความเร็ว 10-20 กม/ชม จะคำนวณค่า TTB และค่า TTC เพื่อพิจารณาในการตัดสินใจ ส่วนในความเร็วสูง (High velocity) คือความเร็ว 25-30 กม/ชม เมื่อถึงระยะทางที่เหมาะสมที่ห่างจากด้านหน้าของวัตถุบดบัง (Obstacle object) แล้วยังไม่พบคนเดินเท้า จะเปลี่ยนเลนไปเลนขวา แต่หากเจอคนเดินเท้าก่อนจะคำนวณค่า TTB (Time-To-Brake) และ TTC (Time-To-Collision) มาเพื่อพิจารณาในการตัดสินใจเช่นเดียวกับความเร็วต่ำ (Low velocity) โดยจะอธิบายค่า TTC และ TTB ดังต่อไปนี้



รูปที่ 17 แผนภาพ Algorithm ที่งานวิจัยนี้นำเสนอ



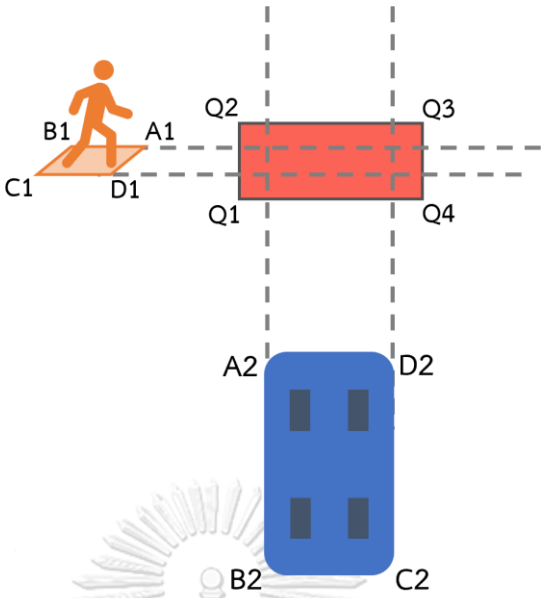
รูปที่ 18 ยานพาหนะทั้งหมดที่งานวิจัยนี้พิจารณา

3.1.1 TTC (Time-To-Collision)

การคำนวณหาค่า TTC ที่บอกเวลาในการชน ที่นำมาใช้พิจารณาในการตัดสินใจใน Algorithm ที่ออกแบบ ในงานวิจัยนี้มีทั้งหมด 3 ค่าคือ TTC ระหว่างคนเดินเท้าและรถอัตโนมัติ ($TTC_{av,ped}$), TTC ระหว่างรถอัตโนมัติและรถที่ตามหลังคันที่ 1 ($TTC_{av,hv1}$) และ TTC ระหว่างรถอัตโนมัติและรถที่ตามหลังคันที่ 2 ($TTC_{av,hv2}$) โดยยานพาหนะที่ตามหลังคันที่ 1 และ 2 คือยานพาหนะที่ตามหลังในเลนซ้ายและเลนขวาตามลำดับ ดังรูปที่ 19 ซึ่งค่า TTC แต่ละตัวมีรายละเอียดดังต่อไปนี้

3.1.1.1 $TTC_{av,ped}$

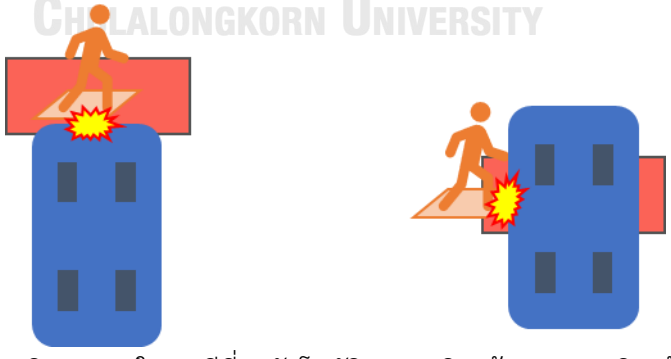
การคำนวณหาค่า $TTC_{av,ped}$ จะถือว่าทั้งคนและรถเคลื่อนที่ในทิศทางที่ตั้งฉากกัน ใช้วิธีของ F. Jimenez และคณะ [8] ที่กล่าวไว้ในบทที่ 2 มาปรับใช้ในการคำนวณ โดยกำหนดการเรียกชื่อแต่ละมุมของรถอัตโนมัติและคนเดินเท้าเป็น A2-D2 และ A1-D1 ตามลำดับ และ Conflict area คือพื้นที่ที่เกิดจาก Boundary box ของคนเดินเท้าและรถอัตโนมัติตัดกัน (Intersection) โดยจะเผื่อ Buffer ลงไปในด้านกว้างและยาวด้านละ 0.3 m กำหนดชื่อมุมของ Conflict area เป็น Q1-Q4 ดังรูปที่ 20



รูปที่ 19 การเรียกชื่อ Boundary Box ของคนเดินเท้า, รถอัตโนมัติ และ Conflict area

กำหนดชื่อเรียกของเวลาแต่ละมุมของวัตถุ (Object) ที่ใช้เพื่อเข้า และออกจาก Conflict area เพื่อให้ง่ายต่อการคำนวณดังต่อไปนี้ $T_{D1,in}$ คือเวลาที่มุม D1 ของคนเดินเท้า ใช้เพื่อเข้า Conflict area ค่า $T_{C1,out}$ คือเวลาที่มุม C1 ของคนเดินเท้า ใช้เพื่อออกจาก Conflict area ค่า $T_{A2,in}$ คือเวลาที่มุม A2 ของรถอัตโนมัติ ใช้เพื่อเข้า Conflict area และ $T_{B2,out}$ คือเวลาที่มุม B2 ของรถอัตมัติ ใช้เพื่อออกจาก Conflict area การชนในสถานการณ์ที่คนเดินเท้าและรถอัตโนมัติมีทิศทางการเคลื่อนที่ตั้งฉากกัน เกิดได้ 2 กรณี คือ กรณีที่รถอัตโนมัติชนคนเดินเท้า และกรณีที่คนเดินเท้าชนรถอัตโนมัติ ดังรูปที่ 21 ซึ่งมีแต่ละกรณีเงื่อนไขดังต่อไปนี้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

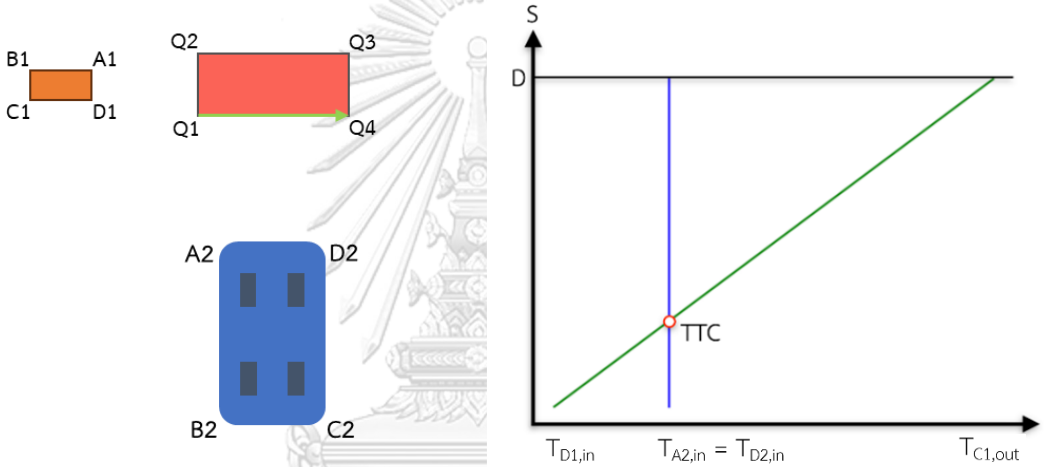


รูปที่ 20 กรณีการเกิดการชนในกรณีที่รถอัตโนมัติชนคนเดินเท้า และคนเดินเท้าชนรถอัตโนมัติ

- กรณีที่รถอัตโนมัติชนคนเดินเท้า

การชนเกิดขึ้นระหว่าง Q1Q4 ใน Conflict area โดยการชนเกิดที่ด้านหน้ารถอัตโนมัติ ดังนั้น จะสนใจเฉพาะมุม A2 และ D2 (ซึ่ง $T_{A2,in}=T_{D2,in}$) และเกิดการชนที่ด้านข้างของคนเดินเท้า ดังนั้นจะสนใจเฉพาะมุม C1 และ D1 โดยจะเกิดการชนขึ้นเมื่อ $T_{C1,out} > T_{A2,in} (=T_{D2,in}) > T_{D1,in}$

เมื่อนำเวลาทั้ง 4 ค่าไปพลอตกราฟ จะได้กราฟรูปที่ 22 กราฟเส้นสีน้ำเงินคือรถอัตโนมัติ กราฟเส้นสีเขียวคือคนเดินเท้า การชนเกิดขึ้นที่ Q1Q4 ดังนั้นจะสนใจคนเดินเท้าที่มุม D1 เข้า Conflict area (s=0) และมุม C1 ที่ออกจาก Conflict area (s=D ซึ่ง D=Q1Q4) สนใจรถอัตโนมัติที่มุม A2 ที่เข้า Conflict area (s=0) และมุม D2 ที่เข้า Conflict area (s=D) จะเห็นว่าจุดตัดของกราฟสองเส้นคือค่า $T_{A2,in}$ นั่นคือค่า $TTC_{av,ped}$ ที่ทำให้เกิดการชนในกรณีที่รถอัตโนมัติชนคนเดินเท้า

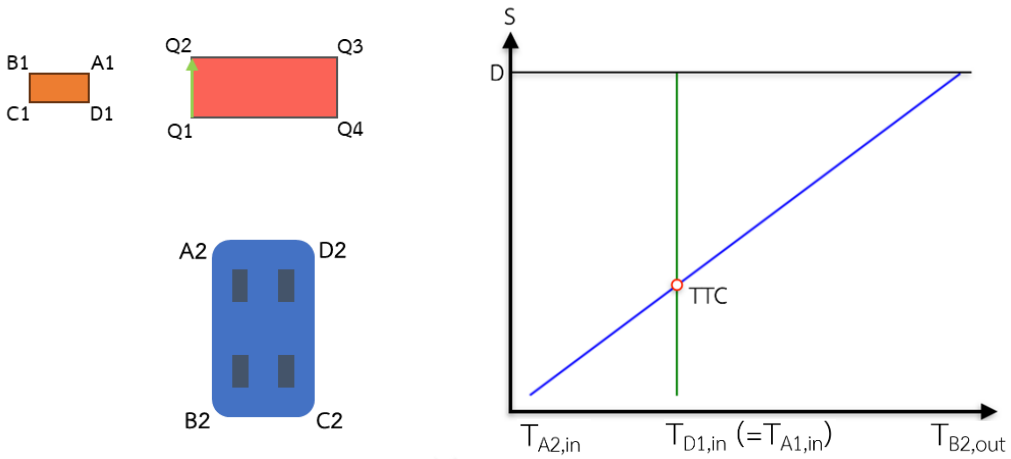


รูปที่ 21 การชนกรณีรถอัตโนมัติชนคนเดินเท้า

- กรณีที่คนเดินเท้าชนรถอัตโนมัติ

การชนเกิดขึ้นระหว่าง Q1Q2 ใน Conflict area โดยการชนเกิดที่ด้านข้างของรถอัตโนมัติ ดังนั้นจะสนใจเฉพาะมุม A2 และ B2 และเกิดการชนที่ด้านหน้าของคนเดินเท้า ดังนั้น จะสนใจเฉพาะมุม A1 และ D1 (ซึ่ง $T_{A1,in}=T_{D1,in}$) โดยจะเกิดการชนขึ้นเมื่อ $T_{B2,out} > T_{D1,in} (=T_{A1,in}) > T_{A2,in}$

จากกราฟรูปที่ 23 เป็นการชนในกรณีที่คนเดินเท้าชนรถอัตโนมัติ การชนเกิดขึ้นที่ Q1Q2 ดังนั้นจะสนใจรถอัตโนมัติที่มุม A2 เข้า Conflict area (s=0) และมุม B2 ที่ออกจาก Conflict area (s=D ซึ่ง D=Q1Q2) สนใจคนเดินเท้าที่มุม D1 ที่เข้า Conflict area (s=0) และมุม A1 ที่เข้า Conflict area (s=D) จะเห็นว่าจุดตัดของกราฟสองเส้นคือค่า $T_{D1,in}$ นั่นคือค่า $TTC_{av,ped}$ ที่ทำให้เกิดการชนในกรณีที่คนเดินเท้าชนรถอัตโนมัติ



รูปที่ 22 กรณีที่คนเดินเท้าชนรถอัตโนมัติ

จากทั้งสองกรณีสามารถสรุปเป็นสมการการหาค่า $\mathbf{TTC_{av,ped}}$ ได้ดังสมการที่ 6

$$\mathbf{TTC_{av,ped}} = \begin{cases} \mathbf{T_{A2,in}}\ ; & \mathbf{T_{C1,out} > T_{A2,in} > T_{D1,out}} \\ \mathbf{T_{D1,in}}\ ; & \mathbf{T_{B2,out} > T_{D1,in} > T_{A2,in}} \end{cases} \quad (6)$$

3.1.1.2 $TTC_{av,hv1}$

เนื่องจากว่าการจะตัดสินใจที่จะชะลอความเร็วหรือเบรกต้องคำนึงถึงยานพาหนะคันหลังด้วย ในการคำนวณค่า $\mathbf{TTC_{AV,HV1}}$ ในงานวิจัยนี้จัดทำขึ้นเพื่อแสดงให้เห็นว่า Algorithm ที่นำเสนอไม่ได้ส่งผลที่ให้ยานพาหนะคันหลังชนท้ายรถอัตโนมัติ หากรักษาระยะห่างตามที่งานวิจัยของ [12] ที่กล่าวไว้ว่าควรรักษาระยะห่าง Time Headway (H_t) ที่แนะนำให้ผู้ขับขี่รักษาไว้มีค่าอยู่ที่ 2 วินาที ซึ่งจะสามารถคำนวณ Headway Distance (H) ได้ดังสมการที่ 7

$$\mathbf{H = H_t V_{HV1}} \quad (7)$$

โดยที่ V_{HV1} คือความเร็วของยานพาหนะ HV1 ดังนั้นจะสามารถหาค่า $TTC_{av,hv1}$ ได้จากสมการที่ 8

$$\mathbf{TTC_{AV,HV1}} = \frac{\mathbf{V_{HV1} H_t - l_{AV}}}{\mathbf{V_{HV1} - V_{AV}}} \quad (8)$$

โดยที่ l_{AV} คือความยาวของรถอัตโนมัติ และ $\mathbf{V_{AV}}$ คือความเร็วของรถอัตโนมัติ โดยกำหนดค่า $TTC_{AV,HV1,Threshold}$ เป็น 1 วินาที ในส่วนของผลการทดลองจะอธิบายว่าทุกเคสเมื่อรถอัตโนมัติเบรกหรือ Emergency braking หรือหักเลี้ยวเพื่อเปลี่ยนเลนค่า $TTC_{av,hv1}$ ที่เกิดขึ้นจะยังน้อยกว่า $TTC_{AV,HV1,Threshold}$ ที่กำหนดเสมอ และจะแสดงค่าอัตราหน่วงที่ยานพาหนะ HV1 ต้องรักษา Time Headway เป็น 2 วินาทีไว้ด้วย

3.1.1.3 $TTC_{av,hv2}$

หากเป็นถนนที่รถอัตโนมัติสามารถตัดสินใจที่จะเปลี่ยนเลนได้ ค่า TTC ระหว่างรถอัตโนมัติกับรถคันหลังในเลนที่จะเปลี่ยนไปอยู่ ควรจะต้องคำนึงถึงด้วย เพื่อให้ทราบว่าหากรถเปลี่ยน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

เลนไปแล้วจะไม่เกิดการชนเกิดขึ้น โดยจะนำงานวิจัย “An Improved Method to Calculate the Time-to-Collision” ที่ได้อธิบายไว้ในบทที่ 2 มาใช้ในการคำนวณ หากคำนวณแล้วเกิดการชนจะไม่หักเลี้ยวออกไป หรือคำนวณแล้วมีความเสี่ยงที่จะเกิดการชนคือค่า $TTC_{av,hv2,threshold}$ น้อยกว่า 3 วินาที จะไม่ทำการหักเลี้ยว

3.1.2 TTB (Time-To-Brake)

การคำนวณค่า TTB ที่รถอัตโนมัติต้องใช้ จะคำนวณจากการใช้ค่าอัตราหน่วงมากที่สุดที่รถอัตมัติจะทำ ซึ่งค่าอัตราหน่วงมากที่สุดกำหนดเป็น -7 m/s^2 ตามเกณฑ์ของบทที่ 2 โดยจะสามารถคำนวณเวลาได้จากสมการที่ 9 โดยจะคิดค่า Time delay เป็น 0.2 วินาที

$$\mathrm{TTB} = -\frac{\mathrm{V_{AV}(t)}}{\mathrm{a_{max}}} + \mathrm{T_{delay}} \qquad (9)$$

เมื่อคำนวณค่า TTB และ TTC ได้แล้ว การตัดสินใจหลังจากเจอคนเดินเท้าใน Algorithm ที่ออกแบบแบ่งเป็นกรณีความเร็วต่ำและความเร็วสูง

- ที่ความเร็วต่ำ ถ้าค่า TTB < TTC จะพิจารณาต่อว่าถ้าค่าอัตราหน่วงที่คำนวณมาเพื่อใช้ในการเบรก |a| < 5.08 m/s^2 จะเบรกด้วยอัตราหน่วงที่คำนวณ (a) แต่หาก |a| > 5.08 จะเบรกด้วย Maximum deceleration แต่ถ้าหาก TTB > TTC จะตัดสินใจ Fullbrake ด้วยค่า Maximum Deceleration เช่นกัน
- ที่ความเร็วสูง ก่อนถึงระยะที่เหมาะสมจาก Conflict area หากรถอัตโนมัติยังไม่สามารถตรวจจับคนเดินเท้า จะทำการหักเลี้ยวเพื่อเปลี่ยนเลน เพื่อเพิ่มระยะการมองเห็นให้กับคนเดินเท้า แต่หากก่อนระยะที่เหมาะสมดังกล่าวรถอัตโนมัติตรวจเจอคนเดินเท้า จะเบรก

ทั้งนี้เราเสนอให้มีการหักเลี้ยวเพื่อเปลี่ยนเลนในความเร็วสูง เนื่องจากหากความเร็วยิ่งต่ำการหักเลี้ยวยิ่งเป็นไปได้ยาก มีสาเหตุมาจากความตึงเครียดของล้อและยาง เมื่อรถหักเลี้ยวด้วยความเร็วต่ำ การบิดและเบี่ยงตัวของล้อและยางมีน้อยกว่าเมื่อรถหักเลี้ยวด้วยความเร็วสูง ทำให้รถมีทิศทางการเคลื่อนที่ที่น่าควบคุมยากขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2 การออกแบบ Motion planning

จากระบบการตัดสินใจดังกล่าวการกระทำ (Action) ที่รถอัตโนมัติเลือกกระทำมีทั้งหมด 3 อย่าง คือ ชะลอความเร็ว, เบรก และหักเลี้ยวหลบด้วยการเปลี่ยนเลน มีรายละเอียดดังต่อไปนี้

3.2.1 การชะลอความเร็ว (Slow down)

เมื่อระบบตรวจจับวัตถุ (Detection system) ของรถอัตโนมัติเจอวัตถุบดบังที่อยู่บนถนน (Obstacle object) จะเลือกชะลอความเร็วทันที การชะลอความเร็วจะทำในช่วงการขับขี่ช่วง Comfort และชะลอความเร็วให้เหมาะสมกับความเร็วต่างๆ ของรถอัตโนมัติ

รถอัตโนมัติที่ตรวจจับเจอวัตถุบดบังในวินาทีแรกที่ต้องชะลอความเร็ว จะสร้างค่าเริ่มต้นของความกว้างและความยาวของวัตถุบดบัง (Obstacle object) ให้กับรถอัตโนมัติก่อน เพื่อคำนวณอัตราหน่วงที่ต้องใช้เพื่อชะลอความเร็วในตอนแรก (a^*_{slow}) จากสมการที่ 10

$$a^*_{slow} = \frac{v_{av,slow}{}^2 - v_{av}(t)^2}{2s^*} \quad (10)$$

โดยที่ $v_{av,slow}$ คือความเร็วสุดท้ายที่ใช้ในการชะลอความเร็วกำหนดให้เป็น 5 km/h, $v_{av}(t)$ คือความเร็วปัจจุบันของรถอัตโนมัติ และ s^* คือระยะทางในการชะลอความเร็วที่คำนวนมาจากค่าความกว้างและความยาวของวัตถุบดบังที่เป็นค่าเริ่มต้น คำนวณจากสมการที่ 11

$$s^* = \sqrt{r_{sensor}{}^2 - L_{OB,lat}{}^2} + L_{OB,default} \quad (11)$$

รูปที่ 24 แสดง Geometry ที่ใช้คำนวณ โดยที่ r_{sensor} คือระยะรัศมีของ sensor, $L_{OB,lat}$ คือ ระยะห่างระหว่างริมขวาของยานพาหนะที่จอดอยู่ข้างถนนกับกึ่งกลางเลนของรถอัตโนมัติในแนว Lateral และ $L_{OB,default}$ คือ ค่าเริ่มต้นของความยาวของรถที่จอดข้างถนนที่ตั้งไว้ขณะที่ยังตรวจจับเจอรถที่จอดอยู่ข้างถนนยังไม่ครบทั้ง Boundary กำหนดให้เป็น 2.5 เมตร

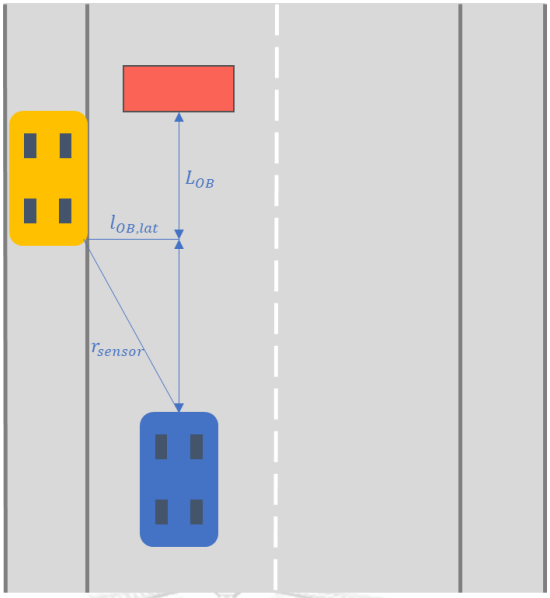
เมื่อรถอัตโนมัติเคลื่อนที่ต่อจนสามารถทราบความยาวของวัตถุบดบัง (Obstacle object) ที่แท้จริงได้ จะคำนวณอัตราหน่วง (a_{slow}) ใหม่อีกครั้งดังสมการที่ 12

$$a_{slow} = \frac{v_{av,slow}{}^2 - v_{av}(t)^2}{2s} \quad (12)$$

โดยที่ s คือระยะทางในการชะลอความเร็วที่คำนวนมาจากค่าความกว้างและความยาวของวัตถุบดบังที่แท้จริง (L_{OB}) ที่คำนวณจากสมการที่ 13

$$s = \sqrt{r_{sensor}{}^2 - L_{OB,lat}{}^2} + L_{OB} \quad (13)$$

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



รูปที่ 23 Geometry ของรถอัตโนมัติขณะตรวจเจอวัตถุอยู่ข้างทาง (Obstacle object)

ตารางที่ 5 แสดงอัตราหน่วงที่ใช้ในการชะลอความเร็วในช่วงที่ตรวจเจอยังไม่ครบทั้ง Boundary ของยานพาหนะที่จอดอยู่ข้างทาง (Obstacle object) จะเห็นว่ามีอัตราหน่วงที่คำนวณได้ทั้งหมด 5 กรณี โดยจะเลือกใช้การชะลอความเร็วไม่เกิน 0.9 m/s^2 ซึ่งเป็นการชะลอในช่วง Comfort zone

ตารางที่ 5 อัตราหน่วงที่ใช้ในการชะลอความเร็ว

Vav (km/h)	การชะลอความเร็วโดยใช้ Default	
	ที่คำนวณ	ที่ใช้
10	-0.12	-0.12
15	-0.35	-0.35
20	-0.65	-0.65
25	-1.04	-0.9
30	-1.44	-0.9

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

3.2.2 การเบรก (Braking)

การเบรกสามารถทำได้ตั้งแต่ช่วง Normal ไปจนถึง Emergency braking โดยเป็นไปตามผลลัพธ์ของการตัดสินใจ ค่าอัตราหน่วงที่ใช้เบรก ($\mathbf{a_{brake}}$) โดยคำนวณจากระยะทางที่รถอัตโนมัติ

ต้องเบรกสนิท (S_{brake}) ซึ่งคำนวณได้จากระบบการรับรู้ (Perception) ดังสมการที่ 14 โดยที่ $V_{av}(t)$ คือความเร็วของรถอัตโนมัติขณะนั้น

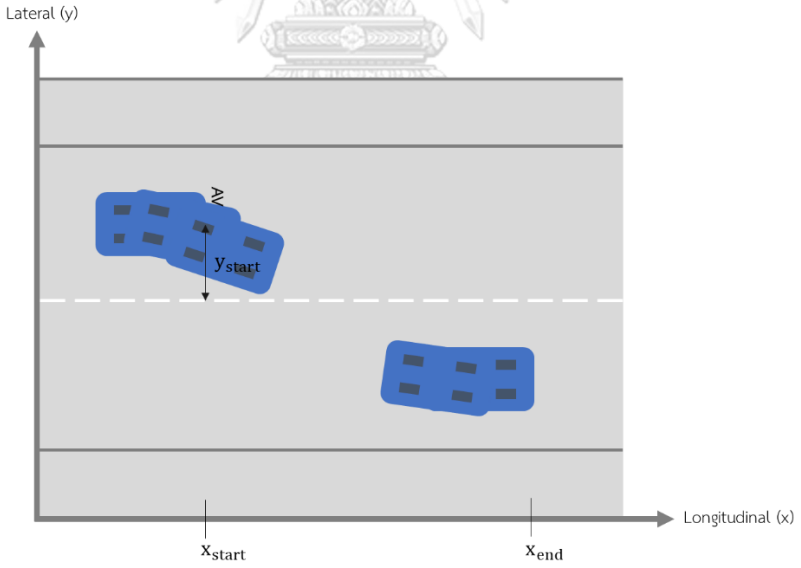
$$a_{brake} = -\frac{V_{av}(t)^2}{2S_{brake}} \quad (14)$$

3.2.3 การหักเลี้ยวหลบ (Steering)

หากระบบการรับรู้ของรถอัตโนมัติ (Perception system) ทราบว่าในเส้นทางนี้สามารถใช้การหักหลบได้ หรือมีพื้นที่พอในการหลักหลบจะสร้างเส้นทางการหักหลบโดยการเปลี่ยนเลนไปยังเลนขวาถัดไป หากผ่านเงื่อนไขทั้งหมด จะสร้างเส้นทางที่คำนวณจากสมการที่ 15

$$y = ax^3 + bx^2 + cx + y_{start} \quad x \in [x_{start}, x_{end}] \quad (15)$$

โดยที่ y คือระยะในแนว Lateral ในหน่วย m, x คือระยะในแนว Longitudinal ในหน่วย m กำหนดให้จุด Original ของแกน Lateral และ Longitudinal อยู่ที่ริมขวาสุดของถนน ส่วน x_{start} และ x_{end} เป็นระยะการเปลี่ยนเลนที่อ้างอิงกับเส้นแบ่งเลน ดังรูปที่ 25 ส่วน y_{start} คือตำแหน่งเริ่มต้นในใน Longitudinal เมื่อรถอัตโนมัติเริ่มเปลี่ยนเลน และเมื่อทราบเส้นทางในการเปลี่ยนเลนได้แล้ว สามารถหา Heading angle (θ) ได้ดังสมการที่ 16



รูปที่ 24 พารามิเตอร์เกี่ยวกับการเปลี่ยนเลน

$$\theta = \arctan\left(\frac{dy}{dx}\right) \quad (16)$$

และสามารถหาความเร็วในแนว Lateral ($\mathbf{v_{lat}}$) และ Longitudinal ($\mathbf{v_{long}}$) ได้ดังสมการที่ 17 และ 18 โดยเรากำหนดให้ความเร็วขณะเปลี่ยนเลนคงที่เสมอดังนั้นจึงสามารถนำความเร็วขณะเปลี่ยนเลน($\mathbf{v_{av,steer}}$) มาคำนวณได้เลย

$$\mathbf{v_{lat}} = \mathbf{v_{av,steer}}\sin\theta \quad (17)$$

$$\mathbf{v_{long}} = \mathbf{v_{av,steer}}\cos\theta \quad (18)$$

และสามารถหาอัตราหน่วงในแนว Lateral และ Longitudinal ได้ดังสมการที่ 19 และ 20 ซึ่งจะกำหนดให้อัตราหน่วงในแนว Lateral อยู่ในช่วง [-4, 4] m/s^2 และอัตราหน่วงในแนว Longitudinal อยู่ในช่วง [-2, 1.5] m/s^2 ซึ่งอยู่ในช่วง Normal driving zone

$$\mathbf{a_{lat}} = \frac{\mathbf{dv_{lat}}}{\mathbf{dt}} \quad (19)$$

$$\mathbf{a_{long}} = \frac{\mathbf{dv_{long}}}{\mathbf{dt}} \quad (20)$$

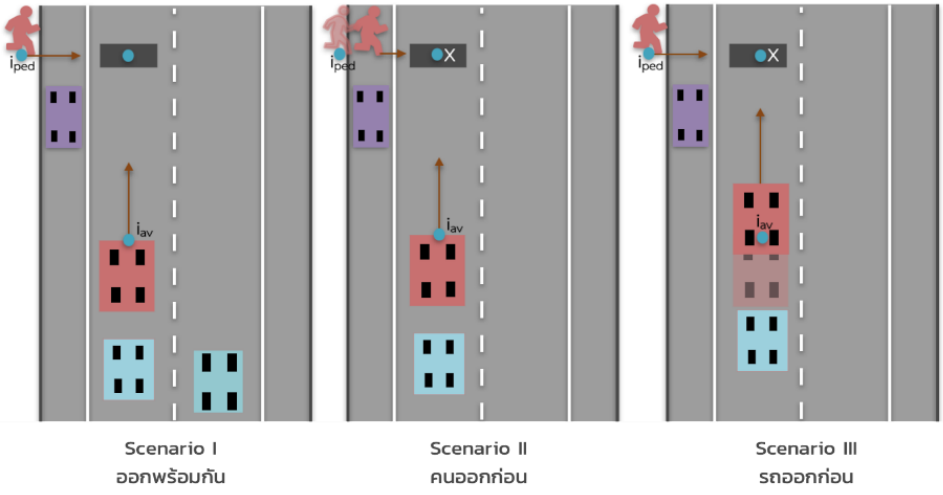
ทั้งนี้ในการเปลี่ยนเลนเราจะพิจารณาค่า Jerk ด้วย ทั้งสองแนวด้วยเนื่องจากอาจเกิด Rollover ได้ Jerk ทั้งสองแนวสามารถคำนวณได้จากสมการที่ 21 และ 22 ซึ่งกำหนดให้ $\mathbf{jerk_{lat}}$ และ $\mathbf{jerk_{long}}$ อยู่ในช่วง [-2, 2] m/s^3

$$\mathbf{jerk_{lat}} = \frac{\mathbf{da_{lat}}}{\mathbf{dt}} \quad (21)$$

$$\mathbf{jerk_{long}} = \frac{\mathbf{da_{long}}}{\mathbf{dt}} \quad (22)$$

3.3 ขั้นตอนการทำการทดลอง

ในงานวิจัยนี้สนใจสถานการณ์การข้ามถนนของคนเดินเท้า 3 สถานการณ์ดังรูปที่ 26



รูปที่ 25 สถานการณ์ที่งานวิจัยนี้นำมาพิจารณา

- สถานการณ์ที่ 1 รถอัตโนมัติและคนเดินเท้าออกจากจุดเริ่มต้นของตัวเองพร้อมกัน

- สถานการณ์ที่ 2 คนเดินเท้าออกจากจุดเริ่มต้น 1, 2 และ 3 วินาที ก่อนที่รถอัตโนมัติจะออกจากจุดเริ่มต้น
- สถานการณ์ที่ 3 รถอัตโนมัติออกจากจุดเริ่มต้น 1, 2 และ 3 วินาที ก่อนที่คนเดินเท้าจะออกจากจุดเริ่มต้น

โดยที่จุดเริ่มต้นของรถอัตโนมัติ (i_{av}) คือจุดแรกที่รถอัตโนมัติตรวจเจอวัตถุบดบัง (Obstacle object) และจุดเริ่มต้นของคนเดินเท้า (i_{ped}) คือจุดแรกที่หากคนเดินเท้าอยู่ในจุดเริ่มต้นนี้จะมองไม่เห็นรถอัตโนมัติ

ตารางที่ 6 พารามิเตอร์ที่ใช้ในงานวิจัย

ตัวแปร	ความหมาย	(m)
r_{sensor}	ระยะเซนเซอร์ที่รถอัตโนมัติตรวจจับวัตถุได้	20
W_{av}	ความกว้าง Boundary รถอัตโนมัติ	2.07
L_{av}	ความยาว Boundary รถอัตโนมัติ	4.48
W_{ped}	ความกว้าง Boundary คนเดินเท้า	0.38
L_{ped}	ความยาว Boundary คนเดินเท้า	0.38
W_{ob}	ความกว้าง Boundary ยานพาหนะที่จอดบังอยู่	1.93
L_{ob}	ความยาว Boundary ยานพาหนะที่จอดบังอยู่	4.72
W_{hv}	ความกว้าง Boundary ยานพาหนะคันหลัง	1.93
L_{hv}	ความยาว Boundary ยานพาหนะคันหลัง	4.72
$L_{ped,long}$	ระยะห่างในแนว Longitudinal ระหว่างคนเดินเท้ากับรถที่จอดอยู่ข้างทาง	2
W_{road}	ความกว้างของถนนใน 1 เลน	3
$W_{shoulder}$	ความกว้างของไหล่ทาง	2

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.2.1 การทำการทดลองใน Simulation

ใน Simulation จะทดลองการใช้ระบบการตัดสินใจที่ออกแบบบน Python แล้วนำมาดูผลเป็นช่วงที่รถอัตโนมัติกระทำว่าอยู่ในช่วง Comfort, Normal, Aggressive, Emergency หรือ Collision โดยจะทำการทดลองทั้งหมด 630 เหตุการณ์ ดังตารางที่ 7 ซึ่งจะทำการทดลองเปรียบเทียบกับเหตุการณ์ที่ไม่ได้ใช้ Algorithm (W/O Algorithm) ที่ออกแบบในงานวิจัยนี้กล่าวคือการที่รถอัตโนมัติวิ่งตรงไปและไม่ชะลอความเร็วเมื่อตรวจเจอคนเดินเท้าที่กำลังข้ามถนนถึงจะตัดสินใจที่จะเบรก โดยจะพิจารณาความเร็วรถอัตโนมัติที่ 10, 15, 20, 25 และ 30 กม/ชม. ความเร็วคนเดินเท้าที่ 1, 1.2, 1.4, 1.6, 1.8 และ 2 เมตร/วินาที และพารามิเตอร์ต่างๆ ที่ใช้ในการคำนวณแสดงในตารางที่ 6

ตารางที่ 7 จำนวนเหตุการณ์ของแต่ละสถานการณ์

สถานการณ์	W/O Algorithm	With Algorithm	
		W/O Steering	With Steering
สถานการณ์ที่ 1	30	30	30
สถานการณ์ที่ 2	90	90	90
สถานการณ์ที่ 3	90	90	90

3.2.2 การทำการทดลองจริง (Experiment)

เมื่อเราได้ผลจากการ Simulation เรียบร้อยแล้ว จะเลือกเหตุการณ์มาทำการทดลองจริงจาก Action เบรกและการเปลี่ยนเลน ที่ปลอดภัยที่สุดมาทำการทดลอง เนื่องจากเหตุการณ์ที่ทำการ Simulation ถือเป็นเหตุการณ์ที่อันตรายมาก จึงควรต้องคำนึงถึงความปลอดภัยเป็นอันดับแรก โดยทำการทดลองหลัง 22.00 น. ที่หน้าคณะวิศวกรรมศาสตร์ ภายในจุฬาลงกรณ์มหาวิทยาลัย มีผู้ควบคุม (Staff) คอยควบคุมสถานการณ์มากกว่า 6 คน และมี Safety driver อยู่บนรถอัตโนมัติคอยควบคุมสถานการณ์ไม่ให้เกิดอุบัติเหตุในเหตุการณ์ที่ไม่คาดคิด

ขั้นตอนในการเตรียมการทดลองจริงจะเป็นขั้นตอนของการเตรียม Software ที่ใช้ระบบ ROS บน Autoware เพื่อให้แน่ใจว่าคำสั่งที่เราป้อนให้รถ ในการทดลองจริงจะใช้รถอัตโนมัติระดับที่ 3 (Turing) ของภาควิชาวิศวกรรมยานยนต์ จุฬาลงกรณ์มหาวิทยาลัย มาใช้ในการทดลองดังรูปที่ 27 ในส่วนของคนเดินเท้าจะใช้เป็น SIL (Software in the loop) ที่ TCP เป็นโปรโตคอลในการรับข้อมูลว่ารถอัตโนมัติถึงจุดเริ่มต้น เพื่อจำลองให้คนเดินเท้าเดิน เมื่อคนเดินเท้าเดินถึงเลนของรถอัตโนมัติ จะ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ส่งข้อมูลไปบอกรถอัตโนมัติ เพื่อให้รถทราบว่ามีคนเดินเท้าอยู่ข้างหน้า ในส่วนของวัตถุบดบัง (Obstacle object) จะใช้เป็น Boundary box ของขนาดรถ Mitsubishi Pajero sport 2012 โดยจะใช้ฟิวเจอร์บอร์โบอร์ดยึดกับโครงที่ทำจากท่อ PVC เพื่อลดความเสียหายจากความเสี่ยงจากอุบัติเหตุ ดังรูปที่ 28

รูปที่ 26 รถอัตโนมัติที่ใช้ในการทดลอง (Turing)

รูปที่ 27 วัตถุบดบัง (Obstacle object)

โดยจะเก็บ ROS bag เป็นข้อมูลตำแหน่ง ความเร็ว ที่รถอัตโนมัติเคลื่อนที่จริง นำมาพลอตกราฟในผลการทดลอง เพื่อยืนยันผล Simulation

บทที่ 4 ผลการทดลอง

ผลการทดลองจะแบ่งเป็น 2 ส่วนด้วยกัน คือผลที่ได้จาก Simulation และการทดลองโดยเลือกเหตุการณ์จาก Simulation มาทำการทดลอง โดยเหตุการณ์ที่นำมาทดลองจะเลือกตามความเหมาะสมและยึดความปลอดภัยเป็นหลัก

4.1 ผลการทดลองจาก Simulation

เบื้องต้นการทำการทดลองจะทำบน Python ก่อนเนื่องจากว่าสามารถทราบการเคลื่อนที่ของวัตถุ (Object) ทั้งรถอัตโนมัติและคนเดินเท้าได้ ซึ่งในการทดลอง Simulation จะแบ่งเป็นแต่ละสถานการณ์ที่ไม่มี Algorithm (Baseline) มี Algorithm ที่ยังไม่มีการหักเลี้ยว และมี Algorithm ที่มีการหักเลี้ยว โดยจะอธิบายผลการทดลองทั้ง 3 สถานการณ์ ในการ Simulation แต่ละอัน ดังต่อไปนี้

4.1.1 Baseline Simulation

รถอัตโนมัติวิ่งด้วยความเร็วคงที่ เมื่อเจอคนเดินเท้าจึงตัดสินใจเบรก ซึ่งกำหนดค่าอัตราหน่วงมากที่สุดที่รถสามารถทำได้เป็น -7 m/s^2 โดยจากสถานการณ์ของความเร็วรถอัตโนมัติเป็น 10-30 กม/ชม และคนเดินเท้าเป็น 1.0-2.0 เมตร/วินาที จำนวนการเกิดการชน (Collision) และเกิดการเบรกฉุกเฉิน (Emergency Braking) ของแต่ละความเร็วของรถอัตโนมัติเป็นดังตารางที่ 8 จะอธิบายเป็นค่าพิกัด (x,y) โดยที่ x คือจำนวนเหตุการณ์ที่เกิด Emergency braking และ y คือจำนวนเหตุการณ์ที่เกิดการชน (Collision)

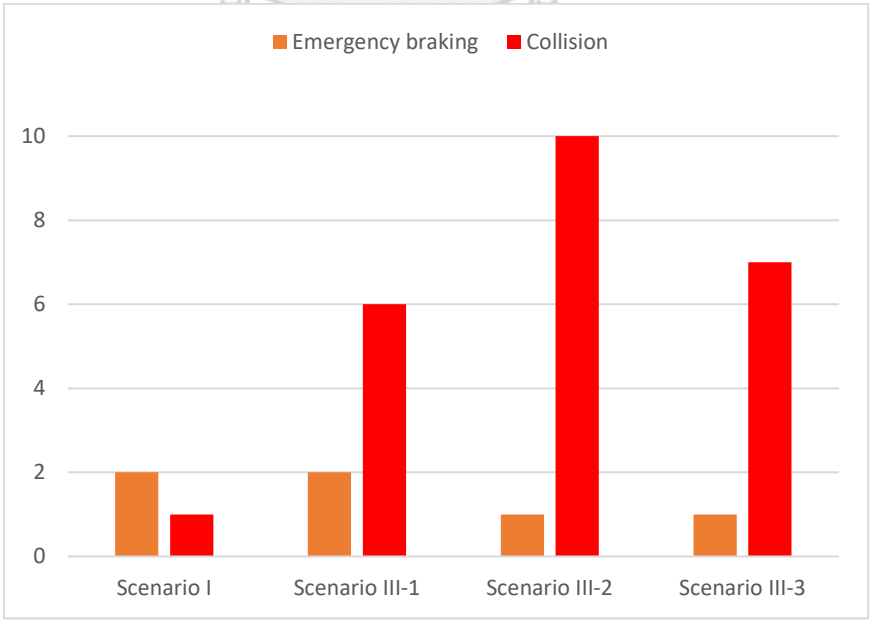
ตารางที่ 8 จำนวนการเกิด Emergency braking และเกิดการชน (collision) ใน Baseline

ความเร็วรถอัตโนมัติ (km/h)		10	15	20	25	30
สถานการณ์ที่ 1		0,0	0,0	0,0	1,0	1,1
สถานการณ์ที่ 2	1 วินาที	0,0	0,0	0,0	0,0	0,0
	2 วินาที	0,0	0,0	0,0	0,0	0,0
	3 วินาที	0,0	0,0	0,0	0,0	0,0
สถานการณ์ที่ 3	1 วินาที	0,0	0,0	1,0	1,2	0,4
	2 วินาที	0,0	0,0	1,2	0,5	0,3
	3 วินาที	0,0	1,0	0,5	0,2	0,0

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

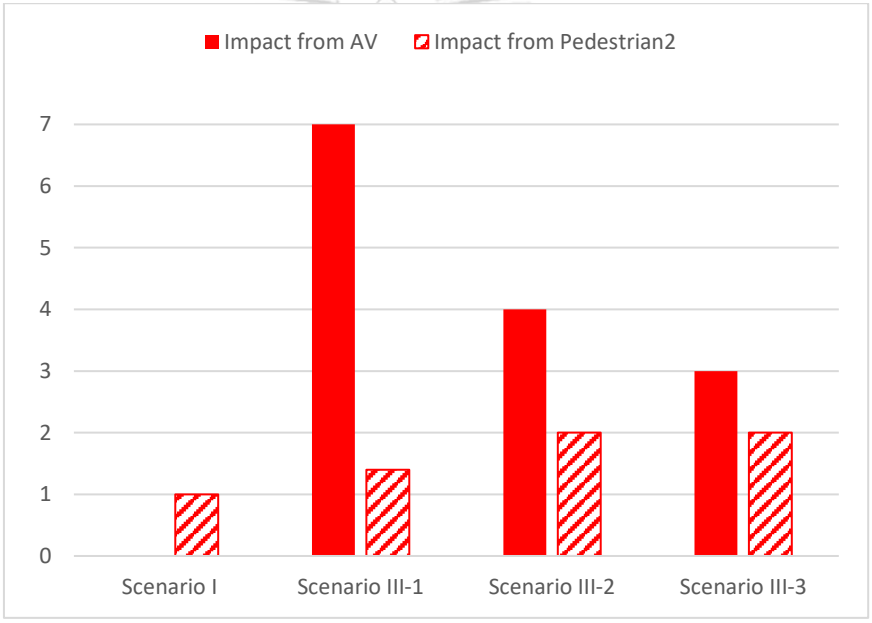
จากกราฟแท่งรูปที่ 30 เป็นกราฟแสดงจำนวนเหตุการณ์ที่เกิด Emergency braking ที่เป็นกราฟแท่ง สีน้ำเงินของทั้ง 3 สถานการณ์ โดยที่สถานการณ์ที่ II ไม่เกิด Emergency braking และไม่เกิดการชนเกิดขึ้นเลย เนื่องจากว่าเป็นสถานการณ์ที่คนเดินเท้าเดินออกจากจุดเริ่มต้นก่อน ทำให้รถอัตโนมัติสามารถเห็นคนเดินเท้าในระยะที่ไกลจาก Conflict area จึงสามารถเบรกได้ทัน แต่ในสถานกาณ์ที่ I และ III เกิด Emergency braking และเกิดการชน (Collision) รายละเอียดของแต่ละสถานการณ์เป็นดังต่อไปนี้

- สถานการณ์ที่ I : เกิด Emergency braking 2 เหตุการณ์ เกิดการชน 1 เหตุการณ์ เป็นการชนกรณีที่คนเดินเท้าชนรถอัตโนมัติ
- สถานการณ์ที่ III-1 เกิด Emergency braking 2 เหตุการณ์ เกิดการชน 6 เหตุการณ์ โดยที่ 3 เหตุการณ์ เป็นการชนกรณีรถอัตโนมัติชนคนเดินเท้า และอีก 3 เหตุการณ์ เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติ
- สถานการณ์ที่ III-2 เกิด Emergency braking 1 เหตุการณ์ เกิดการชน 10 เหตุการณ์ โดยที่ 2 เหตุการณ์ เป็นการชนกรณีรถอัตโนมัติชนคนเดินเท้า และอีก 8 เหตุการณ์ เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติ
- สถานการณ์ที่ III-3 เกิด Emergency braking 7 เหตุการณ์ เกิดการชน 7 เหตุการณ์ โดยที่ 1 เหตุการณ์ เป็นการชนกรณีรถอัตโนมัติชนคนเดินเท้า และอีก 6 เหตุการณ์ เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติ



รูปที่ 28 จำนวนการเกิด Emergency braking และการชน (Collision)

ความรุนแรงที่เกิดการชนเป็นปัจจัยที่เราควรต้องพิจารณา การชนทำให้เกิดความเสียหาย ซึ่งความรุนแรงในการชนอาจจะทำให้เกิดการสูญเสียถึงชีวิต ดังนั้นกราฟรูปที่ 31 แสดงความเร็วสูงสุดที่รถอัตโนมัติชนคนเดินเท้าในสถานการณ์ที่ III-1 รถอัตโนมัติชนคนเดินเท้าที่ความเร็วสูงสุดที่เกิดการชน 7 m/s หรือ 25 กม/ชม ในสถานการณ์ที่ III-2 รถอัตโนมัติชนคนเดินเท้าที่ความเร็วสูงสุดที่เกิดการชน 4 m/s หรือ 14.4 กม/ชม และในสถานการณ์ที่ III-3 รถอัตโนมัติชนคนเดินเท้าที่ความเร็วสูงสุดที่เกิดการชน 3 m/s หรือ10.8 กม/ชม ส่วนกรณีที่คนเดินเท้าชนรถอัตโนมัติ เกิดในสถานการณ์ที่ I ชนที่ความเร็วคนเดินเท้าสูงสุดที่เกิดขึ้นเป็น 1 m/s ในสถานการณ์ที่ III-1 ชนที่ความเร็วคนเดินเท้าสูงสุดที่เกิดขึ้นเป็น 1.4 m/s ในสถานการณ์ที่ III-2 ชนที่ความเร็วคนเดินเท้าสูงสุดที่เกิดขึ้นเป็น 2 m/s และในสถานการณ์ที่ III-3 ชนที่ความเร็วคนเดินเท้าสูงสุดเป็น 2 m/s



รูปที่ 29 ความเร็วขณะเกิดการชนของรถอัตโนมัติและคนเดินเท้า

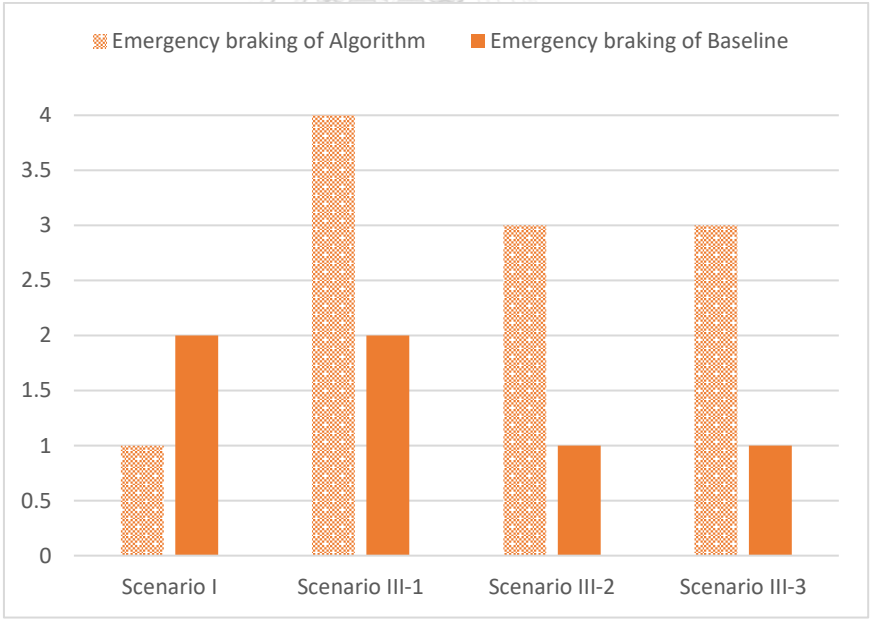
ความเร็วที่รถอัตโนมัติชนคนเดินเท้าดังกล่าว ถือว่าเป็นความเร็วที่อันตราย หากเราสามารถลดความเร็วที่ทำให้เกิดการชนให้น้อยลงได้ ความรุนแรงของอุบัติเหตุจะน้อยลงตามไปด้วย

4.1.2 With Algorithm (Without Steering)

ในวิธีการ Algorithm ที่นำเสนอให้ชะลอความเร็วก่อน เมื่อเจอคนเดินเท้าแล้วจึงคำนวนค่า TTB และ TTC เพื่อนำมาพิจารณาในการตัดสินใจที่จะเบรก หรือ Emergency brake ได้ผลลัพธ์ดังตารางที่ 7 เป็นตารางที่แสดงจำนวนเหตุการณ์ที่เกิด Emergency braking และเกิดการชน (Collision) ในการใช้ Algorithm ที่นำเสนอที่มีเฉพาะการชะลอความเร็ว เบรก และ Emergency brake นำเสนอเป็นค่าพิกัดดังตารางที่ 9

ตารางที่ 9 จำนวนการเกิด Emergency braking และเกิดการชน (collision) ที่ใช้ Algorithm

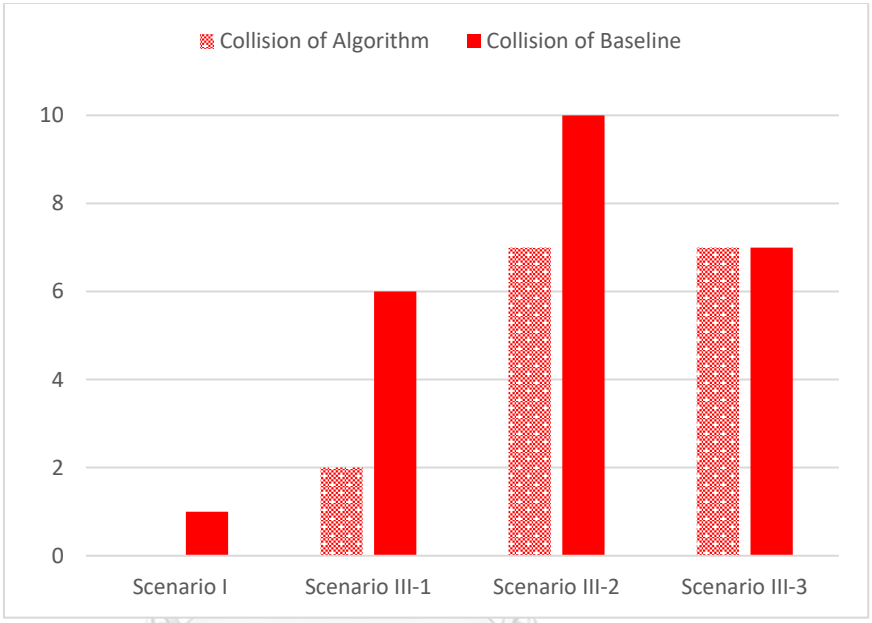
ความเร็วรถอัตโนมัติ (km/h)		10	15	20	25	30
สถานการณ์ที่ 1		0,0	0,0	0,0	0,0	1,0
สถานการณ์ที่ 2	1 วินาที	0,0	0,0	0,0	0,0	0,0
	2 วินาที	0,0	0,0	0,0	0,0	0,0
	3 วินาที	0,0	0,0	0,0	0,0	0,0
สถานการณ์ที่ 3	1 วินาที	0,0	0,0	0,0	0,0	4,2
	2 วินาที	0,0	0,0	0,0	3,3	0,4
	3 วินาที	0,0	0,0	3,2	0,4	0,1



รูปที่ 30 จำนวนเหตุการณ์ที่เกิด Emergency braking ของการใช้ Algorithm เปรียบเทียบกับ Baseline

กราฟแท่งรูปที่ 32 แสดงจำนวนเหตุการณ์ที่เกิด Emergency braking ของ Baseline Simulation และการใช้ Algorithm ที่งานวิจัยนี้นำเสนอ พบว่าสามารถลด Emergency brake ได้ในสถานการณ์ที่ 1 แต่สถานการณ์ที่ III ไม่สามารถลดจำนวน Emergency brake ได้ เพราะการชนที่

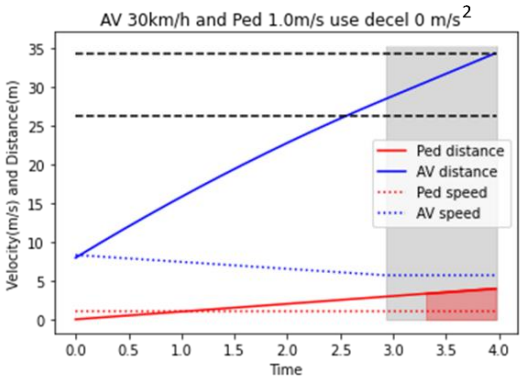
เกิดขึ้นใน Baseline simulation จะกลายมาเป็น Emergency braking ของการที่เราใช้ Algorithm เป็นการตัดสินใจแทน และในกราฟแท่งรูปที่ 33 ที่แสดงจำนวนเหตุการณ์การชน (Collision) ของการใช้ Algorithm ที่ออกแบบเปรียบเทียบกับ Baseline สามารถลดจำนวนการเกิดการชนได้ และลดการชนในกรณีที่รถอัตโนมัติชนคนเดินเท้าได้ 100% ในทุกกรณี แต่ยังเกิดการชนของคนเดินเท้าที่ชนรถอัตโนมัติอยู่



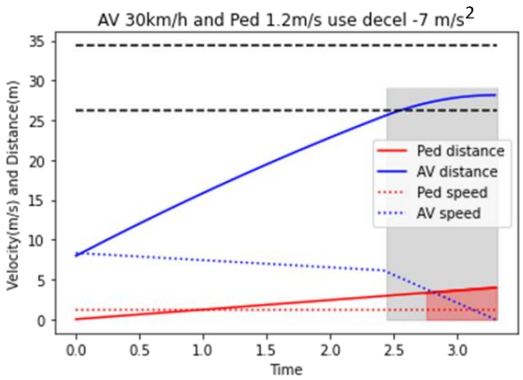
รูปที่ 31 จำนวนเหตุการณ์ที่เกิดการชน (Collision) ของการใช้ Algorithm เปรียบเทียบกับ Baseline

จากการชนทั้งหมดที่เกิดขึ้นเป็นการชนในกรณีที่คนเดินเท้าชนรถอัตโนมัติ จะแสดงเหตุการณ์ทุกเหตุการณ์ที่เกิดการชน เส้นประสีดำแสดงถึงตำแหน่งที่รถอัตโนมัติเข้าสู่ Conflict area และส่วนที่แรเงาสีแดงแสดงถึงเวลาที่คนเดินเท้าเข้าสู่ Conflict area ซึ่งหากรถอัตโนมัติเข้าสู่ Conflict area ขณะที่คนเดินเท้าอยู่ใน Conflict area หรือคนเดินเท้าเข้าสู่ Conflict area ขณะที่รถอัตโนมัติอยู่ใน Conflict area แสดงว่าเกิดการชนเกิดขึ้น ซึ่งมีรายละเอียดในแต่ละสถานการณ์ต่อไปนี้

- สถานการณ์ที่ III-1: จากกราฟรูปที่ 34 เป็นกรณีการชนในกรณีที่คนเดินเท้าชนรถอัตโนมัติ เนื่องจากรถอัตโนมัติถึง Conflict area ก่อนที่จะตรวจเจอคนเดินเท้าที่กำลังจะข้ามถนน การชนที่เกิดขึ้นเนื่องจากคนเดินเท้าเดินด้วยความเร็วคงที่และไม่หยุดเดิน ซึ่งถือว่าเป็นการชนที่ไม่สามารถหลีกเลี่ยงได้ของรถอัตโนมัติ



รูปที่ 34a เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-1 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 1 m/s

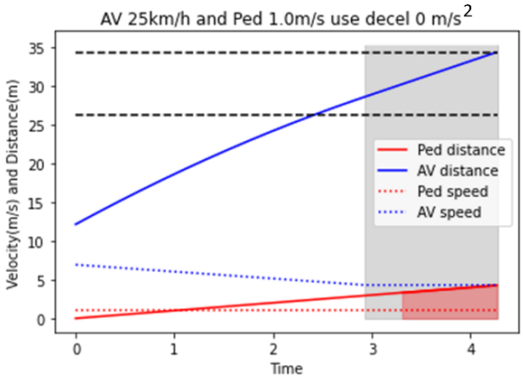


รูปที่ 34b เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-1 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 1 m/s

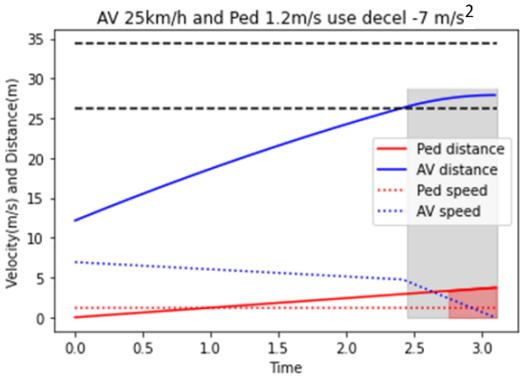
รูปที่ 32 เหตุการณ์การชนในสถานการณ์ที่ III-1

- สถานการณ์ที่ III-2: จากกราฟรูปที่ 35a, 35d, 35e และ 35f เป็นกรณีการชนของคนเดินเท้าชนรถอัตโนมัติ เพราะรถอัตโนมัติเข้าสู่ Conflict area ก่อนที่จะสามารถตรวจจับคนเดินเท้าได้ จึงทำให้รถอัตโนมัติไม่สามารถตัดสินใจเบรกได้ ส่วนกราฟรูปที่ 35b และ 35g เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติเช่นกัน โดยที่รถอัตโนมัติถึง Conflict area พอดีกับที่สามารถตรวจจับคนเดินเท้าได้ จึงมีการเบรกแบบ Emergency braking เกิดขึ้น และกราฟรูปที่ 35c เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติเช่นกัน โดยที่รถอัตโนมัติเห็นคนเดินเท้าก่อนแล้วตัดสินใจเบรกแบบ Emergency braking โดยหยุดสนิทที่ขอบ Conflict area พอดี ซึ่งจะถือว่าเป็นสถานการณ์ที่เสี่ยงที่จะเกิดอุบัติเหตุสูง

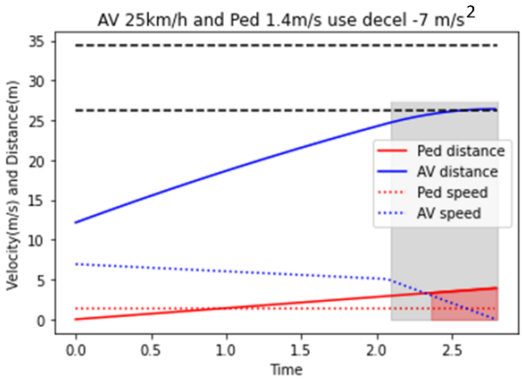
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



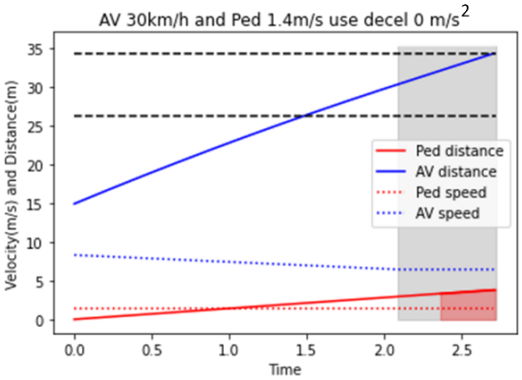
รูปที่ 35a เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1 m/s



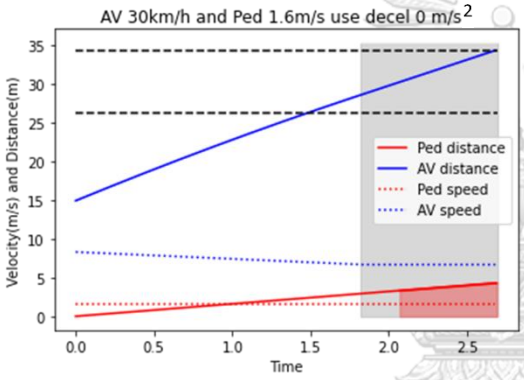
รูปที่ 35b เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.2 m/s



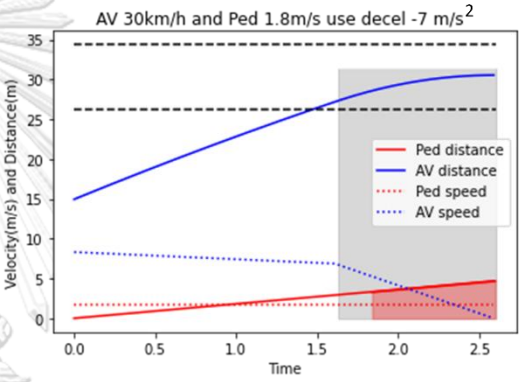
รูปที่ 35c เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.4 m/s



รูปที่ 35d เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 1.4 m/s

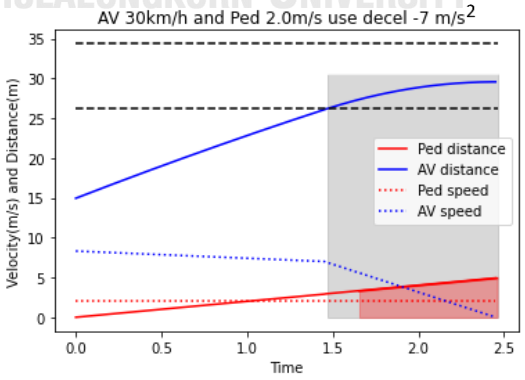


รูปที่ 35e เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 1.6 m/s



รูปที่ 35f เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 1.8 m/s

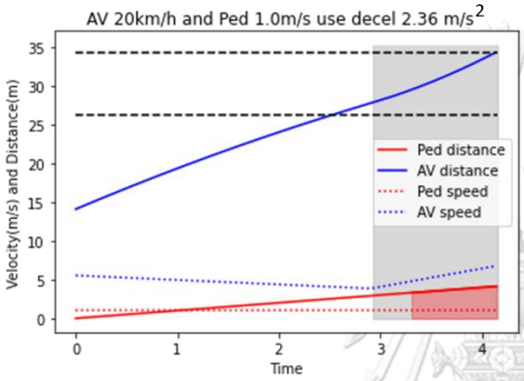
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



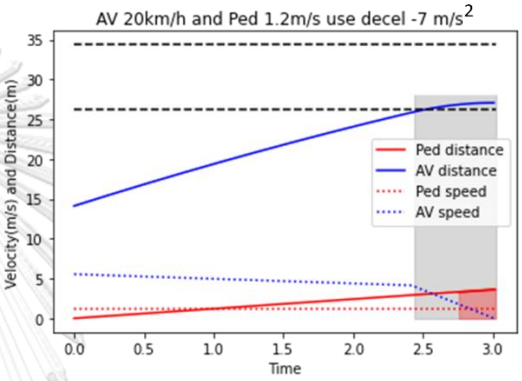
รูปที่ 35g เหตุการที่เกิดการชนในสถานการณ์ที่ III-2 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 2 m/s

รูปที่ 33 เหตุการณ์การชนในสถานการณ์ที่ III-2

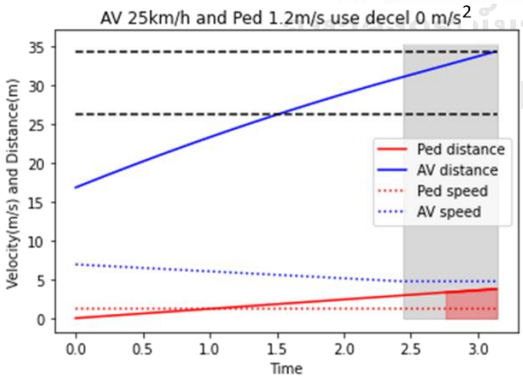
- สถานการณ์ที่ III-3: จากกราฟรูปที่ 36a, 36c, 36d, 36e และ 35h เป็นกรณีการชนของคนเดินเท้าชนรถอัตโนมัติ เพราะรถอัตโนมัติเข้าสู่ Conflict area ก่อนที่จะสามารถตรวจจับคนเดินเท้าได้ จึงทำให้รถอัตโนมัติไม่สามารถตัดสินใจเบรกได้ ส่วนกราฟรูปที่ 36b, 36f และ 36g เป็นการชนกรณีคนเดินเท้าชนรถอัตโนมัติเช่นกัน โดยที่รถอัตโนมัติถึง Conflict area พอดีกับที่สามารถตรวจจับคนเดินเท้าได้ จึงมีการเบรกแบบ Emergency braking เกิดขึ้น เนื่องจากคนเดินเท้าเดินด้วยความเร็วคงที่และไม่หยุดเดินจึงทำให้เกิดการชนเกิดขึ้น



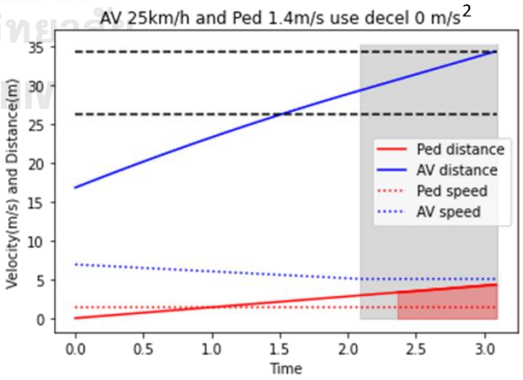
รูปที่ 36a เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 20 กม/ชม และคนเดินเท้าเป็น 1 m/s



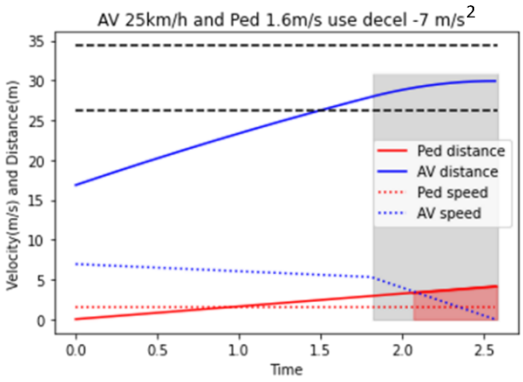
รูปที่ 36b เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 20 กม/ชม และคนเดินเท้าเป็น 1.2 m/s



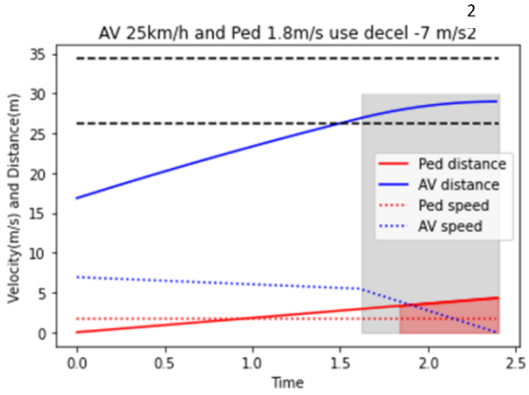
รูปที่ 36c เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.2 m/s



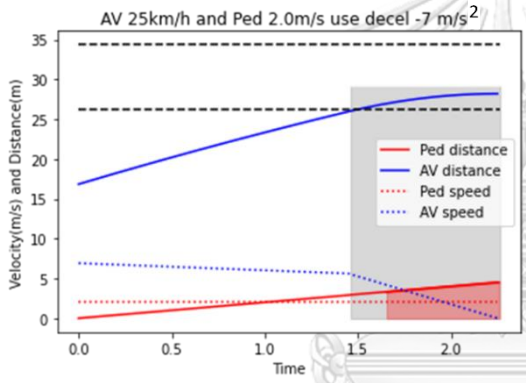
รูปที่ 36d เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.4 m/s



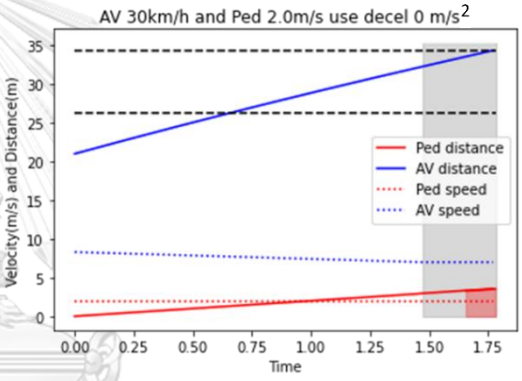
รูปที่ 36e เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.6 m/s



รูปที่ 36f เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 1.8 m/s



รูปที่ 36g เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 25 กม/ชม และคนเดินเท้าเป็น 2 m/s

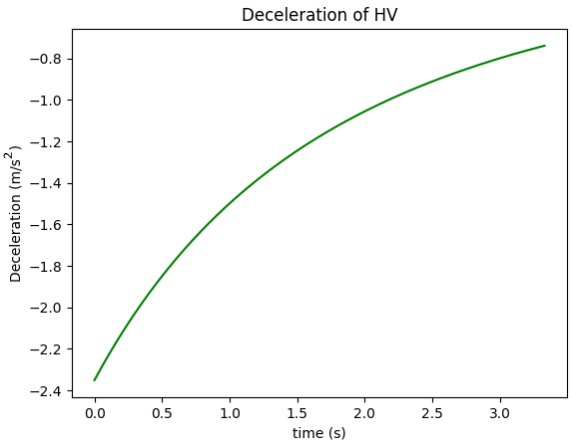


รูปที่ 36h เหตุการณ์ที่เกิดการชนในสถานการณ์ที่ III-3 ที่ความเร็วรถอัตโนมัติเป็น 30 กม/ชม และคนเดินเท้าเป็น 2 m/s

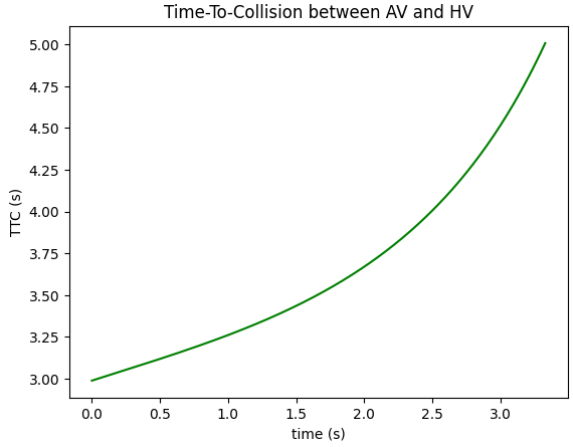
รูปที่ 34 เหตุการณ์การชนในสถานการณ์ที่ III-3

ทั้ง 3 สถานการณ์นี้ที่เกิด Action การเบรกหรือ Emergency braking ของรถอัตโนมัติ กำหนดความเร็วสูงสุดของยานพาหนะ HV1 ไว้ที่ 30 กม/ชม เนื่องจากเป็นการเดินทางในมหาวิทยาลัย จึงไม่ควรขับขี่ด้วยความเร็วสูง ดังนั้นเหตุการณ์ที่ยานพาหนะ 2 คัน มีความเร็วต่างกันมาก คือที่ความเร็วรถอัตโนมัติเป็น 10 กม/ชม ที่กำลังชะลอความเร็ว และยานพาหนะ HV1 มีความเร็ว 30 กม/ชม กำลังรักษา Time Headway เป็น 2 วินาที กับรถอัตโนมัติ ถูกนำมาเสนอค่าอัตราหน่วงที่ยานพาหนะ HV1 ใช้เพื่อรักษา Time Headway ดังกล่าว แสดงค่า $TTC_{AV,HV1}$ และแสดงความเร็วของทั้งยานพาหนะ HV1 และรถอัตโนมัติ ดังรูปที่ 37 38 และ 39 ตามลำดับ โดยจะเห็นว่าอัตราหน่วงสูงสุดที่ยานพาหนะ HV1 ใช้เพื่อรักษา Time Headway ไม่เกิน -2.5 m/s^2 และค่า $TTC_{AV,HV1}$ ที่น้อยที่สุดไม่ต่ำกว่า 3 วินาที ซึ่งไม่ได้อยู่ในเกณฑ์ที่อันตราย

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

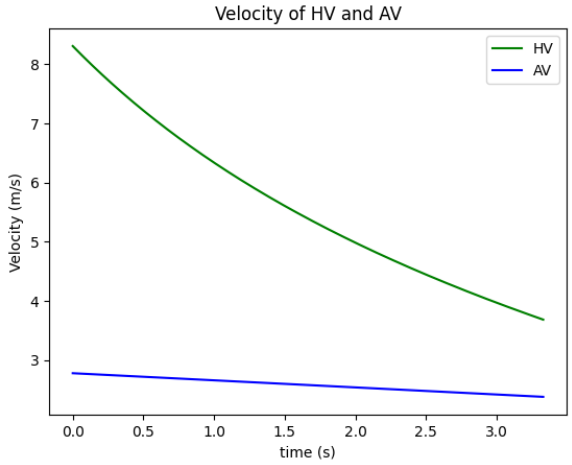


รูปที่ 35 อัตราหน่วงที่ยานพาหนะ HV1 ใช้เพื่อรักษา Time Headway 2 วินาที



รูปที่ 36 ค่า $TTC_{AV,HV1}$ ที่รักษา Time Headway เป็น 2 วินาที

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 37 ความเร็วของยานพาหนะ HV1 และ AV ที่รักษา Time Headway 2 วินาที

จากข้อมูลกราฟที่ 34, 35 และ 36 ที่เกิดการชนทั้งหมดจะนำมาเปรียบเทียบกับกรณีที่ไม่ได้ใช้ Algorithm กับการใช้ Algorithm ที่ยังไม่มีการเปลี่ยนเลน ดังกราฟแท่งรูปที่ 33 พบว่าสามารถลดจำนวนการเกิดอุบัติเหตุในกรณีที่รถอัตโนมัติชนคนเดินเท้าได้ 100% แต่จำนวนการใช้ Emergency braking อาจเพิ่มขึ้นในสถานการณ์ที่ III เนื่องจากว่ารถสามารถตรวจจับคนเดินเท้าได้ในระยะกระชั้นชิดกับ Conflict area และกรณีการชนที่คนเดินเท้าชนรถอัตโนมัติไม่ได้มีจำนวนที่ลดลง เนื่องจากเป็นกรณีที่คนเดินเท้าชนเราไม่สามารถทำอะไรได้ ดังนั้นการเปลี่ยนเลนด้วยความเร็วและระยะทางที่เหมาะสมเพื่อเพิ่มระยะการมองเห็นให้คนเดินเท้าจึงอีกแนวทางหนึ่งในการช่วยลดความเสี่ยงที่จะเกิดอุบัติเหตุในกรณีคนเดินเท้าชนรถอัตโนมัติได้

4.1.3 With Algorithm (With Steering)

การเปลี่ยนเลนเรากำหนดให้มีความเร็วคงที่ขณะเปลี่ยนเลน โดยกำหนดความเร็วต่ำที่สุดขณะเปลี่ยนเลนอยู่ที่ 10 km/h ดังนั้นเราจะนำการเบรกจากหัวข้อ 4.1.2 ที่เป็นการใช้ Algorithm แต่ยังไม่มีการใช้การเปลี่ยนเลน มาดูผลลัพธ์ เริ่มจากความเร็วสูงสุดไปน้อยที่สุด

ตารางที่ 10 แสดงช่วงการวิ่งใน Comfort-Aggressive zone, Emergency braking และการเกิดการชน (Collision) ที่ยังไม่มีการเปลี่ยนเลน ของความเร็วรถอัตโนมัติเป็น 30 กม/ชม โดยสีเขียวแสดงถึงการกระทำ (Action) ของรถอัตโนมัติที่เลือกเบรกอยู่ในช่วง Comfort zone ไปจนถึง Aggressive zone, สีส้มแสดงถึงการเบรกแบบ Emergency brake และสีแดงแสดงถึงการชน (Collision)

ตารางที่ 10 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 30 กม/ชม

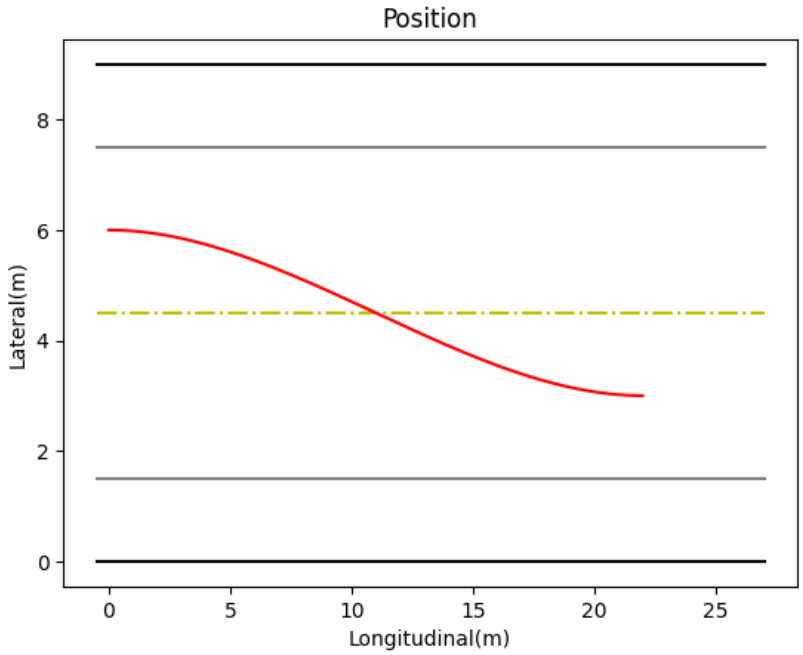
	II-3	II-2	II-2	I	III-1	III-2	III-3
1							
1.2							
1.4							
1.6							
1.8							
2							

จะเห็นว่าจะมีช่วงที่มี Emergency brake และเกิดการชน (Collision) อยู่ในสถานการณ์ที่ I และ III ดังนั้นเราจะประเมิณระยะที่รถอัตโนมัติที่มีความเร็วต้นที่ 30 กม/ชม ควรจะเปลี่ยนเลน เพื่อเพิ่มการมองเห็นให้กับคนเดินเท้า เนื่องจากว่ากรณีที่เกิดการชนทั้งหมดเกิดเฉพาะกรณีที่คนเดินเท้าชนรถอัตโนมัติ และช่วยลด Emergency brake เราจึงต้องการการหักเลี้ยวในสถานการณ์ที่ I ที่ความเร็วคนเดินเท้าเป็น 1 m/s และในสถานการณ์ที่ III-1 ทุกความเร็วของคนเดินเท้า ดังนั้นจึงต้องหาระยะห่างจาก Conflict area ที่เหมาะสม เพื่อให้รถอัตโนมัติเปลี่ยนเลน หากรถอัตโนมัติยังไม่ตรวจเจอคนเดินเท้าก่อนระยะดังกล่าว จะทำการเปลี่ยนเลนเพื่อเพิ่มระยะการมองเห็นให้คนเดินเท้า โดยระยะทางที่เหมาะสมที่ห่างจาก Conflict area ดังกล่าวคำนวณจากสถานการณ์ที่ I ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 1.2 m/s จะทราบเวลาที่คนเดินเท้าเดินเข้ามาในเลนของรถอัตโนมัติ ($t_{ped,in\,road}$) จากสมการที่ 23

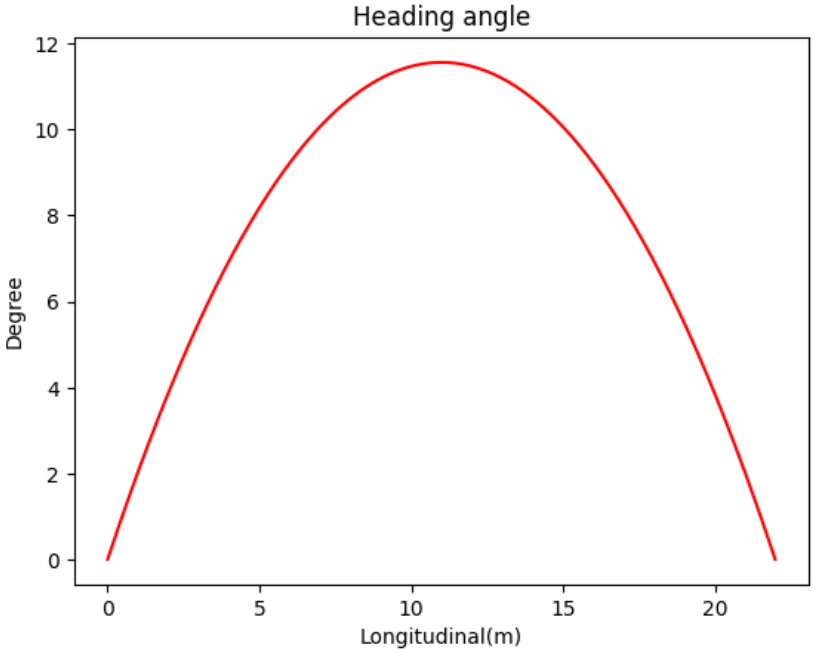
$$t_{ped,in\,road} = \frac{S_{ped,in\,road}}{V_{ped}} \tag{23}$$

โดยที่ค่า $S_{ped,in\,road}$ ที่คนเดินเท้าจะเข้ามาในเลนของรถอัตโนมัติเป็น 2.92 เนื่องจากระยะจากจุดเริ่มต้นของคนเดินเท้าห่างจากรถอัตโนมัติ 4.42 เมตร ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 1.2 m/s จะได้ค่า $t_{ped,in\,road}$ เป็น 2.43 วินาที ซึ่งที่เวลา 2.43 วินาที รถอัตโนมัติเคลื่อนที่มาได้ระยะทาง 17.61 เมตร จากจุดเริ่มต้น โดยห่างจาก Conflict area 9.05 เมตร กล่าวคือที่ระยะ 9.05 เมตร จาก Conflict area หากคนเดินเท้าเดินด้วยความเร็ว 1.2 m/s รถอัตโนมัติจะตรวจเจอคนเดินเท้า แต่ถ้าหากคนเดินเท้าเดินด้วยความเร็วที่ช้ากว่านั้น รถอัตโนมัติจะไม่พบคนเดินเท้า ในสถานการณ์ที่ I ดังนั้นจึงเลือกระยะ 9.05 เมตร จาก Conflict area เป็นระยะที่เหมาะสมในการเปลี่ยนเลนของรถอัตโนมัติ ซึ่งที่ความเร็วต้นของรถอัตโนมัติเป็น 30 กม/ชม ที่ระยะดังกล่าวรถอัตโนมัติชะลอความเร็วจนเหลือ 22.12 กม/ชม ดังนั้นจะเปลี่ยนเลนด้วยความเร็วนี้เลย

ระยะทางทั้งหมดในการเปลี่ยนเลนในแนว Longitudinal ที่ความเร็วต้นของรถอัตโนมัติเป็น 30 กม/ชม คือ 22 เมตร เป็นระยะทางที่น้อยที่สุดที่ทำให้ค่าอัตราหน่วง และ Jerk ทั้งแนว Lateral และแนว Longitudinal ไม่เกินขอบเขตที่กำหนด ซึ่งระยะทางในการเปลี่ยนเลนในแนว Lateral เป็น 3 เมตร ตามความกว้างของถนน 1 เลน ดังรูปที่ 40 และสามารถหา Heading angle ของเส้นทางที่เปลี่ยนเลนดังรูปที่ 41 ในหน่วยองศา มี Heading angle มากที่สุดไม่เกิน 12 องศา



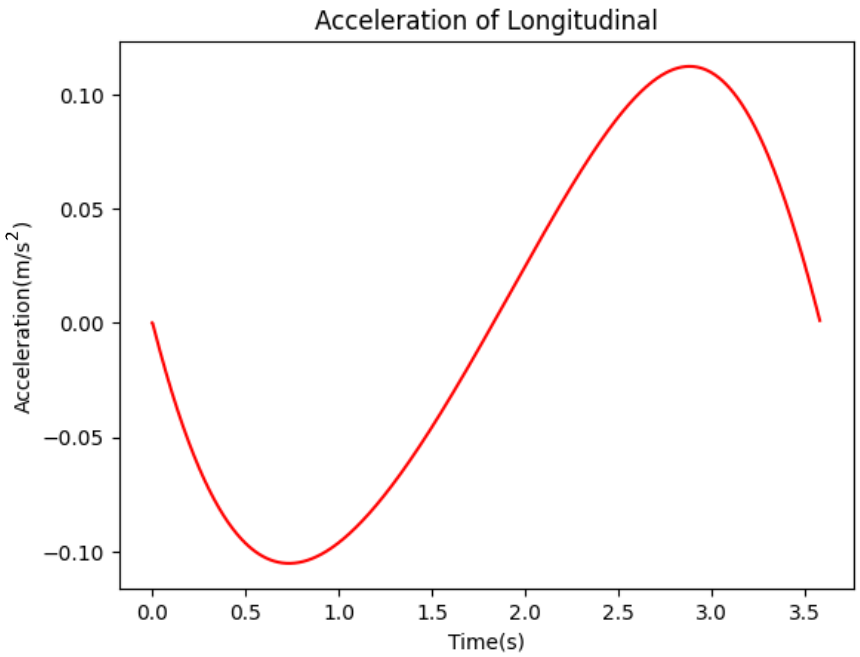
รูปที่ 38 เส้นทางการเปลี่ยนเลนของรถอัตโนมัติ



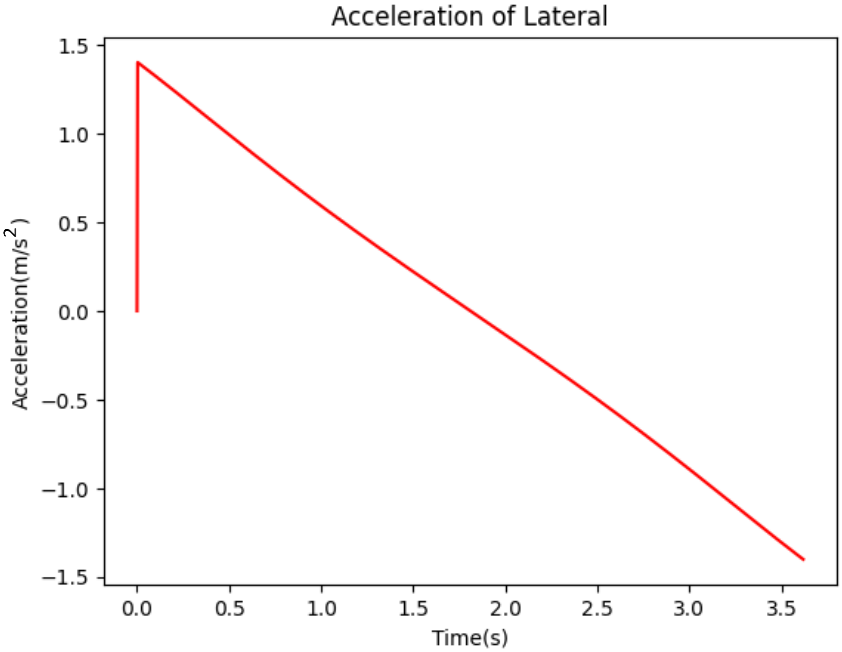
รูปที่ 39 Heading angle ของเส้นทางที่เปลี่ยนเลน

เมื่อหา Heading angle ได้ จะสามารถหาความเร็วในแนว Lateral และแนว Longitudinal ได้ ซึ่งทำให้หาอัตราเร่งกับอัตราหน่วงในแนว Lateral และ Longitudinal ได้ดังกราฟรูปที่ 42 และ

43 โดยความชันทั้งสองกราฟคือ Jerk ทั้งสองแนวเช่นกัน และ Jerk ทั้งสองแนวจากกราฟไม่เกิน 2.0 m/s^3 ถือว่าอยู่ในขอบเขตของการหักเลี้ยวที่กำหนด



รูปที่ 40 ค่าอัตราเร่งและอัตราหน่วงในแนว Longitudinal



รูปที่ 41 ค่าอัตราเร่งและอัตราหน่วงในแนว Lateral

ตารางที่ 11 แสดงช่วงการวิ่งใน Comfort-Aggressive zone, Emergency braking และการเกิดการชน (Collision) ที่ยังไม่มีการเปลี่ยนเลน ของความเร็วรถอัตโนมัติเป็น 25 กม/ชม โดยสีเขียวแสดงถึงการกระทำ (Action) ของรถอัตโนมัติที่เลือกเบรกอยู่ในช่วง Comfort zone ไปจนถึง Aggressive zone, สีส้มแสดงถึงการเบรกแบบ Emergency brake และสีแดงแสดงถึงการชน (Collision) เช่นเดียวกับตารางที่ 10

ตารางที่ 11 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 25 กม/ชม

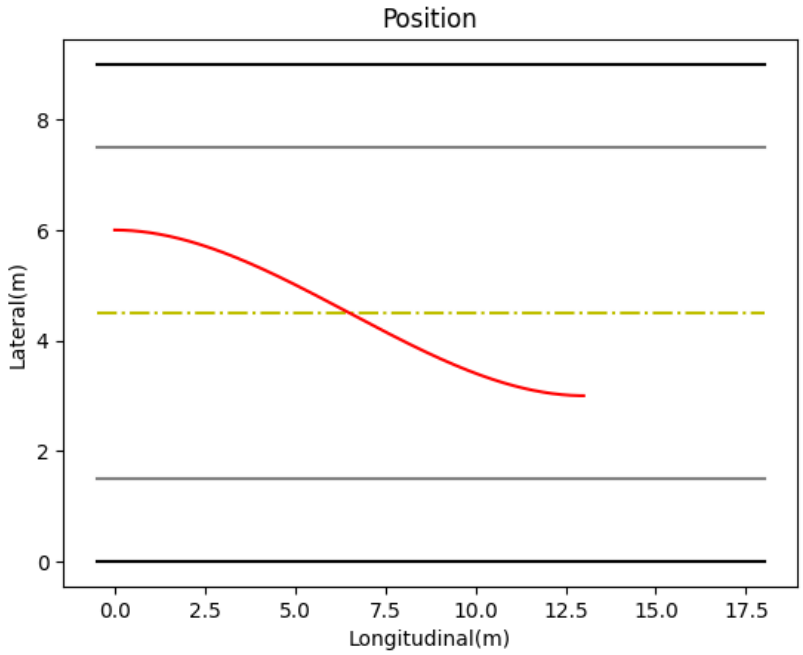
	II-3	II-2	II-2	I	III-1	III-2	III-3
1							
1.2							
1.4							
1.6							
1.8							
2							

จะเห็นว่าจะมีช่วงที่มี Emergency brake และเกิดการชน (Collision) อยู่ในสถานการณ์ที่ III-2 และ III-3 ดังนั้นเราจะประเมิณระยะที่รถอัตโนมัติที่มีความเร็วต้นที่ 25 กม/ชม ที่ควรจะเปลี่ยนเลน เพื่อเพิ่มการมองเห็นให้กับคนเดินเท้า เนื่องจากว่ากรณีที่เกิดการชนทั้งหมดเกิดเฉพาะกรณีที่คนเดินเท้าชนรถอัตโนมัติ และช่วยลด Emergency brake เราจึงต้องการการหักเลี้ยวในสถานการณ์ที่ III-2 ดังนั้นจึงต้องหาระยะห่างจาก Conflict area ที่เหมาะสม เช่นเดียวกับที่ความเร็วต้นของรถอัตโนมัติเป็น 30 กม/ชม โดยระยะทางที่เหมาะสมที่ห่างจาก Conflict area คำนวณจากสถานการณ์ที่ III-2 ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 2 m/s จะทราบเวลาที่คนเดินเท้าเดินเข้ามาในเลนของรถอัตโนมัติ ($t_{ped,in\ road}$) จากสมการที่ 23

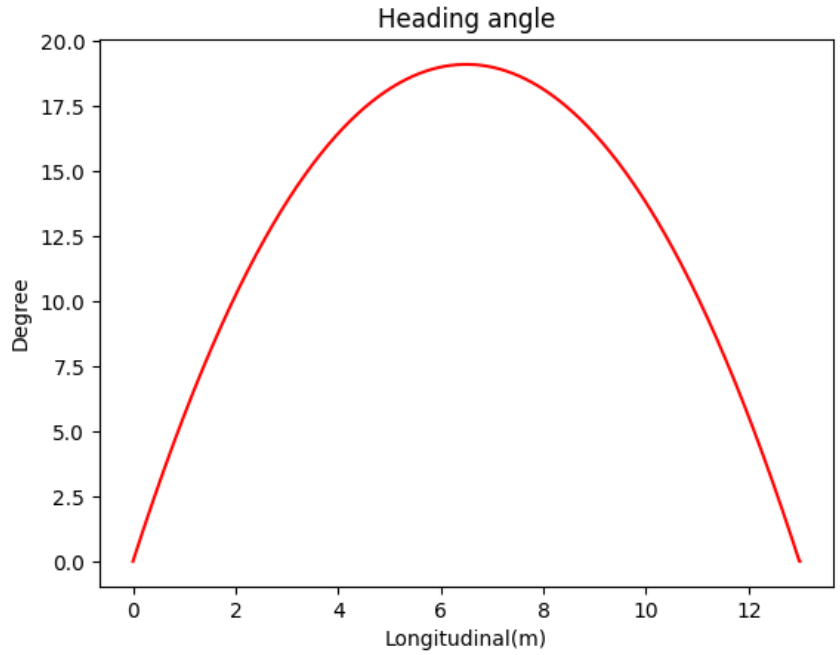
โดยที่ค่า $S_{ped,in\ road}$ ที่คนเดินเท้าจะเข้ามาในเลนของรถอัตโนมัติเป็น 2.92 เมตร ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 2 m/s จะได้ค่า $t_{ped,in\ road}$ เป็น 1.46 วินาที ซึ่งที่เวลาคนเดินเท้าเคลื่อนที่ได้ 1.46 วินาที ในสถานการณ์ที่ III-2 รถอัตโนมัติจะใช้ระยะเวลา 3.46 วินาที เนื่องจากเป็นสถานการณ์ที่รถออกจากจุดเริ่มต้นก่อนคนเดินเท้า 2 วินาที ในเวลาของรถอัตโนมัติดังกล่าว ระยะทางที่รถอัตโนมัติเคลื่อนที่ได้คือ 18.64 เมตร จากจุดเริ่มต้น โดยห่างจาก Conflict area 8.02 เมตร กล่าวคือที่ระยะ 8.02 เมตร จาก Conflict area หากคนเดินเท้าเดินด้วยความเร็ว 2

m/s รถอัตโนมัติจะตรวจเจอคนเดินเท้า ดังนั้นเราจึงเลือกระยะ 8.06 เมตร จาก Conflict area เป็นระยะที่เหมาะสมในการเปลี่ยนเลนของรถอัตโนมัติ เนื่องจากระยะดังกล่าวหากคนเดินเท้าเคลื่อนที่ด้วยความเร็ว 2 m/s ในสถานการณ์ที่ III-2 รถอัตโนมัติจะยังพบคนเดินเท้า ซึ่งที่ความเร็วของคนเดินเท้าที่ช้ากว่า 2 m/s ก็จะยังไม่พบเช่นกัน และที่ความเร็วต้นของรถอัตโนมัติเป็น 25 กม/ชม ในระยะดังกล่าวรถอัตโนมัติชะลอความเร็วจนเหลือ 13.82 กม/ชม ดังนั้นจะเปลี่ยนเลนด้วยความเร็วนี้เลย

ระยะทางทั้งหมดในการเปลี่ยนเลนในแนว Longitudinal ที่ความเร็วต้นของรถอัตโนมัติเป็น 25 กม/ชม เป็น 13 เมตร และระยะทางในการเปลี่ยนเลนในแนว Lateral เป็น 3 เมตร ตามความกว้างของถนน 1 เลน ดังรูปที่ 44 และสามารถหา Heading angle ของเส้นทางที่เปลี่ยนเลนดังรูปที่ 45 ในหน่วยองศา มี Heading angle มากที่สุดไม่เกิน 20 องศา

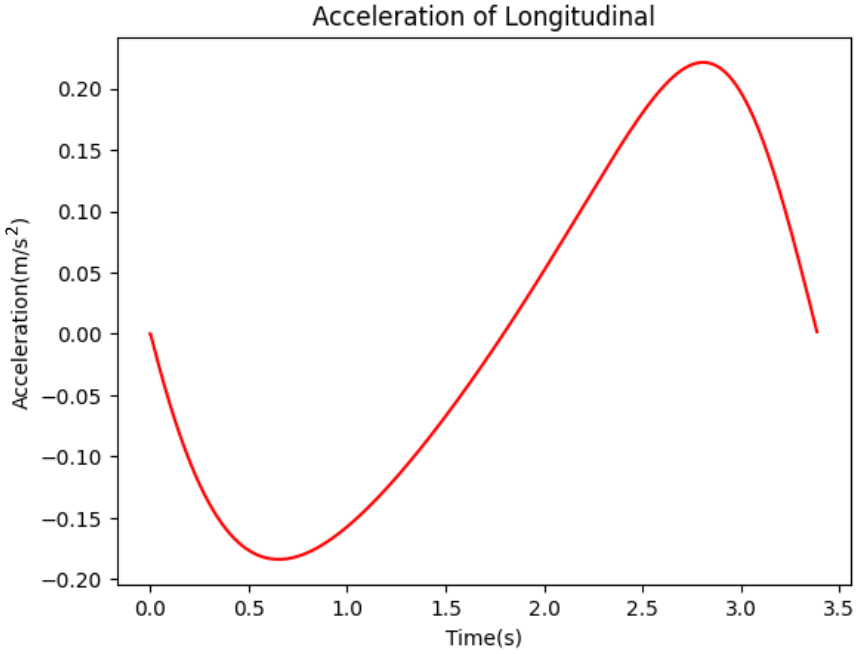


รูปที่ 42 เส้นทางการเปลี่ยนเลนของรถอัตโนมัติ

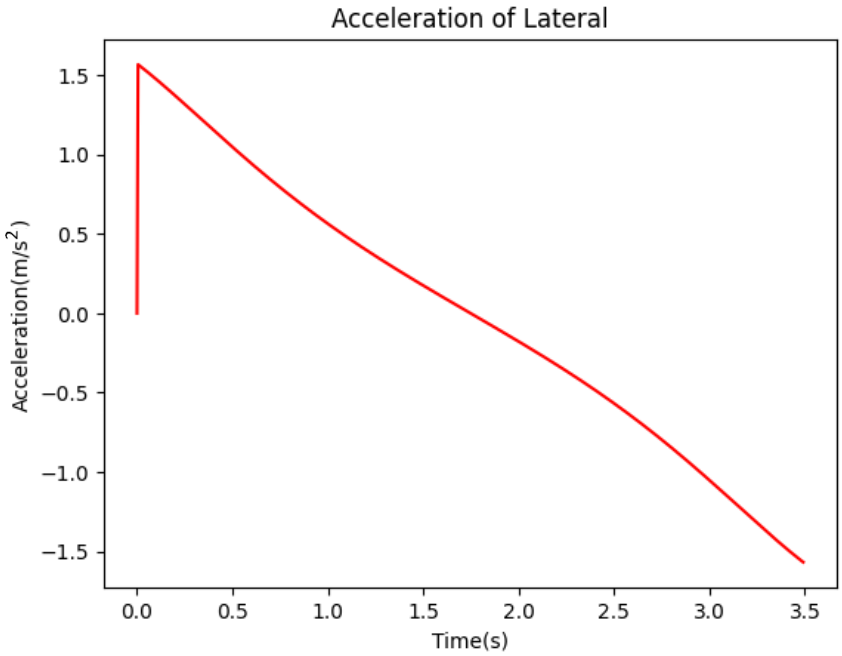


รูปที่ 43 Heading angle ของเส้นทางที่เปลี่ยนเลน

เมื่อหา Heading angle ได้ จะสามารถหาความเร็วทั้งในแนว Lateral และแนว Longitudinal และหาอัตราเร่งกับอัตราหน่วงในแนว Lateral และ Longitudinal ได้ดังกราฟรูปที่ 46 และ 47 ซึ่งความชันทั้งสองกราฟคือ Jerk ทั้งสองแนวเช่นกัน โดยที่ Jerk ทั้งสองแนวจากกราฟไม่เกิน 2.0 m/s^3 นั่นแปลว่าอยู่ในขอบเขตของการหักเลี้ยว



รูปที่ 44 ค่าอัตราเร่งและอัตราหน่วงในแนว Longitudinal



รูปที่ 45 ค่าอัตราเร่งและอัตราหน่วงในแนว Lateral

ตารางที่ 12 แสดงช่วงการวิ่งใน Comfort-Aggressive zone, Emergency braking และการเกิดการชน (Collision) ที่ยังไม่มีการเปลี่ยนเลน ของความเร็วรถอัตโนมัติเป็น 20 กม/ชม โดยสีเขียวแสดงถึงการกระทำ (Action) ของรถอัตโนมัติที่เลือกเบรกอยู่ในช่วง Comfort zone ไปจนถึง Aggressive zone, สีส้มแสดงถึงการเบรกแบบ Emergency brake และสีแดงแสดงถึงการชน (Collision) เช่นเดียวกับตารางที่ 10 และ 11

ตารางที่ 12 การกระทำ (Action) ของรถอัตโนมัติที่ความเร็ว 20 กม/ชม

	II-3	II-2	II-2	I	III-1	III-2	III-3
1							
1.2							
1.4							
1.6							
1.8							
2							

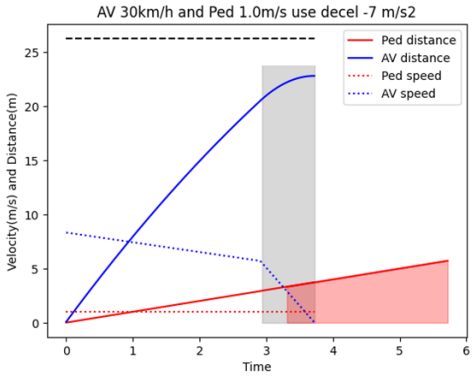
จะเห็นว่าจะมีช่วงที่มี Emergency brake และเกิดการชน (Collision) อยู่ในสถานการณ์ที่ III-3 การประเมิณระยะที่รถอัตโนมัติที่มีความเร็วต้นที่ 20 กม/ชม จะทำเหมือนกรณีความเร็วต้นของรถอัตโนมัติเป็น 25 และ 30 กม/ชม โดยระยะทางที่เหมาะสมที่ห่างจาก Conflict area คำนวณจากสถานการณ์ที่ III-3 ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 1.8 m/s จะทราบเวลาที่คนเดินเท้าเดินเข้ามาในเลนของรถอัตโนมัติ ($t_{ped,in\ road}$) จากสมการที่ 23

โดยที่ค่า $S_{ped,in\ road}$ ที่คนเดินเท้าจะเข้ามาในเลนของรถอัตโนมัติเป็น 2.92 เมตร ที่ความเร็วคนเดินเท้า (V_{ped}) เป็น 1.8 m/s จะได้ค่า $t_{ped,in\ road}$ เป็น 1.62 วินาที ซึ่งที่เวลาคนเดินเท้าเคลื่อนที่ได้ 1.62 วินาที ในสถานการณ์ที่ III-3 รถอัตโนมัติจะใช้ระยะเวลา 4.62 วินาที เนื่องจากเป็นสถานการณ์ที่รถออกจากจุดเริ่มต้นก่อนคนเดินเท้า 3 วินาที ในเวลาของรถอัตโนมัติดังกล่าว ระยะทางที่รถอัตโนมัติเคลื่อนที่ได้คือ 18.73 เมตร ในระยะดังกล่าวรถอัตโนมัติชะลอความเร็วจนเหลือ 9.19 กม/ชม ซึ่งน้อยกว่า 10 กม/ชม ที่เป็นเกณฑ์ที่กำหนดในการเปลี่ยนเลน ดังนั้นที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 20 กม/ชม ไม่ควรทำการเปลี่ยนเลน

จากข้อสรุปด้านบน รถอัตโนมัติจะเปลี่ยนเลนที่ความเร็วต้นของรถอัตโนมัติเป็น 25 และ 30 กม/ชม เท่านั้น เนื่องจากข้อกำหนดทางความเร็ว โดยจะจำลองเหตุการณ์การเปลี่ยนเลนเปรียบเทียบกับการใช้ Algorithm ที่นำเสนอ ที่มีการเบรกเพียงอย่างเดียว ในความเร็วต้นของรถอัตโนมัติทั้ง 2 ความเร็ว ดังต่อไปนี้

- ที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 30 กม/ชม แสดงเหตุการเปรียบเทียบกับการเบรกเพียงอย่างเดียว ในสถานการณ์ที่ I รถอัตโนมัติเปลี่ยนเลนเมื่อถึงระยะ 9.05 เมตร ที่คนเดินเท้าเดินด้วยความเร็ว 1 m/s แล้วยังไม่พบคนเดินเท้าที่เข้ามาในเลนดังรูปที่ 48b เปรียบเทียบกับการใช้ Algorithm ที่นำเสนอโดยไม่มีการเปลี่ยนเลน ที่ความเร็วคนเดินเท้าและความเร็วเริ่มต้นของรถอัตโนมัติ ความเร็วเดียวกัน ดังรูปที่ 48a เมื่อเปรียบเทียบกันพบว่าการเปลี่ยนเลนช่วยให้การขับขี่อยู่ในช่วง Normal driving zone ซึ่งผู้โดยสารจะรู้สึกปลอดภัยมากกว่าในกรณีที่เบรก เนื่องจากต้องเบรกด้วย Emergency braking และช่วยเพิ่มระยะการมองเห็นรถอัตโนมัติให้คนเดินเท้า ทำให้ไม่ให้เกิดการตกใจ

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



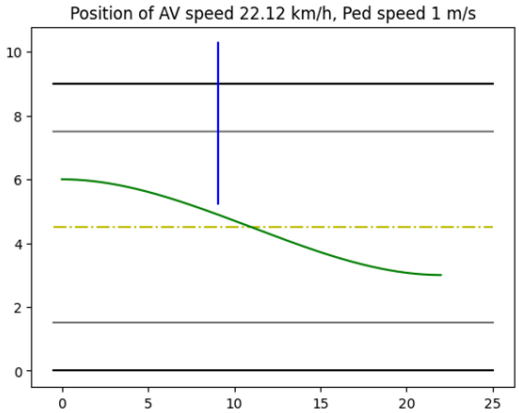
รูปที่ 48a ตัดสินใจเบรก

รูปที่ 48b ตัดสินใจเปลี่ยนเลน

รูปที่ 46 การเบรกและหักเลี้ยวเพื่อเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติ 25 km/h ในสถานการณ์ที่ I

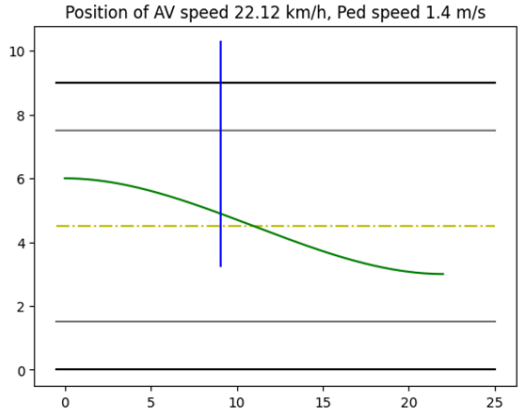
ในสถานการณ์ที่ III-1 รถอัตโนมัติเปลี่ยนเลนเมื่อถึงระยะ 9.05 เมตร แล้วยังไม่พบคนเดินเท้าที่เข้ามาในเลนดังรูปที่ 49 เปรียบเทียบกับการใช้ Algorithm ที่นำเสนอโดยไม่มีการเปลี่ยนเลน ที่ความเร็วเริ่มต้นของรถอัตโนมัติ ความเร็วเดียวกัน เมื่อเปรียบเทียบกันพบว่าการเปลี่ยนเลนช่วยให้การขับขี่อยู่ในช่วง Normal driving zone ซึ่งผู้โดยสารจะรู้สึกปลอดภัยมากกว่าในกรณีที่เบรก เนื่องจากต้องเบรกด้วย Emergency braking และช่วยเพิ่มระยะการมองเห็นรถอัตโนมัติให้คนเดินเท้า ทำให้ไม่ให้เกิดการตกใจ เฉพาะในกรณีที่ความเร็วคนเดินเท้าเป็น 1-1.2 m/s เท่านั้น ดังรูปที่ 49 และ 50 แต่ในกรณีที่ความเร็วคนเดินเท้าเป็น 1.4 m/s เกิดการชนเกิดขึ้น ซึ่งเป็นการชนในกรณีที่คนเดินเท้าเดินด้วยความเร็วคงที่และไม่หยุดเดิน จะพิจารณาเป็นกรณีที่เกิดขึ้นได้ยากมาก เนื่องจากคนเดินเท้าสามารถเห็นรถอัตโนมัติได้ในระยะที่ห่างที่เหมาะสมแล้ว ในความเป็นจริงการข้ามถนนคนเดินเท้าควรจะหยุดเดิน ดังรูปที่ 51 ซึ่งในความเร็วคนเดินเท้า 1.6-2 m/s ก็เกิดการชนด้วยเช่นกัน

Chulalongkorn University



รูปที่ 47 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1 m/s ในสถานการณ์ที่ III-1

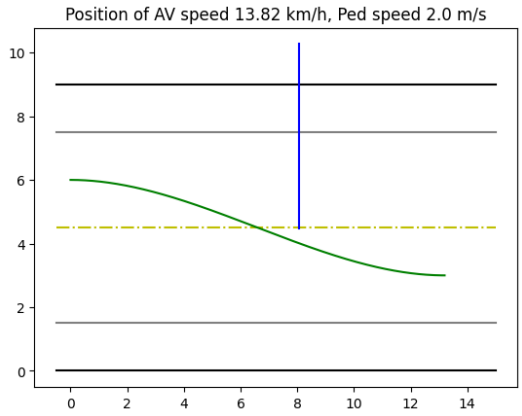
รูปที่ 48 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1.2 m/s ในสถานการณ์ที่ III-1



รูปที่ 49 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 1.4 m/s ในสถานการณ์ที่ III-1

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

- ที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 กม/ชม แสดงเหตุการเปรียบเทียบกับการเบรกเพียงอย่างเดียว ในสถานการณ์ที่ III-2 รถอัตโนมัติเปลี่ยนเลนเมื่อถึงระยะ 8.06 เมตร แล้วยังไม่พบคนเดินเท้าที่เข้ามาในเลนดังรูปที่ 52 เปรียบเทียบกับการใช้ Algorithm ที่นำเสนอโดยไม่มีการเปลี่ยนเลน ที่ความเร็วคนเดินเท้าและความเร็วเริ่มต้นของรถอัตโนมัติ ความเร็วเดียวกัน เมื่อเปรียบเทียบกันพบว่าการเปลี่ยนเลนช่วยให้การขับขี่อยู่ในช่วง Normal driving zone ซึ่งผู้โดยสารจะรู้สึกปลอดภัยมากกว่าในกรณีที่เบรก เนื่องจากต้องเบรกด้วย Emergency braking และช่วยเพิ่มระยะการมองเห็นรถอัตโนมัติให้คนเดินเท้า ทำให้ไม่ให้เกิดการตกใจ



รูปที่ 50 การเปลี่ยนเลนที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และความเร็วคนเดินเท้า 2 m/s ในสถานการณ์ที่ III-2

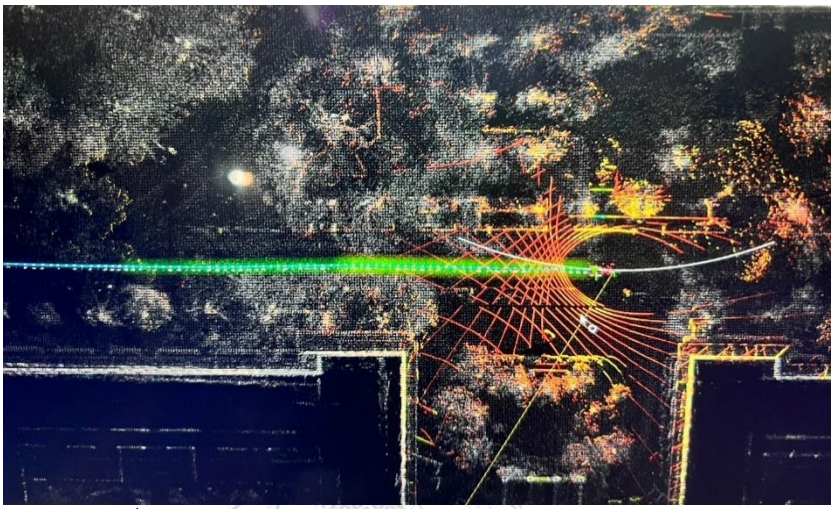
4.2 ผลการทดลองจากการทดลองจริง (Experiment)

การทำทดลองจริงจะทำการทดลองที่หน้าคณะวิศวกรรมศาสตร์ ของจุฬาลงกรณ์มหาวิทยาลัย ทำการทดลองหลังเวลา 22.00 น. เนื่องจากเป็นเวลาที่มหาวิทยาปิดทำการ และปิดประตูเข้าออกมหาวิทยาลัย โดยมีผู้ควบคุม (Staff) คอยควบคุมสถานการณ์เมื่อทำการทดลอง ในการทดลองใช้คนเดินเท้าเป็น Pedestrian Robot และวัตถุบดบัง (Obstacle object) ที่ทำจากท่อ PVC และใช้ฟิวเจอร์บอร์ดเป็นพื้นที่บดบัง ที่มีขนาดตามขนาดของรถยนต์ Mitsubishi Pajero ดังรูปที่ 53 โดยช่วงการชะลอความเร็วดังกล่าวจะทำการเซ็ตไปใน Waypoint ก่อนแล้ว ซึ่งวัตถุบดบัง (Obstacle object) ที่จำลองเป็นรถยนต์ทำการจำลองเป็นด้านท้ายของรถยนต์ และก่อนจำลองสถานการณ์นั้นๆ จะให้ Pedestrian robot ห่างวัตถุบดบังดังกล่าว 3 เมตร ก่อนเพื่อความปลอดภัยของอุปกรณ์

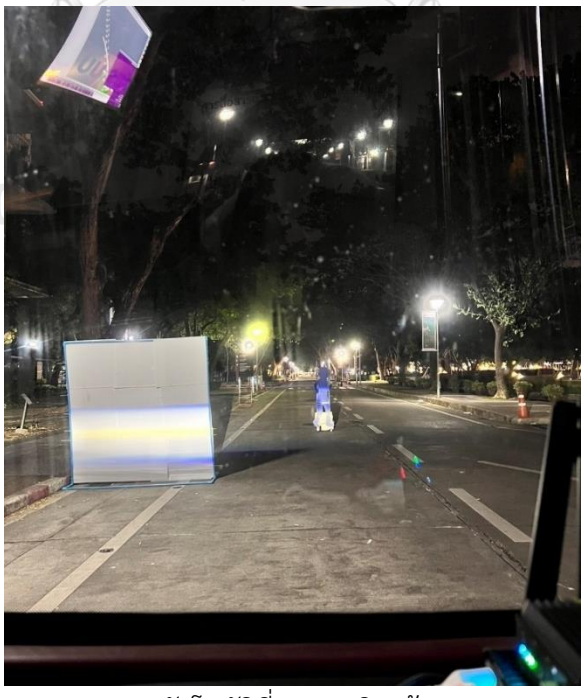
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

รูปที่ 51 จำลองเหตุการณ์ที่ใช้ทำการทดลอง

ก่อนการทดลองเพื่อวิ่งอัตโนมัติแต่ละครั้งจะต้องทำการระบุตำแหน่ง (Localization) ของรถอัตโนมัติ และตรวจสอนเส้นทาง (Waypoint) ให้แน่ในว่าอยู่ในเส้นทางที่ถูกต้อง ดังรูปที่ 54 และรูปที่ 55 เป็นภาพตัวอย่างที่ Safety driver บนรถอัตโนมัติมองเห็น Pedestrian robot จำเป็นต้องมี Safety driver เนื่องจากเมื่อเกิดเหตุการณ์ฉุกเฉินจะต้องให้ Safety driver คอยควบคุมสถานการณ์ให้ไม่ให้เกิดอันตราย

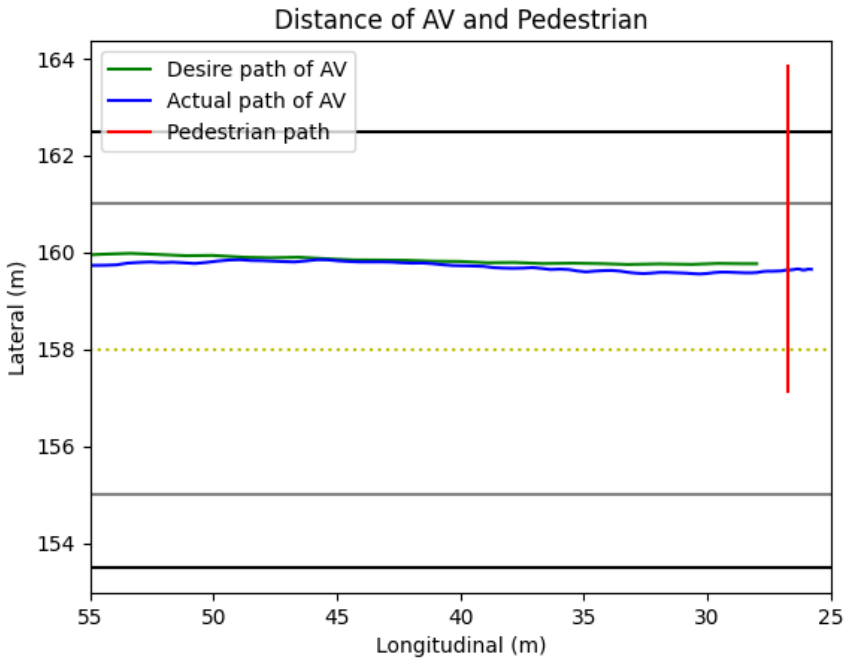


รูปที่ 52 การระบุตำแหน่งและเตรียมเส้นทางบนรถอัตโนมัติ



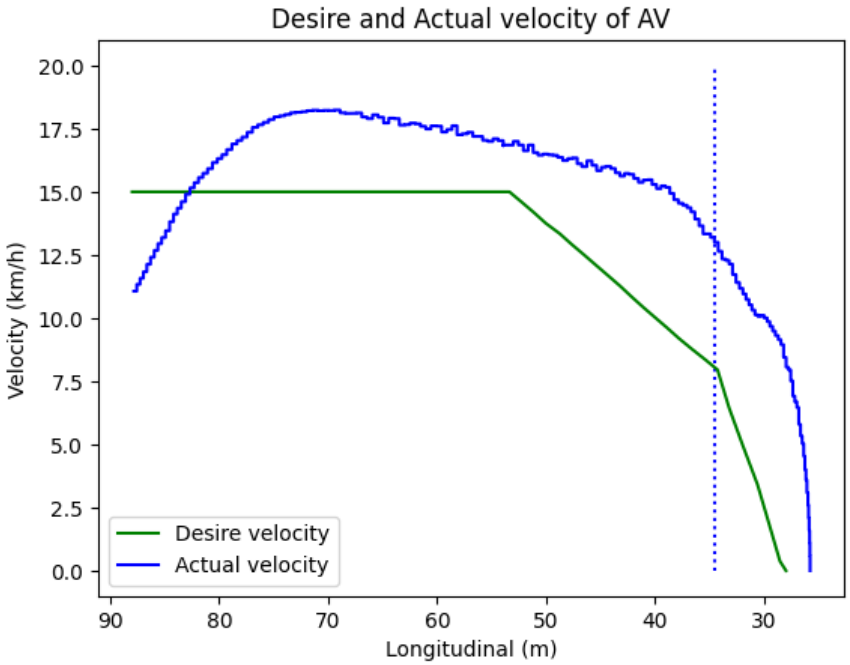
รูปที่ 53 มุมมองบนรถอัตโนมัติที่เจอคนเดินเท้า (Pedestrian robot)

การทดลองการเบรกในเลือกเหตุการณ์ในการจำลองเหตุการณ์ (Simulation) ในสถานการณ์ที่ III-3 ที่รถอัตโนมัติออกจากจุดเริ่มต้นก่อนคนเดินเท้าที่ออกจากจุดเริ่มต้น 3 วินาที ที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 15 km/h และคนเดินเท้าเดินด้วยความเร็ว 1.2 m/s จากผลการจำลองเหตุการณ์ (Simulation) พบว่ารถอัตโนมัติชะลอความเร็วเมื่อเจอวัตถุบดบัง (Obstacle object) ซึ่งถูกกำหนดลงไปในเส้นทางตั้งแต่แรกแล้ว (Waypoint) ด้วยอัตราหน่วง 0.35 m/s^2 หลังจากที่รถอัตโนมัติตรวจพบคนเดินเท้า จะเบรกด้วยอัตราหน่วง 1.28 m/s^2 สาเหตุที่เลือกเหตุการณ์นี้มาทำการทดลองพิจารณาด้านความปลอดภัยเป็นหลัก เนื่องจากเป็นเหตุการณ์จากการทำการจำลองเหตุการณ์ (Simulation) ทั้งหมดโดยใช้ Algorithm ที่นำเสนอ พบว่าเหตุการณ์นี้เป็นเหตุการณ์การตัดสินใจเบรกที่ใช้อัตราหน่วงน้อยที่สุด และมีความเร็วต้นของรถอัตโนมัติที่ใช้ไม่สูงมาก ดังกราฟที่ 56 แสดงเส้นทางที่กำหนดให้รถอัตโนมัติเคลื่อนที่ในเหตุการณ์ดังกล่าว (Desire path) กับเส้นทางที่รถอัตโนมัติเคลื่อนที่จริงตามเส้นทางที่กำหนด (Actual path) และเส้นทางของ Pedestrian robot โดยนำค่าจากการจำลองเหตุการณ์มาพลอตเพื่อแสดงว่าการทดลองนี้เกิดการชน (Collision) เกิดขึ้น ซึ่งการชนนี้ไม่ได้มาจาก Algorithm ที่นำเสนอ แต่มาจากการที่รถอัตโนมัติไม่สามารถเบรกได้ตามที่ Algorithm กำหนดเนื่องจาก Dynamic ของรถอัตโนมัติ



รูปที่ 54 การเคลื่อนที่ในแนว Longitudinal และ Lateral ของรถอัตโนมัติและคนเดินเท้าในสถานการณ์ที่ III-3

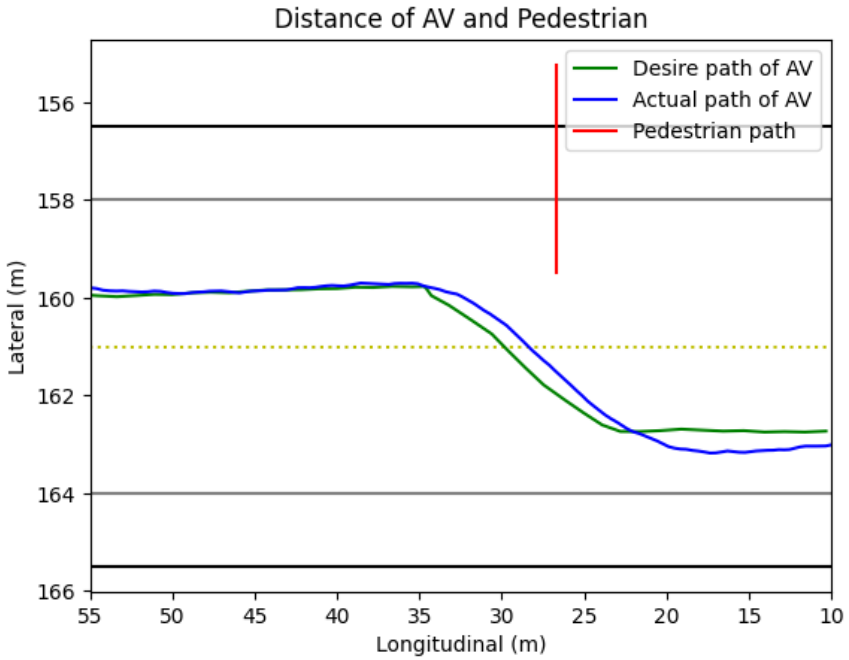
จากเหตุการณ์ดังกล่าวที่เกิดการชนที่มาจาก Low-level ของรถอัตโนมัติ กราฟรูปที่ 57 แสดงความเร็วที่กำหนด (Desire velocity) และความเร็วที่รถอัตโนมัติทำตามความเร็วที่กำหนด (Actual velocity) จะเห็นว่าก่อนเส้นประสีน้ำเงินคือช่วงที่รถอัตโนมัติเคลื่อนที่ตาม Waypoint โดยเส้นประสีน้ำเงินแสดงตำแหน่งของรถอัตโนมัติที่ตรวจเจอ Pedestrian robot ในการทดลองจริง จะเห็นว่าตำแหน่งที่รถอัตโนมัติตรวจเจอ Pedestrian robot ตรงกับใน Simulation และรถอัตโนมัติสั่งเบรกจริง ณ ขณะนั้น



รูปที่ 55 ความเร็วที่กำหนดในระบบการตัดสินใจที่ออกแบบ และความเร็วที่รถอัตโนมัติใช้วิ่งจริง

การทดลองการเปลี่ยนเลน เพื่อเพิ่มระยะการมองเห็นให้รถอัตโนมัติ เลือกเหตุการณ์ในการจำลองเหตุการณ์ (Simulation) ในสถานการณ์ที่ III-3 ที่รถอัตโนมัติออกจากจุดเริ่มต้นก่อนคนเดินเท้าที่ออกจากจุดเริ่มต้น 3 วินาที ที่ความเร็วเริ่มต้นของรถอัตโนมัติเป็น 25 km/h และคนเดินเท้าเดินด้วยความเร็ว 1 m/s จากผลการจำลองเหตุการณ์ (Simulation) พบว่ารถอัตโนมัติชะลอความเร็วเมื่อเจอวัตถุบดบัง (Obstacle object) ซึ่งจากการทดลองเหตุการณ์การเบรกในรูปที่ 41 และรูปที่ 42 ที่ทำให้ทราบว่ารถอัตโนมัติมี Velocity error อยู่มาก ดังนั้นการชะลอความเร็วแล้วหักเลี้ยวเพื่อเปลี่ยนเลนจึงเป็นเหตุการณ์ที่น่ากังวล ดังนั้นการทำการทดลองจึงนำค่า Simulation ของเหตุการณ์นี้ที่กล่าวว่าขณะรถอัตโนมัติชะลอความเร็วจาก 25 km/h ด้วยอัตราหน่วง 0.9 m/s^2 จนถึงระยะ 6.06 เมตร จากด้านท้ายของวัตถุบดบัง (Obstacle object) แล้วยังไม่พบคนเดินเท้าจะหักเลี้ยวเพื่อเปลี่ยน

เลน ซึ่งระยะดังกล่าว รถอัตโนมัติถูกชะลอความเร็วจนเหลือ 13.82 km/h ดังนั้นการทดลองจริงจึงทำการทดลองเป็นการที่รถอัตโนมัติวิ่งด้วยความเร็วคงที่ 13.82 km/h เมื่อถึงระยะ 6.06 m จากวัตถุบดบัง (Obstacle object) จะสั่งให้รถอัตโนมัติหักเลี้ยวเพื่อเปลี่ยนเลนตามที่ Algorithm นำเสนอ ดังกราฟที่ 58 แสดงเส้นทางที่กำหนดให้รถอัตโนมัติเคลื่อนที่ในเหตุการณ์ดังกล่าว (Desire path) กับเส้นทางที่รถอัตโนมัติเคลื่อนที่จริงตามเส้นทางที่กำหนด (Actual path) และเส้นทางของ Pedestrian robot โดยนำค่าจากการจำลองเหตุการณ์มาพลอตเพื่อแสดงว่าการทดลองนี้ไม่เกิดการชน (Collision) เกิดขึ้น และเป็นไปตามที่ Algorithm กำหนด



รูปที่ 56 การหักเลี้ยวเพื่อเปลี่ยนเลนที่ความเร็วต้นของรถอัตโนมัติเป็น 25 km/h และคนเดินเท้าเป็น 1 m/s ในสถานการณ์ที่ III-3

จากการทดลอง (Experiment) เพื่อยืนยังผลการจำลองเหตุการณ์ทั้งหมด (Simulation) พบว่า Algorithm ที่งานวิจัยนี้นำเสนอ รถอัตโนมัติสามารถสั่งการการตัดสินใจที่ออกแบบได้จริง แต่ด้วยเรื่องของ Low level ทั้ง Velocity error ทั้ง Lateral error การระบุตำแหน่งรถอัตโนมัติ (Localization) รวมไปถึงการที่ไม่ได้นำ Dynamic model มาใช้เพื่อจำลองเหตุการณ์ ทำให้เกิดข้อผิดพลาดทาง Dynamic ของรถอัตโนมัติดังที่เห็น แต่ทั้งนี้ การทำงานของ Algorithm ที่ออกแบบเป็นไปตามที่กำหนด

บทที่ 5 สรุปผล และข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้ทำการออกแบบระบบการตัดสินใจ (Decision-making) ในสถานการณ์การข้ามถนนของคนเดินเท้าที่มาวัตถุบดบังอยู่ในไหล่ทาง (Obstacle object) ด้วยวิธีการเปรียบเทียบจากค่า TTC กับ TTB ซึ่งมีการเผื่อ Time delay ลงไปในระบบการตัดสินใจด้วย และวิธีการหาเส้นทางการเปลี่ยนเลนที่เหมาะสมเมื่อการเบรกไม่เพียงพอ โดยการเบรกคำนึงถึงยานพาหนะที่ตามหลังรถอัตโนมัติ ให้สามารถเบรกหรือชะลอความเร็วได้ตาม Time Headway ที่ 2 วินาที และการเปลี่ยนเลนคำนึงถึงยานพาหนะคันหลังในเลนที่รถอัตโนมัติกำลังจะเปลี่ยนไปเลนนั้น โดยทำการเปลี่ยนเลนเมื่อมีค่า $TTC_{AV,HV2}$ มากกว่า 3 วินาที การเปลี่ยนเลนทำเพื่อให้คนเดินเท้าสามารถมองเห็นรถอัตโนมัติได้ในระยะที่ไกลขึ้น และลดความเสี่ยงที่จะเดินออกมา โดยกำหนดให้ยานพาหนะที่ขับตามรถอัตโนมัติทั้งสองเลน HV1 และ HV2 และรถอัตโนมัติ ขับด้วยความเร็วไม่เกิน 30 km/h เนื่องจากเป็นการขับขี่ในมหาวิทยาลัย จึงมีการกำหนดความเร็วสูงสุดที่ควรขับขี่

งานวิจัยนี้ใช้เกณฑ์การขับขี่ของรถ Shuttle Bus คือรถรูปแบบ Bus ที่ใช้รับส่งผู้โดยสารที่ระบุจุดเริ่มต้นและจุดหมายปลายทางไว้ โดยไม่ได้จอดรับ-ส่งผู้โดยสารระหว่างทาง เกณฑ์การขับขี่ดังกล่าวจะนำมาใช้ทั้งการตัดสินใจเบรก โดยการเลือกค่าอัตราหน่วงเป็นค่าอัตราหน่วงมากที่สุดในช่วง Emergency braking และเลือกค่าอัตราหน่วงในแนว Lateral กับแนว Longitudinal และการเปลี่ยนแปลงค่าอัตราหน่วงในแนว Lateral กับแนว Longitudinal ให้ไม่เกินช่วง Aggressive zone เปลี่ยนเลน เนื่องจากเราคำนึงถึงผู้โดยสารทั้งหมด (Passenger) ที่อยู่บนรถอัตโนมัติ ให้มีความปลอดภัยและตื่นตระหนกน้อยที่สุดที่เป็นไปได้

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ในแนวคิดการออกแบบระบบการตัดสิน (Decision-making) ที่งานวิจัยนี้นำเสนอ สามารถในไปใช้ได้ 2 กรณี กรณีแรกเป็นกรณีที่รถอัตโนมัติวิ่งบนถนนสองเลนที่เป็นเลนสวน เป็นกรณีที่รถอัตโนมัติไม่สามารถเปลี่ยนเลนได้ จากการจำลองเหตุการณ์ (Simulation) ใน 3 สถานการณ์คือ 1. สถานการณ์ที่คนเดินเท้าออกจากจุดเริ่มต้นพร้อมกับที่รถอัตโนมัติออกจากจุดเริ่มต้น 2.สถานการณ์ที่คนเดินเท้าออกจากจุดเริ่มต้น 1 2 และ 3 วินาที ก่อนที่รถอัตโนมัติจะออกจากจุดเริ่มต้น และ 3. สถานการณ์ที่รถอัตโนมัติออกจากจุดเริ่มต้น 1 2 และ 3 วินาที ก่อนที่คนเดินเท้าจะออกจากจุดเริ่มต้น พบว่า Algorithm ของงานวิจัยนี้ที่มีการเบรกเพียงอย่างเดียว สามารถลดการการชน (Collision) ในกรณีที่รถอัตโนมัติชนคนเดินเท้าได้ 100% แต่ในการชนกรณีที่คนเดินเท้าชนรถอัตโนมัติอาจจะยังเกิดอยู่ สาเหตุมาจากรถอัตโนมัติเข้า Conflict area ก่อนที่จะตรวจเจอคนเดินเท้า ทำให้รถอัตโนมัติไม่ได้ตัดสินใจที่จะเบรกเลย และคนเดินเท้าอาจเจอรถอัตโนมัติในระยะที่ใกล้มาก จึงเป็นไปได้ว่าจะยังเกิด

การชนในรูปแบบนี้อยู่ แต่โอกาสที่จะเกิดขึ้นในความเป็นจริงขึ้นอยู่กับความระมัดระวังของตัวบุคคลที่จะข้ามถนนด้วย และในกรณีที่สองคือรถอัตโนมัติวิ่งบนถนนเลนคู่ ที่สามารถเปลี่ยนเลนได้ จากการจำลองเหตุการณ์ (Simulation) ทั้ง 3 สถานการณ์ดังกล่าว พบว่าการเปลี่ยนเลนช่วยเพิ่มระยะห่างการมองเห็นแก่คนเดินเท้าที่กำลังจะข้ามถนน ให้คนเดินเท้าสามารถมองเห็นและหยุดเดินได้ทัน เพื่อลดมุมอับ (Blind spot) ให้แก่คนเดินเท้าด้วย โดยการเปลี่ยนเลน เปลี่ยนด้วยเส้นทาง ความเร็ว อัตราเร่งและอัตราหน่วงในเกณฑ์การขับขี่ที่นำเสนอให้เหมาะสม ทั้งการวิ่งของรถอัตโนมัติในถนนเลนคู่หรือในถนนเลนสวน

หลังจากการทำการจำลองเหตุการณ์ (Simulation) จะนำเหตุการณ์ที่น่าสนใจมาทำการทดลองบนรถอัตโนมัติในสถานการณ์จริง (Experiment) ในพื้นที่ปิดที่จุฬาลงกรณ์มหาวิทยาลัย ในเวลาหลัง 22.00 น. คือเวลาที่มหาวิทยาปิดทำการ โดยมีผู้ช่วยในการทำการทดลองเพื่อดูความปลอดภัยบนถนน ซึ่งเหตุการณ์ที่สนใจคือเหตุการณ์ที่มีความเสี่ยงน้อยที่สุดในการเบรกมาทดลอง เนื่องจากปัจจัยด้านความปลอดภัยในการทดลองจริง คือสถานการณ์ที่ III-3 เหตุการณ์ที่ความเร็วต้นของรถอัตโนมัติเป็น 15 km/h และคนเดินเท้าเดินด้วยความเร็ว 1.2 m/s เนื่องจากในช่วงที่รถอัตโนมัติตัดสินใจเบรก รถอัตโนมัติเบรกด้วยอัตราหน่วง -1.28 m/s^2 ซึ่งเป็นอัตราหน่วงที่น้อยที่สุดในทั้ง 3 สถานการณ์ที่รถอัตโนมัติตัดสินใจจะเบรก และสถานการณ์การเปลี่ยนเลนที่ความเร็วต้นของรถอัตโนมัติเป็น 25 km/h และคนเดินเท้าเป็น 1 m/s ในสถานการณ์ที่ III-1 ที่เปลี่ยนเลนด้วยความเร็วรถอัตโนมัติเป็น 13.82 km/h เนื่องจากเป็นเหตุการณ์การเปลี่ยนเลนด้วยความเร็วที่ต่ำที่สุด จึงมีความปลอดภัยสูงสูงในกรณีการเปลี่ยนเลนที่นำมาทำการทดลอง จากทั้งสองเหตุการณ์ที่นำมาทดลองพบว่ารถอัตโนมัติมี Lateral error อยู่ประมาณ 20 cm

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

5.2 ข้อเสนอแนะ

การออกแบบระบบการตัดสินใจ (Decision-making) ที่ออกแบบควรมี Time delay น้อยที่สุดเท่าที่เป็นไปได้ เพื่อให้ทันต่อการตัดสินใจ และเมื่อออกแบบเส้นทางการเบรกตามที่ตัดสินใจ รถอัตโนมัติควรต้องแก้ไข Lateral error เพื่อให้การเบรกเป็นไปตามที่ออกแบบ และแก้ไข Velocity error เพื่อให้การเบรกเป็นไปตามที่ออกแบบเช่นเดียวกัน

ในงานวิจัยนี้ไม่ได้นำ Dynamic model ของรถอัตโนมัติมาพิจารณาเนื่องจากมีพารามิเตอร์ที่ไม่ทราบค่าเยอะมาก ด้วยระยะเวลาในการทำงานวิจัย และต้นทุนของการทำงานวิจัยทำให้ไม่สามารถหาพารามิเตอร์ต่างๆ เพื่อนำมาทำเป็น Dynamic model ได้ แต่หากทราบ Dynamic model ของรถอัตโนมัติคันที่ใช้ทำการทดลอง การออกแบบเส้นทางและการทดลองจะมีความแม่นยำมากยิ่งขึ้น

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

1. Computer ที่ใช้ทั้งหมดแสดงรายละเอียดดังตารางที่ 1

รูปที่ 57 คอมพิวเตอร์เครื่องที่ 1

รูปที่ 58 คอมพิวเตอร์เครื่องที่ 2

ตารางที่ 13 คุณสมบัติคอมพิวเตอร์ที่ใช้

คอมพิวเตอร์	เครื่องที่ 1	เครื่องที่ 1
หน้าที่	จำลองเหตุการณ์	เตรียม Software ใช้ทดลอง
Manufacturer	Apple	ประกอบ
CPU	Intel Core i5 Quad-Core / 2GHz	Intel core i9 10900K / 3.70GHz
Ram	16 GB 3733 MHz	32 GB
Display Resolution	2560 x 1600	-
HDD	512 GB	-
Operation System	macOS Ventura	Ubuntu

2. Software ที่ใช้จำลองเหตุการณ์

กำหนด Parameter ใน Paremeter.py

```
import math

def parameter_ped_av():
    r_sensor = 20
    AV_width = 1.61
    AV_length = 4.3
    Ped_width = 0.5
    l_ped_long = 2
    buffer = 0.3
    CF_width = AV_width + buffer
    CF_length = Ped_width + buffer
    return r_sensor, AV_width, AV_length, Ped_width, l_ped_long, CF_width,
CF_length

def parameter_other():
    OB_width = 1.82
    OB_length = 4.7
    HV_width = 1.82
    HV_length = 4.7
    road_width = 3
    shoulder_width = 1.5
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
    return OB_width, OB_length, HV_width, HV_length, road_width, shoulder_width

r_sensor, AV_width, AV_length, Ped_width, l_ped_long, CF_width, CF_length =
parameter_ped_av()
OB_width, OB_length, HV_width, HV_length, road_width, shoulder_width =
parameter_other()

iav_C = math.sqrt(r_sensor**2 - ((road_width + shoulder_width - OB_width)/2)**2) +
OB_length + l_ped_long
beta = math.asin((road_width/2 + shoulder_width - (shoulder_width - OB_width)/2) /
20)
iped_C = math.tan(beta) * iav_C

print(f'iav_C = {round(iav_C,4)} m')
print(f'iped_C = {round(iped_C,4)} m')
```

นำ Parameter.py มาใช้เพื่อ Simulation แต่ละสถานการณ์ว่าเกิดการชนหรือไม่

```
import matplotlib.pyplot as plt
import numpy as np
import parameter

time = 0.01
a_brake_max = -5.08
a_full_brake = -7
iav_C, iped_C = parameter.iav_C, parameter.iped_C
r_sensor, AV_width, AV_length, Ped_width, l_ped_long, CF_width, CF_length =
parameter.parameter_ped_av()
OB_width, OB_length, HV_width, HV_length, road_width, shoulder_width =
parameter.parameter_other()
ipedF = iped_C - road_width/2 - (shoulder_width - OB_width)/2

class Object:
    def __init__(self, start, speed):
        self.position = start
        self.velocity = speed

    def slowdown(self, time, a_slowdown):
        self.velocity += (a_slowdown*time)
        self.position += self.velocity*time - 0.5*a_slowdown*(time**2)

    def constant(self, time):
        self.position += self.velocity*time

    def fullbrake(self, time):
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            self.velocity += a_full_brake*time
            self.position += self.velocity*time

        def brakemax(self, time):
            self.velocity += a_brake_max*time
            self.position += self.velocity*time

        def brake(self, time, deceleration):
            self.velocity += deceleration*time
            self.position += self.velocity*time

        def steering(self, time, steer):
            self.velocity += 0


    def decision_making(TTB, TTC, a):
        if TTB < TTC:
            if abs(a) < 5.08:
                b = a
                print('Brake')
                action = 'Brake'
                AV_2.brake(time, b)
                collision_status.append('Not')
                action_av.append('Brake')
            else:
                b = -5.08
                print('Brakemax')
                action = 'Brakemax'
                AV_2.brakemax(time)

        else:
            b = a_full_brake
            action = 'Fullbrake'
            AV_2.fullbrake(time)
            Tb4_in = (iped_C - CF_length/2 - Ped.position) / Ped.velocity
            Tb3_out = (iped_C + CF_length/2 - Ped.position + Ped_width) / Ped.velocity
            Ta1_in = (iav_C - CF_width/2 - AV.position) / AV.velocity
            Ta2_out = (iav_C + CF_width/2 - AV.position + AV_length) / AV.velocity
            TB4_in.append(Tb4_in)
            TB3_out.append(Tb3_out)
            TA1_in.append(Ta1_in)
            TA2_out.append(Ta2_out)
            print('Fullbrake')
            action_av.append('Fullbrake')
                #steering
        return b, action
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
def select_a_slowdown(u):
    a_select = -u**2 / (2*iav_C)
    if abs(a_select) <= 0.9:
        a_slowdown = a_select
    else:
        a_slowdown = -0.9
    return a_slowdown

def append_to_plot(sec, av_position, ped_position, av_velocity, ped_velocity):
    Time.append(sec)
    AV_dist.append(av_position)
    AV_speed.append(av_velocity)
    Ped_dist.append(ped_position)
    Ped_speed.append(ped_velocity)

scenario = float(input('You see scenario '))
for v_av in range(10, 31, 5):
    a_slow = select_a_slowdown(u=v_av/3.6)
    #with pd.ExcelWriter( f'./CSV
using/Brake/III/1/Result_Brake_Vav{str(v_av)}.xlsx') as writer:
    for v_ped in range(10, 21, 2):
        v_hv = 60
        if scenario == 1 or scenario == 2.1 or scenario == 2.2 or scenario == 2.3:
            t = 0
            initial_pos_av = 0
            initial_pos_hv = 0 * v_hv/3.6
            if scenario == 1:
                initial_pos_ped = 0
            elif scenario == 2.1:
                initial_pos_ped = 1 * v_ped/10
            elif scenario == 2.2:
                initial_pos_ped = 3 * v_ped/10
            else:
                initial_pos_ped = 3 * v_ped/10
        else:
            if scenario == 3.1:
                t = 1
            elif scenario == 3.2:
                t = 2
            elif scenario == 3.3:
                t = 3
            else:
                print('You put fault value')
            initial_pos_av = (v_av/3.6)*t + 0.5*a_slow*(t**2)
            initial_pos_ped = 0
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            initial_pos_hv = 0 * v_hv/3.6
        AV = Object(initial_pos_av , v_av/3.6)
        HV = Object(initial_pos_hv, v_hv/3.6)
        Ped = Object(initial_pos_ped, v_ped/10)

        i = 0
        Ped_dist, Ped_speed, iped_F, HV_dist, HV_speed, hv_dec = [], [], [], [],
[], []
        AV_dist, AV_speed, AV_dec_cal, AV_dec_use = [], [], [], []
        TA1_in, TA2_out, TB4_in, TB3_out = [], [], [], []
        detect_status, collision_status, action_av, case, Time = [], [], [], [], []

        while i <= 20 and AV.position <= 34.2551 and Ped.position <= 5.7201 and
AV.velocity > 0.05 :
            iped_F.append(ipedF)
            Ped.constant(time)
            if Ped.position < ipedF :
                AV.slowdown(time, a_slow)
                append_to_plot(i, AV.position, Ped.position, AV.velocity,
Ped.velocity)
                AV_dec_cal.append(0)
                AV_dec_use.append(a_slow)
                Tb4_in = (iped_C - CF_length/2 - Ped.position) / Ped.velocity
                Tb3_out = (iped_C + CF_length/2 - Ped.position + Ped_width) /
Ped.velocity
                Ta1_in = (iav_C - CF_width/2 - AV.position) / AV.velocity
                Ta2_out = (iav_C + CF_width/2 - AV.position + AV_length) /
AV.velocity
                TB4_in.append(Tb4_in)
                TB3_out.append(Tb3_out)
                TA1_in.append(Ta1_in)
                TA2_out.append(Ta2_out)
            else:
                break
            i += time

        brake = []
        j = i
        TTB = (-AV.velocity) / (a_full_brake) + 0.2
        initial_pos_ped_2 = Ped.position
        initial_pos_av_2 = AV.position
        v_av_last = AV.velocity
        Ped_2 = Object(initial_pos_ped_2, v_ped/10)
        AV_2 = Object(initial_pos_av_2, v_av_last)
        a = -(AV_2.velocity**2) / (2*(iav_C - 2 - AV_2.position))
        if Tb3_out > Ta1_in > Tb4_in:
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            to_print_caseI = 'True'
            to_print_caseII = 'False'
            Collision = 'Occur'
            while j < 20 and AV_2.position <= 34.3551 and Ped_2.position <= 5.7201
and AV_2.velocity > 0.05 :
                Ped_2.constant(time)
                detect_status.append('Detected')
                # a = -(AV_2.velocity**2) / (2*(iav_C - 2 - AV_2.position))
                brake.append(a)
                TTC = Ta1_in
                case.append('I')
                b, action = decision_making(TTB=TTB, TTC=TTC, a=a)
                append_to_plot(j, AV_2.position, Ped_2.position, AV_2.velocity,
Ped_2.velocity)
                j += time

        elif Ta2_out > Tb4_in > Ta1_in:
            to_print_caseI = 'False'
            to_print_caseII = 'True'
            Collision = 'Occur'
            while j < 20 and AV_2.position <= 34.3551 and Ped_2.position <= 5.7201
and AV_2.velocity > 0.05 :
                Ped_2.constant(time)
                detect_status.append('Detected')
                # a = -(AV_2.velocity**2) / (2*(iav_C - 2 - AV_2.position))
                brake.append(a)
                TTC = Tb4_in
                case.append('II')
                b, action = decision_making(TTB=TTB, TTC=TTC, a=a)
                append_to_plot(j, AV_2.position, Ped_2.position, AV_2.velocity,
Ped_2.velocity)
                j += time
        else:
            to_print_caseI = 'Fasle'
            to_print_caseII = 'False'
            Collision = 'Not'
            while j < 20 and AV_2.position <= 34.3551 and Ped_2.position <= 5.7201
and AV_2.velocity > 0.05 :
                Ped_2.constant(time)
                action = 'Constant speed'
                detect_status.append('Detected')
                b = 0
                TTC = 100
                AV_2.constant(time)
                # print('Constant')
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
                append_to_plot(j, AV_2.position, Ped_2.position, AV_2.velocity,
Ped_2.velocity)
                collision_status.append('Not')
                action_av.append('Constant')
                case.append('-')
                j += time

        plt.plot(Time, Ped_dist, 'r-', label='Ped distance')
        plt.plot(Time, AV_dist, 'b-', label='AV distance')
        x = np.linspace(0, j, 100)
        y1 = np.linspace(max(AV_dist)+1, max(AV_dist)+1, 100)
        plt.plot(x, y1, color='white')
        plt.fill_between(x, y1, where=(x >= i), color='grey', alpha=0.3)
        if scenario == 2.1:
            x_ped = np.linspace((3.3101)/(v_ped/10)-1, (5.7201)/(v_ped/10)-1, 100)
            y_ped = np.linspace(3.3101, 5.7201, 100)
        elif scenario == 2.2:
            x_ped = np.linspace((3.3101)/(v_ped/10)-2, (5.7201)/(v_ped/10)-2, 100)
            y_ped = np.linspace(3.3101, 5.7201, 100)
        elif scenario == 2.3:
            x_ped = np.linspace((3.3101)/(v_ped/10)-3, (5.7201)/(v_ped/10)-3, 100)
            y_ped = np.linspace(3.3101, 5.7201, 100)
        elif scenario == 1:
            x_ped = np.linspace((3.3101)/(v_ped/10), (5.7201)/(v_ped/10), 100)
            y_ped = np.linspace(3.3101, 5.7201, 100)
        else:
            if j >= 3.3101/(v_ped/10):
                x_ped = np.linspace((3.3101/(v_ped/10)), j, 100)
                y_ped = np.linspace(3.3101, j*(v_ped/10), 100)
            else:
                x_ped = np.linspace(3.3101/(v_ped/10), 3.3101/(v_ped/10), 100)
                y_ped = np.linspace(3.3101, 3.3101, 100)
        plt.plot(x_ped, y_ped, color='red')
        plt.fill_between(x_ped, y_ped, color='red', alpha=0.3)
        plt.plot(Time, Ped_speed, 'r:', label='Ped speed')
        plt.plot(Time, AV_speed, 'b:', label='AV speed')
        x = np.linspace(0, j, 100)
        y1 = np.linspace(26.2551, 26.2551, 100)
        y2 = np.linspace(34.3551, 34.3551, 100)
        if AV_2.position >= 20:
            plt.plot(x, y1, color='black', linestyle='--')
            if AV_2.position >= 26.2551:
                plt.plot(x, y2, color='black', linestyle='--')
            else:
                pass
        else:
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
            pass
        # plt.fill_between(x, y1, where=(y1>26.221), color='blue', alpha=0.3)
        plt.legend()
        plt.title(f'AV {v_av}km/h and Ped {v_ped/10}m/s use decel {round(b,2)}
m/s2')
        plt.ylabel('Velocity(m/s) and Distance(m)')
        plt.xlabel('Time')
        plt.show()

        print('1. When pedestrian had been detected, condition of collision was
checked.')
        print(f'   Condition caseI : {round(Tb3_out,2)} > {round(Ta1_in,2)} >
{round(Tb4_in,2)} is {to_print_caseI}')
        print(f'   Condition caseII : {round(Ta2_out,2)} > {round(Tb4_in,2)} >
{round(Ta1_in,2)} is {to_print_caseII}')
        print(f'   *** So {Collision}')
        if Collision == 'Occur':
            print('2. Decision-making is working.')
            print(f'   TTB = {round(TTB,4)} (Time delay = 0.2 s)')
            print(f'   TTC = {round(TTC,4)}')
            print(f'   Calculate deceleration = {round(a,2)}')
            print(f'   *** So action {action}')
            print('        ------------------------------')
            print(f'Deceleration use brake = {round(b, 4)} m/s2')
        print(f'Ped was detected at AV position = {round(initial_pos_av_2, 4)} m at
{round(i,4)} sec at velocity {round(v_av_last,4)} m/s.')
        print(f'The end loop AV.position = {round(AV_2.position, 4)} m and
AV.velocity = {round(AV_2.velocity, 4)} m/s')
        print(f'The end loop Ped.position = {round(Ped_2.position, 4)} m')
        print('   ')
        print(f'Deceleration slowdown is {a_slow,2} m/s2')
        print('   ')
        print('   ')
        print('------------------------------------------------------------------
--')
```

รูปที่ 59 Code Python ที่ใช้จำลองเหตุการณ์ของ Algorithm

คำนวณความสัมพันธ์ระหว่างยานพาหนะคันหลัง HV1 และรถอัตโนมัติ

```
import math
import matplotlib.pyplot as plt

a_full_brake = -1

class Object:
```

```
    def __init__(self, start, speed):
        self.position = start
        self.velocity = speed

    def slowdown(self, time, a_slowdown):
        self.velocity += (a_slowdown*time)
        self.position += self.velocity*time - 0.5*a_slowdown*(time**2)

    def constant(self, time):
        self.position += self.velocity*time

    def fullbrake(self, time):
        self.velocity += a_full_brake*time
        self.position += self.velocity*time

    def brake(self, time, deceleration):
        self.velocity += deceleration*time
        self.position += self.velocity*time

i = 0
time = 0.01
time_max = 100

HV = Object(3, 30/3.6)
AV = Object(24.37, 10/3.6)

HV.constant(time)

Time, TTC, decel = [], [], []

while i <= time_max:
    AV.slowdown(time, -0.12)
    a = ((AV.velocity**2) - (HV.velocity**2)) / (2*21.37)
    HV.brake(time, a)
    ttc = (abs(AV.position - HV.position - 4.7)) / (abs(AV.velocity - HV.velocity))
    Time.append(i)
    TTC.append(ttc)
    decel.append(a)
    if AV.velocity <= 0:
        break

    i += time

plt.plot(Time, decel, 'g-')
plt.show()
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
plt.plot(Time, TTC, 'g-')
plt.show()
```

รูปที่ 60 Code Python ที่ใช้จำลองเหตุการณ์ระหว่างรถอัตโนมัติและยานพาหนะคันหลัง

2. Software ที่ใช้เพื่อทำการทดลองจริง

ใช้ TCP เพื่อส่งข้อมูลเมื่อ Pedestrian robot เข้ามาในถนนเลนที่รถอัตโนมัติกำลังจะวิ่งไป

```
#!/usr/bin/env python
import rospy
import socket
from std_msgs.msg import String

def tcp_server():
    # Initialize the ROS node
    rospy.init_node('tcp_server')

    # Define the server's IP address and port
    SERVER_IP = '127.0.0.1'  # Replace with the actual server IP
    SERVER_PORT = 1234  # Replace with the actual server port

    # Create a TCP socket
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    # Bind the socket to the server address and port
    server_socket.bind((SERVER_IP, SERVER_PORT))

    # Listen for incoming connections
    server_socket.listen(1)

    rospy.loginfo("TCP server is listening for incoming connections...")

    # Create a publisher for the TCP data
    pub = rospy.Publisher('tcp_out', String, queue_size=10)

    while not rospy.is_shutdown():
        # Accept a client connection
        client_socket, client_address = server_socket.accept()
        rospy.loginfo("Accepted connection from: %s", client_address)

        # Receive data from the client
        data = client_socket.recv(1024)
        rospy.loginfo("Received from client: %s", data.decode())
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        # Publish the TCP data to the 'tcp_out' topic
        pub.publish(data.decode())

        # Close the client socket
        client_socket.close()

    # Close the server socket
    server_socket.close()

if __name__ == '__main__':
    try:
        tcp_server()
    except rospy.ROSInterruptException:
        pass
```

รับค่าจาก TCP เพื่อตัดสินใจตามที่ Algorithm ออกแบบ

```
import rospy
import math
from autoware_msgs.msg import LaneArray, Lane, Waypoint
from geometry_msgs.msg import PoseStamped, TwistStamped, Point, Twist
from nav_msgs.msg import Odometry
from std_msgs.msg import String


class smrc_pull_over(object):
    def __init__(self):
        rospy.init_node('smrc_pull_over')
        rospy.Subscriber('/tcp_server', String, self.tcp_callback)

        rospy.Subscriber('/current_pose', PoseStamped, self.pose_callback)
        rospy.Subscriber('/current_velocity', TwistStamped, self.vel_callback)
        rospy.Subscriber('/original_waypoints', LaneArray, self.lane_callback)
        self.cur_pose = PoseStamped()
        self.lane_raw = LaneArray()

        self.waypoint_publisher = rospy.Publisher('/final_waypoints', LaneArray,
queue_size=10)
        self.current_speed = None
        self.deceleration = 0
        self.cur_idx = None

    def lane_callback(self, msg):
        self.lane_raw = msg

    def pose_callback(self, msg):
```

```
            if msg:
                self.cur_pose = msg
                dist = []
                for lane in self.lane_raw.lanes:
                    for wp in lane.waypoints:
                        dist.append(math.hypot(self.cur_pose.pose.position.x -
    wp.pose.pose.position.x, self.cur_pose.pose.position.y - wp.pose.pose.position.y))
                if dist:
                    self.cur_idx = dist.index(min(dist))

        def vel_callback(self,msg):
            self.current_speed = msg.twist.linear.x

        def brake_action(self, decel):
            new_lane = Lane()
            for lane in self.lane_raw.lanes:
                new_lane.waypoints = lane.waypoints[0:self.cur_idx + 1]
                new_lane.header = lane.header
                for i in range(self.cur_idx + 1, len(lane.waypoints)):
                    # a u s
                    new_wp = Waypoint()
                    dx = lane.waypoints[i-1].x - lane.waypoints[i].x
                    dy = lane.waypoints[i-1].y - lane.waypoints[i].y
                    dist = math.hypot(dx, dy)
                    v = max(math.sqrt((self.current_speed**2)+(2*decel*dist)),0)
                    new_waypoint = lane.waypoints[i]
                    new_waypoint.twist.twist.linear.x = v
                    new_lane.waypoints.append(new_waypoint)
            new_lanes = LaneArray()
            new_lanes.lanes.append(new_lane)
            self.waypoint_publisher.publish(new_lanes)

        def tcp_callback(self, msg):
            if msg.data == '1.2': # At v_av = 30 km/h
                deceleration = -2.67
                self.brake_action(decel=deceleration)
            elif msg.data == '1': # At v_av = 25 km/h
                deceleration = -5.08
                self.brake_action(decel=deceleration)
            elif msg.data == '2': # At v_av = 30 km/h or 25 km/h
                deceleration = -7
                self.brake_action(decel=deceleration)
            else:
                pass

        def main(self):
```

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

```
        rospy.spin()


if _name_ == '_main_':
    smrc_pull_over()
```

รูปที่ 61 Code Python ที่ใช้ TCP โปรโตคอลในการเชื่อมต่อ Pedestrian และรถอัตโนมัติ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

บรรณานุกรม

1. Organization, W.H., *Report On Road Safty.* 2023.
2. Yue, L., et al., *In-depth approach for identifying crash causation patterns and its implications for pedestrian crash prevention.* Journal of safety research, 2020. **73**: p. 119-132.
3. International, S., *SAE Levels of Driving Automation.* 2021.
4. Kato, S., et al. *Autoware on board: Enabling autonomous vehicles with embedded systems*. in *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. 2018. IEEE.
5. Yilmaz, A., O. Javed, and M. Shah, *Object tracking: A survey.* Acm computing surveys (CSUR), 2006. **38**(4): p. 13-es.
6. Petrovskaya, A. and S. Thrun, *Model based vehicle tracking for autonomous driving in urban environments.* Proceedings of robotics: science and systems IV, Zurich, Switzerland, 2008. **34**.
7. Abdel-Aty, M., et al., *Evaluation of automated emergency braking system's avoidance of pedestrian crashes at intersections under occluded conditions within a virtual simulator.* Accident Analysis & Prevention, 2022. **176**: p. 106797.
8. Jiménez, F., J.E. Naranjo, and F. García, *An improved method to calculate the time-to-collision of two vehicles.* International Journal of Intelligent Transportation Systems Research, 2013. **11**: p. 34-42.
9. Bae, I., J. Moon, and J. Seo, *Toward a comfortable driving experience for a self-driving shuttle bus.* Electronics, 2019. **8**(9): p. 943.
10. Hu, X., et al., *Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles.* Mechanical systems and signal processing, 2018. **100**: p. 482-500.
11. Ciblooglu, M., U. Karapinar, and M.T. Söylemez. *Hybrid controller approach for an autonomous ground vehicle path tracking problem*. in *2017 25th Mediterranean Conference on Control and Automation (MED)*. 2017. IEEE.
12. Larbwisuthisaroj, S. and N. Noomwongs, *Study on the effect of driver reaction*

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

times to the collision possibility and evasion ability due to the forward collision warning system using driving simulator. 2016.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

ประวัติผู้เขียน

ชื่อ-สกุล	วรดา ขวัญเจริญ
วัน เดือน ปี เกิด	17 กันยายน 2540
สถานที่เกิด	ระยอง
วุฒิการศึกษา	จุฬาลงกรณ์มหาวิทยาลัย
ที่อยู่ปัจจุบัน	34 หมู่ 5 ต.สองสลึง อ.แกลง จ.ระยอง 21110
ผลงานตีพิมพ์	-
รางวัลที่ได้รับ	-

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University