

การพัฒนาระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรกร่วมกับการเลี้ยวหลบโดยใช้หลัก
พลศาสตร์ยานยนต์



บทคัดย่อและแฟ้มข้อมูลฉบับเต็มของวิทยานิพนธ์ตั้งแต่ปีการศึกษา 2554 ที่ให้บริการในคลังปัญญาจุฬาฯ (CUIR)
เป็นแฟ้มข้อมูลของนิสิตเจ้าของวิทยานิพนธ์ ที่ส่งผ่านทางบัณฑิตวิทยาลัย

The abstract and full text of theses from the academic year 2011 in Chulalongkorn University Intellectual Repository (CUIR)
are the thesis authors' files submitted through the University Graduate School.

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2559
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Development of Braking and Steering Maneuver Guidance – Forward Collision Warning
System Using Vehicle Dynamics Approach.

Mr. Sedtawud Larbwisuthisaroj



A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering Program in Mechanical Engineering

Department of Mechanical Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2016

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์

การพัฒนาระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรกร่วมกับการเลี้ยวหลบโดยใช้หลักพลศาสตร์ยานยนต์

โดย

นายเสกฐวุฒิ ลาภวิสุทธิศาโรจน์

สาขาวิชา

วิศวกรรมเครื่องกล

อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

ผู้ช่วยศาสตราจารย์ ดร. นกสิทธิ์ นุ่มวงษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาโทบัณฑิต

.....คณบดีคณะวิศวกรรมศาสตร์

(รองศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

.....ประธานกรรมการ

(รองศาสตราจารย์ ดร. อังคิรี ศรีภคการ)

.....อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก

(ผู้ช่วยศาสตราจารย์ ดร. นกสิทธิ์ นุ่มวงษ์)

.....กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สันหพศ จันทรานุกวัฒน์)

.....กรรมการภายนอกมหาวิทยาลัย

(ดร. ชินะ เพ็ญชาติ)

เสถียรวุฒิ ลามวิสุทธิสารโรจน์ : การพัฒนาระบบเตือนการชนท้ายด้วยการแนะนำการควบคุม เบรกร่วมกับการเลี้ยวหลบโดยใช้หลักพลศาสตร์ยานยนต์ (Development of Braking and Steering Maneuver Guidance – Forward Collision Warning System Using Vehicle Dynamics Approach.) อ.ที่ปรึกษาวิทยานิพนธ์หลัก: ผศ. ดร. นกสิทธิ์ นุ่มวงษ์, 100 หน้า.

การชนท้ายเป็นอุบัติเหตุที่เกิดขึ้นมากที่สุดในประเทศไทย โดยสาเหตุหลักที่ทำให้เกิดอุบัติเหตุจากการชนท้ายขึ้น คือความผิดพลาดจากผู้ขับขี่ ซึ่งการแก้ปัญหาที่เกิดจากความผิดพลาดจากผู้ขับขี่ที่สามารถแก้ไขได้โดยระบบ Active Safety System ซึ่งในงานวิจัยนี้มีจุดประสงค์เพื่อพัฒนาระบบเตือนการชนด้านหน้า ซึ่งเป็นหนึ่งในระบบ Active Safety System เช่นกัน ระบบเตือนการชนด้านหน้ามีการทำงานโดยจะเตือนให้ผู้ขับขี่ทราบถึงโอกาสที่จะเกิดการชนขึ้น ซึ่งวิธีที่ผู้โดยสารจะตอบสนองเพื่อหลีกเลี่ยงการชนนั้นโดยส่วนใหญ่จะใช้วิธีการเบรก ซึ่งอาจไม่เพียงพอในบางสภาวะ เช่น ที่ความเร็วสูง เป็นต้น ในงานวิจัยนี้จึงพิจารณาการเลี้ยวหลบเพื่อเป็นวิธีเพิ่มเติมในการหลีกเลี่ยงการชน ซึ่งเส้นทางการเคลื่อนที่ของรถยนต์จากการเบรก และการเลี้ยวหลบถูกคำนวณด้วยแบบจำลองทางพลศาสตร์ยานยนต์ และนำผลที่ได้ไปคำนวณหาค่า เวลาที่จะเกิดการชน (Time-To-Collision (TTC)) ต่ำสุดที่ผู้ขับขี่จะสามารถหลีกเลี่ยงการชนได้ ซึ่ง TTC เป็นตัวแปรสำคัญที่ถูกใช้ใน ระบบเตือนการชนด้านหน้า โดยทำหน้าที่ในการเป็นขอบเขตที่ใช้ในการเตือนผู้ขับขี่ และใช้พิจารณาถึงโอกาสในการหลีกเลี่ยงการชน รวมถึงวิธีการที่ใช้ในการหลีกเลี่ยงการชน ระบบเตือนการชนด้านหน้าในงานวิจัยนี้ ใช้ระบบตรวจจับวัตถุด้านหน้าที่ใช้กระบวนการทาง Image Processing เพื่อคำนวณหาค่า TTC เพื่อนำมาเตือนตามแผนภาพค่า TTC ต่ำสุดที่ได้จากการใช้แบบจำลองทางพลศาสตร์ยานยนต์ ซึ่งระบบถูกนำมาทดสอบด้วยโปรแกรมจำลองการขับขี่ และทดสอบด้วยรถยนต์จริงในสภาวะแวดล้อมแบบปิด เพื่อศึกษาพฤติกรรมการตอบสนองของผู้ขับขี่ต่อระบบเตือนการชนด้านหน้า ผลจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสถานที่ทดสอบที่มีสภาวะแบบปิด ที่ความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง แสดงให้เห็นว่าระบบเตือนการชนด้านหน้าสามารถใช้งานกับรถยนต์จริงได้ แต่ค่า TTC ที่ระบบเตือนการชนด้านหน้าทำการเตือนจริง มีความล่าช้าซึ่งเกิดจาก เวลาในการประมวลผล และเวลาในการตอบสนองของผู้ขับขี่ จึงต้องทำการชดเชยค่า TTC 0.7 วินาที เพื่อให้เวลาในการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 1 วินาที ตามต้องการ

ภาควิชา วิศวกรรมเครื่องกล

ลายมือชื่อนิสิต

สาขาวิชา วิศวกรรมเครื่องกล

ลายมือชื่อ อ.ที่ปรึกษาหลัก

ปีการศึกษา 2559

5770344421 : MAJOR MECHANICAL ENGINEERING

KEYWORDS: DRIVER REACTION TIME, STEERING AND BRAKING GUIDANCE, FORWARD COLLISION WARNING SYSTEM, EVASIVE MANEUVER / MINIMUM TIME-TO-COLLISION / TIME-TO-COLLISION

SEDTAWUD LARBWISUTHISAROJ: Development of Braking and Steering Maneuver Guidance – Forward Collision Warning System Using Vehicle Dynamics Approach.. ADVISOR: ASST. PROF. NUKSIT NOOMWONGS, Ph.D., 100 pp.

Rear-end collision is the most frequently occurring accident in Thailand. The reasons of the rear-end collision are mostly the driver's errors, which can be solved by the active safety system. The objective of this research is to develop the forward collision warning system, which is active safety system. The forward collision warning system will warn the driver if the accident possible. The drivers mostly use braking as the evasive maneuver. Only braking is not enough to prevent the accident in some cases, such as in high velocity. The steering evasive maneuver is considered as an addition evasive maneuver in this research. Both braking and steering evasive maneuver were predicted by vehicle dynamics model. Therefore, the evasive paths were used to calculated minimum Time-To-Collision (TTC) that the driver can prevent the accident. TTC is an important variable that used as a threshold of the forward collision warning system, and also, used to decide the evasive maneuver. Image processing was used in this research as a vehicle detection system, in addition, to calculate TTC. The forward collision warning system was tested using driving simulator and by the real vehicle in the closed environment. The experiments were carried out to study the driver response to the forward collision warning system. The results of the real vehicle experiment show that the forward collision warning system can be used with the real vehicle at 20 – 40 km/hr. However, the actual TTCs are delayed by the processing time and driver reaction time. According to the research result, TTC should be compensated by 0.7 s.

Department: Mechanical Engineering Student's Signature

Field of Study: Mechanical Engineering Advisor's Signature

Academic Year: 2016

กิตติกรรมประกาศ

ขอกราบขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. นักสิทธิ์ นุ่มวงษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ ที่กรุณา สละเวลาอันมีคุณค่าของท่านเพื่อให้ความรู้ คำแนะนำ คำปรึกษา และข้อคิดต่าง ๆ ที่เป็นประโยชน์ในการทำวิทยานิพนธ์นี้ รวมทั้งยังจัดหาทุน อุปกรณ์ พร้อมทั้งยังติดต่อหาผู้สนับสนุน อนุเคราะห์สถานที่ ที่จำเป็นต่อวิทยานิพนธ์นี้ จนทำให้วิทยานิพนธ์สำเร็จรุดังไปด้วยดี

ขอกราบขอบพระคุณ ประธานกรรมการสอบวิทยานิพนธ์ รองศาสตราจารย์ ดร. อังคิรี ศรีภคากร ท่านกรรมการสอบวิทยานิพนธ์ ผู้ช่วยศาสตราจารย์ ดร. สัณห์ยศ จันทรานุกวัฒน์ และท่านกรรมการผู้ทรงคุณวุฒิภายนอกมหาวิทยาลัย ดร. ชินะ เพ็ญชาติ ที่กรุณารับเป็นคณะกรรมการสอบ และตรวจสอบวิทยานิพนธ์

นอกจากนี้ยังขอกราบขอบพระคุณ ผู้ที่มีส่วนเกี่ยวข้องผู้อื่น ๆ ที่ช่วยเป็นกำลังใจ แนะนำ ช่วยเหลือ ผู้จัดทำในการทำวิทยานิพนธ์นี้ให้สำเร็จลุล่วงไปได้ด้วยดี และที่สำคัญต้องขอกราบขอบพระคุณ บิดา มารดา ที่คอยสนับสนุนทั้งด้านทุนการศึกษา รวมถึงคำแนะนำ เป็นกำลังใจ ให้ผู้จัดทำได้ทำวิทยานิพนธ์สำเร็จมา ณ โอกาสนี้ด้วย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ.....	ช
สารบัญรูปภาพ	ญ
สารบัญตาราง.....	ฒ
บทที่ 1 บทนำ	1
1.1 ที่มา และความสำคัญ.....	1
1.2 วัตถุประสงค์.....	3
1.3 ขอบเขตของโครงการ	3
1.4 ขั้นตอนการดำเนินงาน	4
1.5 แผนการดำเนินงาน	4
1.6 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 บริหารศรัทธาธรรม	6
2.1 งานวิจัยเรื่องระบบป้องกันการชนด้านหน้า	6
2.2 อุปกรณ์ตรวจจับสำหรับระบบป้องกันการชนด้านหน้า	7
2.3 ระบบตรวจจับวัตถุ.....	8
2.4 ปัญหาของงานวิจัยที่ผ่านมา	11
บทที่ 3 การออกแบบระบบเตือนการชนท้าย และการทดสอบ	12
3.1 การออกแบบระบบตรวจจับรถยนต์ด้านหน้า	12
3.1.1 กล้องสเตอริโอวิชัน	13
3.1.2 แผนภาพ U Disparity และ V Disparity.....	15

3.1.3 Haar-like feature	17
3.1.4 กระบวนการ Machine Learning แบบ AdaBoost	19
3.2 Model Simulation	22
3.2.1 3 Degree of Freedom Full Body Vehicle Dynamics Model	23
3.2.2 Relaxation Length Tire Model	24
3.2.3 Wheel Model	25
3.2.4 Driver Model	26
3.2.5 Time-To-Collision	28
3.2.6 การจำลองเส้นทางการหลีกเลี่ยงการชน	29
3.2.7 การเปรียบเทียบเส้นทางการหลีกเลี่ยงการชนที่ได้จาก Geometrical Approach และ Vehicle Dynamics Approach	31
3.3 แผนภาพ TTC ต่ำสุด	33
3.4 ผลของ Driver Reaction Time ต่อระบบ Forward Collision Warning System	37
3.5 ขั้นตอนการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสถานะแวดล้อมแบบ ปิด	41
บทที่ 4 ผลการทดสอบ	43
4.1 Model Evaluation	43
4.2. TTC ที่ได้จากการทดสอบ	50
4.3 ความล่าช้าเนื่องจาก Driver Reaction Time	52
4.4 เปรียบเทียบ Actual Warning TTC กับ Min TTC Diagram	54
4.5 ตัวอย่างการออกแบบค่า TTC ที่ใช้ในการเตือนผู้ขับขี่	55
บทที่ 5 สรุปผล และข้อเสนอแนะ	59
5.1 สรุปผลงานวิจัย	59
5.2 ข้อเสนอแนะ	61

รายการอ้างอิง	62
ภาคผนวก ก อุปกรณ์ที่ใช้ในการทดสอบ	66
ภาคผนวก ข ระบบเตือนการชนด้านหน้า	72
ประวัติผู้เขียนวิทยานิพนธ์	100



สารบัญรูปภาพ

รูปที่ 1 สาเหตุที่ทำให้เกิดอุบัติเหตุจากการชนท้ายรถยนต์คันอื่น.....	1
รูปที่ 2 การควบคุมรถในขณะก่อนการเกิดอุบัติเหตุ.....	2
รูปที่ 3 ภาพจำลองสถานการณ์ในการทดสอบระบบ Collision Avoidance	6
รูปที่ 4 ภาพจำลองสถานการณ์การทำงานของระบบ Collision Avoidance	7
รูปที่ 5 ภาพเปรียบเทียบระหว่างภาพ (a) Disparity (b) V-Disparity จากภาพต้นฉบับ (c).....	9
รูปที่ 6 ภาพเปรียบเทียบระหว่างภาพ (a) Disparity (b) U-Disparity จากภาพต้นฉบับ (c).....	9
รูปที่ 7 แผนผังของระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรกร่วมกับการเลี้ยวหลบ .	12
รูปที่ 8 การติดตั้งกล้อง 2 ตัว แบบ Stereo Camera.....	13
รูปที่ 9 ตัวอย่างต่าง ๆ ในการติดตั้งกล้อง 2 ตัว แบบ Stereo Camera [11].....	14
รูปที่ 10 แผนภาพ Disparity Map	15
รูปที่ 11 ภาพการสร้างเมทริกซ์ของ V-Disparity Map จากเมทริกซ์ของ Disparity Map [12].....	16
รูปที่ 12 (a) V-Disparity Map (b) Disparity Map และ (c) U-Disparity Map.....	17
รูปที่ 13 ตัวอย่างของสี่เหลี่ยมลักษณะต่าง ๆ ในหน้าต่างที่ใช้ในการตรวจจับ	17
รูปที่ 14 หน้าต่างที่ใช้ในการตรวจจับในรูปแบบต่าง ๆ [13].....	18
รูปที่ 15 การนำ Haar-like Feature มาใช้ในการตรวจจับรถยนต์	19
รูปที่ 16 อัลกอริทึมของการเรียนรู้โดย AdaBoost [14]	20
รูปที่ 17 Positive Samples.....	21
รูปที่ 18 Negative Samples.....	21
รูปที่ 19 ข้อมูลอ้างอิงที่ใช้ในการตรวจจับรถยนต์หลังจากการทำ Machine Learning	22
รูปที่ 20 แผนภาพ แบบจำลองทางพลศาสตร์ยานยนต์.....	23
รูปที่ 21 การจำลอง Vehicle Dynamics Model ผ่านโปรแกรม Matlab Simulink	23
รูปที่ 22 3 Degree of Freedom Full Body Vehicle Model [18]	24

รูปที่ 23	กลศาสตร์ของยางขณะทำการเบรก [19]	26
รูปที่ 24	ภาพจำลองสถานที่ทดสอบของ แบบจำลองผู้ขับขี่ [18]	26
รูปที่ 25	ความสัมพันธ์ระหว่างความล่าช้าที่เกิดจากการตอบสนองของผู้ขับขี่กับความเร็วยานต์ [18]	27
รูปที่ 26	ความสัมพันธ์ระหว่าง Proportional Gain ในการเลี้ยงกับความเร็วยานต์ [18].....	27
รูปที่ 27	ความล่าช้าที่เกิดจากพฤติกรรมของการมองทางด้านหน้าของผู้ขับขี่ในการเลี้ยงกับความเร็วยานต์ [18].....	28
รูปที่ 28	เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 20 กิโลเมตรต่อชั่วโมง	30
รูปที่ 29	เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง	30
รูปที่ 30	เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 60 กิโลเมตรต่อชั่วโมง	30
รูปที่ 31	เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 80 กิโลเมตรต่อชั่วโมง	31
รูปที่ 32	เปรียบเทียบเส้นทางการเคลื่อนที่ของรถยนต์ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน	31
รูปที่ 33	เปรียบเทียบเส้นทางการเคลื่อนที่ของรถยนต์ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน	32
รูปที่ 34	แผนภาพ แบบจำลองทางพลศาสตร์ยานยนต์ ที่มี Input เป็น เส้นทางการหลีกเลี่ยงการชนที่ถูกคำนวณด้วยวิธี Geometrical Approach	33
รูปที่ 35	ระยะทางเหลือระหว่างเส้นทางการเคลื่อนที่ของรถยนต์ และสิ่งกีดขวาง.....	33
รูปที่ 36	เส้นทางการหลีกเลี่ยงการชนด้วยการเบรกเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง เมื่อเตือนผู้ขับขี่ที่ TTC ต่ำสุด เท่ากับ 0.9 วินาที	34
รูปที่ 37	เส้นทางการหลีกเลี่ยงการชนด้วยการเลี้ยวหลบเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง เมื่อเตือนผู้ขับขี่ที่ TTC ต่ำสุด เท่ากับ 0.66 วินาที.....	34
รูปที่ 38	ค่า TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชนที่ความเร็วต่าง ๆ	35
รูปที่ 39	บริเวณภายในแผนภาพ TTC ต่ำสุดที่ทำให้เกิดความสามารถในการหลีกเลี่ยงการชนต่างกัน.....	35
รูปที่ 40	เส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ A ของแผนภาพ TTC ต่ำสุด.....	36

รูปที่ 41 เส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ B ของ แผนภาพ TTC ต่ำสุด.....	36
รูปที่ 42 เส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ C ของ แผนภาพ TTC ต่ำสุด.....	36
รูปที่ 43 เส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ D ของ แผนภาพ TTC ต่ำสุด.....	37
รูปที่ 44 สถานการณ์จำลองที่ใช้ใน Driving Simulator [5]	38
รูปที่ 45 ความสัมพันธ์ระหว่าง Driver Reaction Time ที่ความเร็วต่าง ๆ	39
รูปที่ 46 กราฟแสดงผลกระทบของ Driver Reaction Time ที่ส่งผลต่อ TTC	40
รูปที่ 47 กราฟแสดง TTC ที่ถูกชดเชยด้วย Driver Reaction Time แล้ว	40
รูปที่ 48 การทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิด ขณะที่รถยนต์เคลื่อนที่เข้าใกล้เป้าหมาย	42
รูปที่ 49 ภาพขณะระบบเตือนการชนด้านหน้าเตือนให้ผู้ขับขี่เบรกเพื่อหลีกเลี่ยงการชนด้วย LED	42
รูปที่ 50 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะเบรกเพื่อ หลีกเลี่ยงการชนที่ความเร็ว 20 กม./ชม.	43
รูปที่ 51 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะเบรกเพื่อ หลีกเลี่ยงการชนที่ความเร็ว 30 กม./ชม.	44
รูปที่ 52 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะเบรกเพื่อ หลีกเลี่ยงการชนที่ความเร็ว 40 กม./ชม.	44
รูปที่ 53 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการ ทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 20 กิโลเมตรต่อชั่วโมง	45
รูปที่ 54 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการ ทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง	46
รูปที่ 55 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการ ทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง	46

รูปที่ 56 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะทีเลี้ยวหลบเพื่อหลีกเลี่ยงการชนความเร็ว 20 กม./ชม.....	47
รูปที่ 57 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะทีเลี้ยวหลบเพื่อหลีกเลี่ยงการชนความเร็ว 30 กม./ชม.....	48
รูปที่ 58 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะทีเลี้ยวหลบเพื่อหลีกเลี่ยงการชนความเร็ว 40 กม./ชม.....	49
รูปที่ 59 กราฟแสดงความสัมพันธ์ระหว่าง TTC และ Processing Time กับความเร็วของรถยนต์ ขณะเบรกเพื่อหลีกเลี่ยงการชน	50
รูปที่ 60 กราฟแสดงความสัมพันธ์ระหว่าง TTC และ Processing Time กับความเร็วของรถยนต์ ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน	51
รูปที่ 61 กราฟแสดงความสัมพันธ์ระหว่าง Driver Reaction Time กับความเร็วของรถยนต์ของDriving Simulator และการทดสอบ	52
รูปที่ 62 กราฟแสดงความสัมพันธ์ระหว่าง Driver Reaction Time กับความเร็วของรถยนต์ของDriving Simulator และ การทดสอบจริง ของผู้ขับขี่คนเดียวกัน	53
รูปที่ 63 กราฟเปรียบเทียบค่า TTC ที่ระบบเตือน กับ แผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน.....	54
รูปที่ 64 กราฟเปรียบเทียบค่า TTC ที่เหลือให้ผู้ขับขี่ได้ควบคุมรถจริง กับ แผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน.....	55
รูปที่ 65 แสดงค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 1 วินาที	56
รูปที่ 66 ตัวอย่างค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนแปรผันตามความเร็ว	57
รูปที่ 67 ตัวอย่างค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนแปรผันตามความเร็ว	57

รูปที่ 68	กล้อง IDS 1220LE พร้อมเลนส์ที่มีระยะโฟกัส 8 มิลลิเมตร	67
รูปที่ 69	คอมพิวเตอร์ที่ใช้ในการประมวลผล	68
รูปที่ 70	Microcontroller Arduino Uno.....	68
รูปที่ 71	Neo Pixel LED Stick	69
รูปที่ 72	กล้อง Gopro Hero 5 Black	69
รูปที่ 73	รถยนต์ที่ใช้ในการทดสอบ	70
รูปที่ 74	ภาพรถยนต์จำลอง	70
รูปที่ 75	GPS พร้อมเสาอากาศ	71
รูปที่ 76	ตัวอย่างภาพระบบเตือนการชนด้านหน้าขณะใช้งาน	73
รูปที่ 77	แสดง Code ที่ใช้ในการพัฒนาระบบเตือนการชนด้านหน้า	80
รูปที่ 78	ภาพแสดงโปรแกรมที่ใช้ในการปรับแต่งค่าเพื่อสร้าง Disparity Map.....	80
รูปที่ 79	ภาพแสดงCodeโปรแกรมที่ใช้ในการปรับแต่งค่าเพื่อสร้าง Disparity Map.....	85
รูปที่ 80	ภาพแสดงโปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Canny Edge Detection	85
รูปที่ 81	ภาพแสดง Code โปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Canny Edge Detection....	88
รูปที่ 82	ภาพแสดง โปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Hough Transformation	89
รูปที่ 83	ภาพแสดง Code โปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Hough Transformation .	95
รูปที่ 84	แผนผังของระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรกร่วมกับการเลี้ยว หลบ	96
รูปที่ 85	แผนภาพจำลองระบบเตือนการชนด้านหน้า	97
รูปที่ 86	ภาพแสดง Code โปรแกรมที่ใช้ในการเตือนผู้ขับขี่ด้วย Arduino Uno	99

สารบัญตาราง

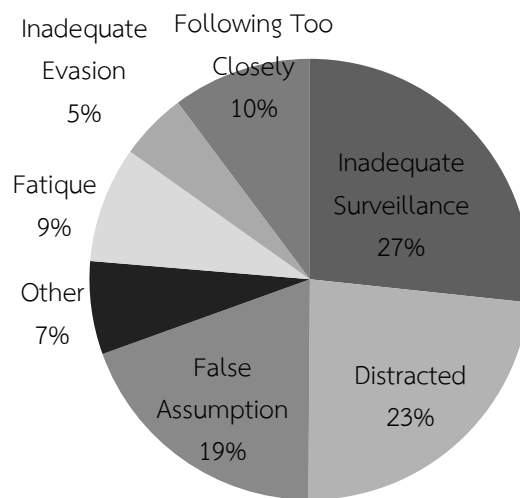
ตารางที่ 1	แผนการดำเนินงาน.....	4
ตารางที่ 2	ตัวอย่าง คุณสมบัติ ข้อดี และข้อเสีย ของเซ็นเซอร์ชนิดต่าง ๆ [7].....	8
ตารางที่ 3	ตัวอย่าง วิธีการต่าง ๆ ในขั้นตอน Features Extraction และ Classification	10
ตารางที่ 4	ตัวแปรที่ถูกเก็บค่าขณะการทดสอบด้วย Driving Simulator	38
ตารางที่ 5	ตัวแปรที่ถูกเก็บค่าขณะการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง	41
ตารางที่ 6	ค่าความล่าช้าเฉลี่ยจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง	58
ตารางที่ 7	คุณสมบัติของกล้อง IDS 1220LE.....	67
ตารางที่ 8	คุณสมบัติของคอมพิวเตอร์ที่ใช้ในการประมวลผล	68
ตารางที่ 9	คุณสมบัติของ Microcontroller Arduino Uno.....	69
ตารางที่ 10	คุณสมบัติของกล้อง Gopro Hero 5 Black.....	70
ตารางที่ 11	คุณสมบัติ GPS ที่ใช้ในการเก็บข้อมูล ตำแหน่ง และความเร็วของรถยนต์	71

บทที่ 1

บทนำ

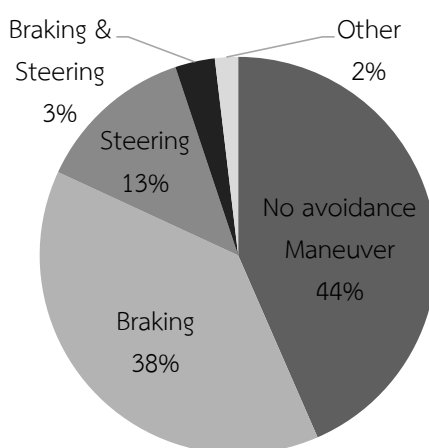
1.1 ที่มา และความสำคัญ

ในปี พ.ศ. 2558 องค์การอนามัยโลก (World Health Organization) ได้ประมาณการเสียชีวิตที่เกิดจากอุบัติเหตุทางท้องถนน โดยประเทศไทยถูกประมาณการเสียชีวิตไว้ที่ 36.2 รายต่อประชากร 100,000 คน [1] ซึ่งมีอันดับมากเป็นอันดับ 2 ของโลก รองจากประเทศลิเบีย จากการทำการสำรวจลักษณะการเกิดอุบัติเหตุทางรถยนต์ในประเทศไทย อุบัติเหตุแบบชนท้ายเกิดขึ้นมากที่สุดที่ 24% [2] โดยความเสียหายทางเศรษฐกิจ และความเสียหายต่อชีวิตคน ที่เกิดจากอุบัติเหตุแบบชนท้ายนั้น มากเป็นอันดับที่ 2 จากการสำรวจของ National Highway Traffic Safety Administration (NHTSA) [3] จากข้อมูลดังกล่าวมาจะเห็นว่า การเกิดอุบัติเหตุจากการชนท้ายรถยนต์คันอื่น ก่อให้เกิดความเสียหายอย่างมาก ดังนั้น การลดความเป็นไปได้ในการเกิดอุบัติเหตุชนิดนี้ จะสามารถลดการสูญเสียทั้งทางเศรษฐกิจ และชีวิตให้กับประเทศได้



รูปที่ 1 สาเหตุที่ทำให้เกิดอุบัติเหตุจากการชนท้ายรถยนต์คันอื่น

สาเหตุที่ทำให้เกิดอุบัติเหตุจากการชนท้าย มีอยู่ 6 สาเหตุหลัก ๆ [3] ดังที่แสดงใน รูปที่ 1 ซึ่งเกิดมาจากความผิดพลาดของคนขับทั้งสิ้น โดยปัญหาดังกล่าวสามารถถูกแก้ไขได้โดยใช้ระบบที่เรียกว่า Active Safety System ซึ่งเป็นระบบที่มีหน้าที่ป้องกันไม่ให้อุบัติเหตุเกิดขึ้น ซึ่งในปัจจุบันระบบ Active Safety System ที่ถูกคิดค้นขึ้นมาไม่ว่าจะอยู่ในช่วงการวิจัย หรือถูกนำมาใช้งานในรถยนต์แล้ว มีอยู่หลายประเภท เช่น Collision Avoidance System, Automatic Braking, Adaptive Cruise Control, Lane Departure Warning System.



รูปที่ 2 การควบคุมรถในขณะก่อนการเกิดอุบัติเหตุ

นอกจากนี้การควบคุมรถในขณะก่อนการเกิดอุบัติเหตุเป็นปัจจัยในการหลีกเลี่ยงการเกิดอุบัติเหตุ และส่งผลต่อความรุนแรงของอุบัติเหตุด้วยเช่นกัน รูปที่ 2 เป็นกราฟแสดงการควบคุมรถในขณะก่อนการเกิดอุบัติเหตุ ซึ่งถูกสำรวจโดย National Highway Traffic Safety Administration (NHTSA) [3] โดยเป็นการสำรวจข้อมูลทางอุบัติเหตุหลังการเกิดอุบัติเหตุขึ้นแล้ว เพื่อเป็นการแสดงให้เห็นถึงการควบคุมรถในขณะก่อนเกิดอุบัติเหตุ เพื่อพยายามหลีกเลี่ยงการชน จะเห็นว่าผู้ขับขี่ จำนวน 44 % ไม่ทำการควบคุมรถเพื่อหลีกเลี่ยงการชน รองลงมา จำนวน 38 % ใช้เบรกเพื่อพยายามหลีกเลี่ยงการชนขึ้น ซึ่งก็ยังไม่สามารถหลีกเลี่ยงได้อยู่ดี (ข้อมูลที่ได้เป็นข้อมูลที่ได้จากการเก็บข้อมูลหลังการเกิดอุบัติเหตุแล้ว) ทำให้การใช้เบรกอย่างเดียวไม่พอในการหยุดรถเพื่อไม่ให้เกิดอุบัติเหตุ วิธีการอีกวิธีที่สามารถหลีกเลี่ยงการเกิดอุบัติเหตุได้คือการควบคุมรถให้เลี้ยวหลบรถคันข้างหน้า โดยการควบคุมรถให้เลี้ยวหลบรถคันข้างหน้าจะมีประสิทธิภาพในการลดอุบัติเหตุลงได้มากกว่าการเบรกที่ความเร็วสูงกว่า [4]

ดังนั้นงานวิจัยชิ้นนี้จึงมีแนวคิดที่จะพัฒนาระบบ Active Safety System เพื่อที่จะลดอุบัติเหตุจากสาเหตุดังกล่าว โดยใช้ระบบ Forward Collision Guidance เนื่องด้วยความสะดวกในการติดตั้งที่มีมากกว่า และ มีค่าใช้จ่ายที่น้อยกว่า ระบบ Autonomous Emergency Braking

1.2 วัตถุประสงค์

1. เพื่อพัฒนาระบบ Forward Collision Guidance ที่สามารถใช้งานแบบ Stand Alone เพื่อป้องกันการเกิดอุบัติเหตุจากการชนท้ายรถยนต์ด้านหน้า
2. เพื่อพัฒนาระบบ Forward Collision Guidance ที่สามารถนำมาใช้งานได้กับรถยนต์จริงในสนามทดสอบที่มีสภาพแวดล้อมแบบปิด
3. เพื่อพัฒนาระบบ Forward Collision Guidance ที่สามารถแนะนำวิธีหลีกเลี่ยงการชนให้ผู้ขับขี่ได้

1.3 ขอบเขตของโครงการ

1. ใช้ Dynamics Model ของรถยนต์ในการคาดการณ์เส้นทางการเคลื่อนที่ของรถยนต์
2. ทดสอบผลของการใช้ระบบ Forward Collision Guidance ต่อผู้ขับขี่ด้วย Driving Simulator
3. ทดสอบผลของการใช้ระบบ Forward Collision Guidance ต่อผู้ขับขี่ในสนามทดสอบที่มีสภาพแวดล้อมแบบปิด
4. ระบบ Forward Collision Guidance ทำการแนะนำผู้ขับขี่เพื่อหลีกเลี่ยงการชนด้วย การเบรกและการเลี้ยวหลบ
5. ระบบ Forward Collision Guidance ใช้ความเร็วของรถยนต์เป็นแบบคงที่ตามที่กำหนดในการทดสอบแต่ละครั้ง
6. ระบบ Forward Collision Guidance ถูกออกแบบมาให้ใช้กับเหตุการณ์ที่กำหนดในงานวิจัยนี้เท่านั้น
7. ระบบ Forward Collision Guidance จะถูกทดสอบในความเร็วที่ 20 – 40 กม.ต่อ ชม. เพื่อความปลอดภัยของผู้ทดสอบ
8. รถยนต์ที่ใช้ในการทดสอบเป็นรถยนต์ส่วนบุคคลทั่วไป ขนาด 4 ที่นั่ง
9. เป้าหมายรูปรถยนต์ที่ทำมาจากโพลีคาร์บอเนตใช้เพื่อเป็นเป้าหมายในการทดสอบเพื่อความปลอดภัยในการทดสอบ
10. รถยนต์ที่ใช้ในการทดสอบมีเพียงระบบ ABS ติดตั้งในรถยนต์ โดยไม่รวมระบบ Forward Collision Guidance ที่กล่าวในงานวิจัยชิ้นนี้

ขั้นตอน	2558		2559										2560									
	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	
4.ทดสอบระบบตรวจจับรถยนต์ ด้านหน้า																						
5.ออกแบบ Dynamics Model เพื่อใช้ในการคำนวณ เส้นทางการหลบหลีก																						
6.คำนวณเส้นทางการหลบหลีก และสร้างแผนภาพ TTC ต่ำสุด ของการเบรก และเลี้ยวหลบ เพื่อหลีกเลี่ยงการชน																						
7.ทดสอบผลของการใช้ระบบ Forward Collision Guidance ต่อผู้ขับขี่ด้วย Driving Simulator																						
8.พัฒนาระบบ Forward Collision Guidance สำหรับ การใช้งานในรถยนต์จริง																						
9.ทดสอบ Forward Collision Guidance ในสถานทดสอบที่มี สภาพแวดล้อมแบบปิด																						
10.สรุปผล และจัดทำเอกสาร																						

1.6 ประโยชน์ที่คาดว่าจะได้รับ

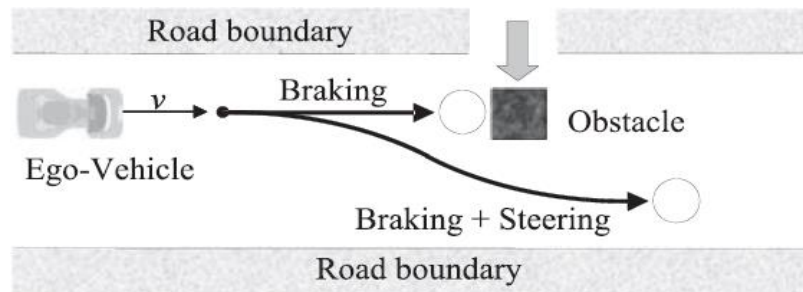
1. ระบบ Forward Collision Guidance ที่สามารถใช้งานแบบ Stand Alone เพื่อป้องกันการเกิดอุบัติเหตุจากการชนท้ายรถยนต์ด้านหน้า
2. ระบบ Forward Collision Guidance ที่สามารถนำมาใช้งานได้กับรถยนต์จริง ในสนามทดสอบที่มีสภาพแวดล้อมแบบปิด
3. ระบบ Forward Collision Guidance ที่สามารถแนะนำวิธีหลีกเลี่ยงการชนให้ผู้ขับขี่ได้
4. แผนภาพ Time-To-Collision ต่ำสุดที่ผู้ขับขี่สามารถหลีกเลี่ยงการชนได้ด้วยวิธีต่าง ๆ
5. ระบบตรวจจับรถยนต์ด้านหน้าที่สามารถตรวจจับรถยนต์ และบอกระยะห่างได้

บทที่ 2

ปริทรรศน์วรรณกรรม

2.1 งานวิจัยเรื่องระบบป้องกันการชนด้านหน้า

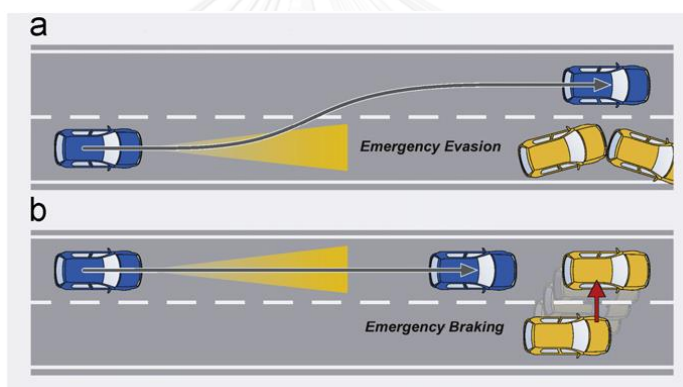
บทความวิจัยเรื่อง “Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory” [5] เป็นงานวิจัยที่ออกแบบระบบ Autonomous collision avoidance ด้วยการเบรกและเลี้ยวหลบของรถยนต์ โดยตรวจจับสิ่งกีดขวางด้านหน้ารถยนต์ และระยะห่างระหว่างรถยนต์กับสิ่งกีดขวางด้วย LIDAR โดยที่รถยนต์เคลื่อนที่ด้วยความเร็ว 30 กม./ชม. สิ่งกีดขวางมีความเร็วใกล้เคียงกับคนเดินถนนที่ 1 กม./ชม. โดยมีลักษณะการเดินเข้ามาจากทางด้านข้างถนนบริเวณหน้ารถยนต์ ซึ่งถ้ามีความเป็นไปได้ที่จะเกิดอุบัติเหตุเกิดขึ้น ระบบจะมีการป้องกันการเกิดอุบัติเหตุได้ทั้งสิ้น 4 วิธีซึ่งแต่ละวิธีจะมีเงื่อนไขที่ต้องบรรลุต่างกันดังนี้ 1) รถยนต์เบรกตัวด้วยแรงเบรกสูงสุดโดยรถยนต์จะหยุดที่ตำแหน่งก่อนถึงสิ่งกีดขวาง 2) รถยนต์เบรกตัวด้วยแรงเบรกสูงสุดโดยรถยนต์จะหยุดที่ตำแหน่งหลังสิ่งกีดขวาง โดยการเบรกทำให้สิ่งกีดขวางสามารถเคลื่อนที่หลบออกไปจากเส้นทางการเคลื่อนที่ของรถยนต์ได้ทันก่อนที่รถยนต์จะมาถึงตำแหน่งที่สิ่งกีดขวางอยู่ และจึงหยุดในลำดับต่อไป 3) ในกรณีที่ไม่สามารถเบรกรถยนต์เพื่อป้องกันการเกิดอุบัติเหตุได้ การเลี้ยวหลบสิ่งกีดขวางจึงถูกนำมาใช้ โดยในกรณีนี้จะ เป็นกรณีที่ช่องทางการจราจรที่รถยนต์จะทำการเปลี่ยนไปมีความกว้างมากพอจนที่ไม่จำเป็นต้องนำมาพิจารณาในการคำนวณเส้นทางการเคลื่อนที่ 4) กรณีนี้จะมีลักษณะคล้ายในกรณีที่ 3 แต่จะแตกต่างกันที่ช่องทางการจราจรที่รถยนต์ได้ทำการเปลี่ยนไปมีความกว้างน้อยลงมาทำให้ต้องนำมาพิจารณาในการคำนวณเส้นทางการเคลื่อนที่ของรถยนต์ โดยการคำนวณเส้นทางการเคลื่อนที่ของรถยนต์ในขณะที่มีการเลี้ยวหลบเป็นการคำนวณเส้นทางด้วยวิธีการทางเรขาคณิต



รูปที่ 3 ภาพจำลองสถานการณ์ในการทดสอบระบบ Collision Avoidance

ของบทความวิจัยเรื่อง “Autonomous collision avoidance system by combined control of steering and braking using geometrically optimised vehicular trajectory” [5]

บทความวิจัยเรื่อง “Anticollision system PRORETA with automatic braking and steering” [6] เป็นงานวิจัยเกี่ยวกับระบบ Collision-avoidance ด้วยการเบรก และการเลี้ยวหลบ โดยใช้ Camera Sensor และ Long Range Radar ในการตรวจจับสิ่งกีดขวาง และระยะห่างระหว่างรถยนต์กับสิ่งกีดขวาง โดยที่ระบบมีการทำงานในลักษณะที่รถยนต์คันที่ติดตั้งระบบ และรถยนต์ที่กีดขวางเคลื่อนที่ไปในทิศทางเดียวกัน ในช่องจราจรเดียวกัน โดยมีการป้องกันอุบัติเหตุทั้งสิ้น 2 วิธี โดยแต่ละวิธีมีรายละเอียดและเงื่อนไขดังนี้ 1) Blocked Lane คือการที่เส้นทางการเคลื่อนที่ถูกรบกวนโดยวัตถุซึ่งเป็นการขวางในลักษณะที่อยู่ในช่องทางเดียวกับรถยนต์ที่ขับ ซึ่งจะมีการป้องกันไม่ให้เกิดอุบัติเหตุโดยการเลี้ยวรถยนต์หลบไปอีกช่องทางหนึ่ง 2) Cutting-In Vehicle คือการที่เส้นทางการเคลื่อนที่ถูกรบกวนโดยการที่รถยนต์เคลื่อนที่เข้ามาขวางเส้นทางการเคลื่อนที่จากอีกช่องจราจรหนึ่ง ซึ่งจะมีการป้องกันการเกิดอุบัติเหตุโดยการเบรกรถยนต์ด้วยแรงเบรกสูงสุด ซึ่งงานวิจัยมีคำนวณการเคลื่อนที่ของรถยนต์โดยการตั้งสมมติฐานให้รถยนต์เคลื่อนที่มีลักษณะเส้นทางคล้ายตัว “S” สมการ Sigmoidal Function จึงถูกนำมาใช้ในการคำนวณทิศทางเคลื่อนที่ของรถยนต์



รูปที่ 4 ภาพจำลองสถานการณ์การทำงานของระบบ Collision Avoidance

ของบทความวิจัยเรื่อง “Anticollision system PRORETA with automatic braking and steering” ในสถานการณ์ต่าง ๆ ที่กล่าวไว้ [6]

2.2 อุปกรณ์ตรวจจับสำหรับระบบป้องกันการชนด้านหน้า

บทความวิจัยเรื่อง “Vehicles detection techniques for collision avoidance systems: A reviews” [7] บทความนี้เป็นการเปรียบเทียบงานวิจัยต่าง ๆ ในเรื่องการตรวจจับยานพาหนะที่นำมาใช้กับระบบ Collision Avoidance โดยมีการแบ่งเป็นหัวข้อต่าง ๆ 3 หัวข้อคือ 1) เซ็นเซอร์ 2) ขั้นตอนการตรวจจับยานพาหนะ และ 3) การติดตามเป้าหมายหลังจากที่โดนตรวจจับแล้ว ซึ่งในส่วนนี้จะขอกล่าวในส่วนของเซ็นเซอร์ที่ใช้ในการตรวจจับ ซึ่งจากบทความเซ็นเซอร์ที่จะถูกแบ่งออกเป็น

3 ประเภทคือ 1) Active Sensors เป็นเซ็นเซอร์ที่การปล่อยสัญญาณออกไปจากตัวเซ็นเซอร์เอง เช่น Radar และ Laser หรือ Lidar 2) Passive Sensors เป็นเซ็นเซอร์ที่ไม่มีการส่งสัญญาณออกไปจากตัวเซ็นเซอร์ เช่น Acoustic และ กล้อง 3) Fusion of Sensors เป็นการนำเซ็นเซอร์หลาย ๆ ชนิดมาทำงานร่วมกัน โดยบทความเรื่องนี้มีการวิเคราะห์ข้อดี และข้อเสียของเซ็นเซอร์ชนิดต่าง ๆ ดังแสดงในตารางที่ 2

ตารางที่ 2 ตัวอย่าง คุณสมบัติ ข้อดี และข้อเสีย ของเซ็นเซอร์ชนิดต่าง ๆ [7]

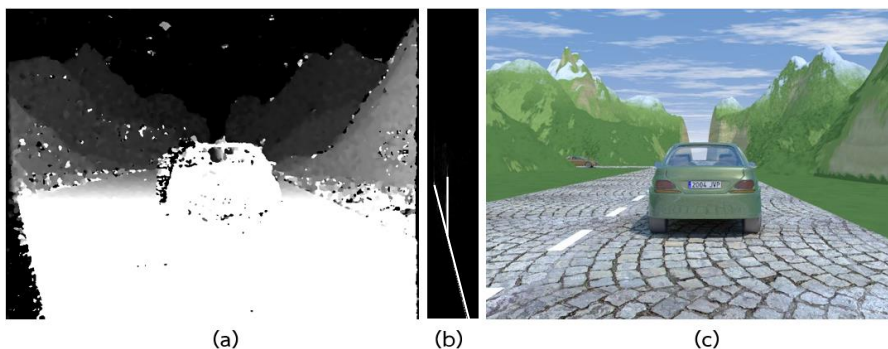
Sensor Type	Specific Sensor	Distance	Cost	Advantages	Disadvantages
Acoustic	SONY ECM-77B	Depends on sound waves amplitude and mic sensitivity	≈ 350 USD	<ul style="list-style-type: none"> • Omni-directional microphone • An economical solution • Real time 	<ul style="list-style-type: none"> • Interference problem • Noise sensitive • Short range
Radar	Delphi Adaptive Cruise Control	175 m	2,000 USD	<ul style="list-style-type: none"> • Measure distance directly with less computing resources • Longer detection range than acoustic and optical sensor • Robust in foggy or rainy day, and during night time. 	<ul style="list-style-type: none"> • Interference problem • Higher cost than Acoustic • Classification issue • More Power consumption than acoustic and optical sensor
Laser/Lidar	Velodyne HDL-64E Laser Rangefinder (3D LIDAR)	120 m	75,000 USD	<ul style="list-style-type: none"> • Longer detection range than acoustic and optical sensor • Independent of weather conditions • Modern lidar/laser scanners acquire high resolution and 3D information 	<ul style="list-style-type: none"> • Road infrastructure dependency • More Power consumption than other sensors • High speed 3D scanners are expensive
	SICK LMS511-10100 (2D)	80 m	7,000 USD		
Optical (camera)	SV-625B	100m for day 12m for night (Depth of focus)	160 USD	<ul style="list-style-type: none"> • Low cost, easier to install and maintain • Higher resolution and wider view angle • Extensive information in images • Independent of any modifications to the road infrastructure. • Accumulate data in nonintrusive way 	<ul style="list-style-type: none"> • Image quality depends on lighting and weather conditions • Requires more computing resources to process the images
Fusion of Sensors	Not Applicable	Depends on sensors fused	Depends on sensors fused	<ul style="list-style-type: none"> • Increases system robustness and reliability • Broadens the sensing capabilities • Collect maximum information of surroundings 	<ul style="list-style-type: none"> • Separate algorithms for each sensor • Expensive

ซึ่งจากคุณสมบัติต่าง ๆ แล้ว Optical Sensors ถูกนำมาใช้ เนื่องจากมีราคาที่ไม่สูง ติดตั้งได้ง่าย สามารถให้ข้อมูลต่าง ๆ ได้มากกว่าระยะทาง เช่น รูปลักษณะของวัตถุ ทำให้การตรวจจับวัตถุที่ต้องการแม่นยำขึ้น และลดข้อผิดพลาดลง นอกจากนี้ยังมีอุปกรณ์เสริมที่สามารถทำให้คุณสมบัติของกล้องเปลี่ยนไปตามที่เราต้องการได้ แต่เนื่องจากกล้องเพียงตัวเดียวไม่สามารถหาระยะของวัตถุได้จึงได้ใช้กล้อง 2 ตัวเป็น Stereo Camera เพื่อทำให้สามารถหาระยะทางของวัตถุได้ด้วย

2.3 ระบบตรวจจับวัตถุ

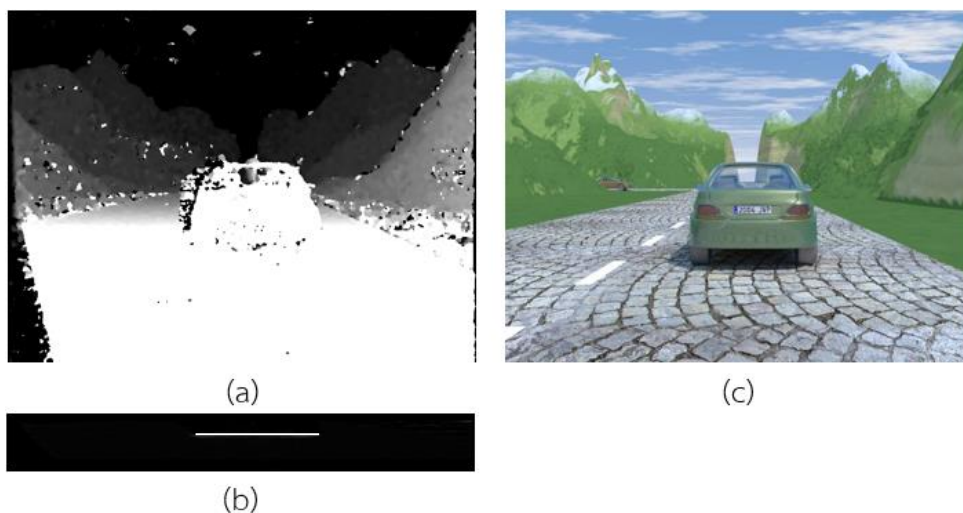
จากการใช้ Stereo Camera ทำให้ระบบสามารถรับรู้ลักษณะของสภาพแวดล้อมเป็น 3 มิติ บทความวิจัยเรื่อง “Real Time Obstacle Detection in Stereovision on Non Flat Road Geometry Through “V-disparity” Representation” [8] บทความวิจัยเรื่องนี้เป็นบทความวิจัยเรื่องวิธีการแยกวัตถุที่อยู่บนพื้นถนนออกจากระนาบของถนนโดยใช้ V-Disparity จากการประมวลผลภาพที่ได้จาก Stereo Vision Camera โดยการเปลี่ยนภาพ Disparity ซึ่งอยู่ในระนาบ U,V และมีค่า

ความลึกของภาพเป็น Disparity Value เป็นภาพ V-Disparity ซึ่งอยู่ในระนาบของ Disparity Value, V และมีค่าความลึกเป็นจำนวนพิกเซล โดยเมื่อมีวัตถุอยู่บนถนน ภาพ V-Disparity จะแสดงเส้นตรงตั้งฉากกับแกน Disparity Value มีความยาวเท่ากับความสูงของวัตถุ โดยที่วัตถุในภาพ Disparity และเส้นตรงในภาพ V-Disparity จะมีตำแหน่งในแกนตั้ง (แกน v) ของภาพ ที่ตำแหน่งเดียวกัน



รูปที่ 5 ภาพเปรียบเทียบระหว่างภาพ (a) Disparity (b) V-Disparity จากภาพต้นฉบับ (c)

บทความวิจัยเรื่อง “U-V Disparity : An efficient algorithm for stereovision based scene analysis” [9] เป็นการพัฒนาการตรวจจับวัตถุที่อยู่เหนือผิวถนนด้วยภาพ V-Disparity โดยการสร้างภาพอีกภาพหนึ่งโดยใช้วิธีใกล้เคียงกับภาพ V-Disparity คือ ภาพ U-Disparity ซึ่งในระนาบ U-Disparity Value สำหรับภาพ U-Disparity จะช่วยให้เราสามารถแยกแยะตำแหน่งวัตถุได้ดีขึ้นเนื่องจาก U-Disparity จะได้เส้นที่แสดงการมีอยู่ของวัตถุเป็นเส้นตรงขนานกับแกน Disparity value ซึ่งตำแหน่งของวัตถุที่แสดงในภาพ U-Disparity จะมีตำแหน่งในแกน U ของภาพ U-Disparity ที่ตำแหน่งเดียวกับ แกน U ในภาพ Disparity



รูปที่ 6 ภาพเปรียบเทียบระหว่างภาพ (a) Disparity (b) U-Disparity จากภาพต้นฉบับ (c)

การใช้ภาพ U-V Disparity ในการตรวจจับวัตถุที่อยู่เหนือผิวถนนเป็นการหาวัตถุโดย ไม่สนใจ ชนิด หรือรูปร่าง ของวัตถุ ซึ่งเรียกวิธีการตรวจจับวัตถุได้ก่อนที่จะมีการยืนยันนี้ว่า Hypothesis Generation และหลังจากที่วัตถุถูกตรวจจับแล้ว ต้องผ่านการยืนยันชนิดของวัตถุอีกครั้งหนึ่งซึ่งเรียก วิธีการนี้ว่า Hypothesis Verification โดยวิธี Hypothesis Verification จะถูกแบ่งออกเป็นขั้นตอนย่อย ๆ ด้วยกันอีก 2 ขั้นตอน คือ 1. Features Extraction และ 2. Classification โดยแต่ละขั้นตอนจะมีวิธีการหลาย ๆ วิธีแตกต่างกันไป จากบทความเรื่อง “Vehicle Detection Techniques for Collision Avoidance Systems : A Review” [7] ได้ทำการรวบรวมวิธีการในขั้นตอนต่าง ๆ ไว้ดังนี้

ตารางที่ 3 ตัวอย่าง วิธีการต่าง ๆ ในขั้นตอน Features Extraction และ Classification

Features Extraction	Classification
Histogram of Oriented Gradient	Artificial Neural Networks
Gabor	Support Vector Machine
Principal Component Analysis	AdaBoost
Haar-Like	Mahalonobis distance
Haar-Wavelet	

โดยขั้นตอนการทำ Features Extraction จะเป็นการดึงลักษณะต่าง ๆ ของวัตถุที่เราต้องการยืนยันโดยใช้วิธีต่าง ๆ หลังจากนั้นนำข้อมูลลักษณะที่ได้มาทำการ Classification ในตัวอย่างที่เราต้องการตรวจสอบ โดยก่อนจะทำการ Classification ได้นั้นตัวโปรแกรมต้องผ่านการสอนด้วยภาพตัวอย่างของวัตถุที่ต้องการยืนยันด้วยวิธีต่าง ๆ

วิธีการในส่วนของ Features Extraction ที่มีการนำมาใช้ในการยืนยันวัตถุที่เป็น รถยนต์ รถจักรยานยนต์ และ คนเดินถนน คือ Haar-Wavelet Haar-like และ Gabor วิธี Haar-like มีความสะดวกในการทำงานมากที่สุด และสามารถทำการสอนตัวโปรแกรมได้ง่ายมากที่สุด ด้วยวิธีการ AdaBoost ซึ่งเป็นการลดปัญหาของการทำงานในส่วนของ Images processing ลงไปได้ ซึ่งจากบทความเรื่อง “An Improved Algorithm Based on Ada Boost for Vehicle Recognition” [10] ได้ทำการทำสอบวิธีต่างด้วยโปรแกรม Matlab 7.1 ด้วยภาพทั้งหมด 6040 ภาพซึ่งแบ่งเป็น ภาพยานพาหนะ 4266 ภาพ และภาพที่ไม่ใช่ยานพาหนะ 1774 ภาพ ผลที่ได้ออกมา วิธีการ Haar-Like + AdaBoost ให้ผลการตรวจจับที่ 97.09% ซึ่งเป็นอันดับ 2 รองมาจากวิธีการที่เสนอโดยบทความเรื่องนี้ซึ่งได้ผลที่ 97.43% ซึ่งจุดประสงค์ของบทความนี้คือเพิ่มความสามารถในส่วนของ AdaBoost ทำให้การสอนโปรแกรมทำได้ไวขึ้น ซึ่ง 97.09% ถือเป็นเปอร์เซ็นต์ที่สูง รวมถึงความ

สะดวกในการนำมาใช้งานวิธี Haar-like และ AdaBoost จึงถูกนำมาใช้ในการยืนยันวัตถุในวิทยานิพนธ์นี้

2.4 ปัญหาของงานวิจัยที่ผ่านมา

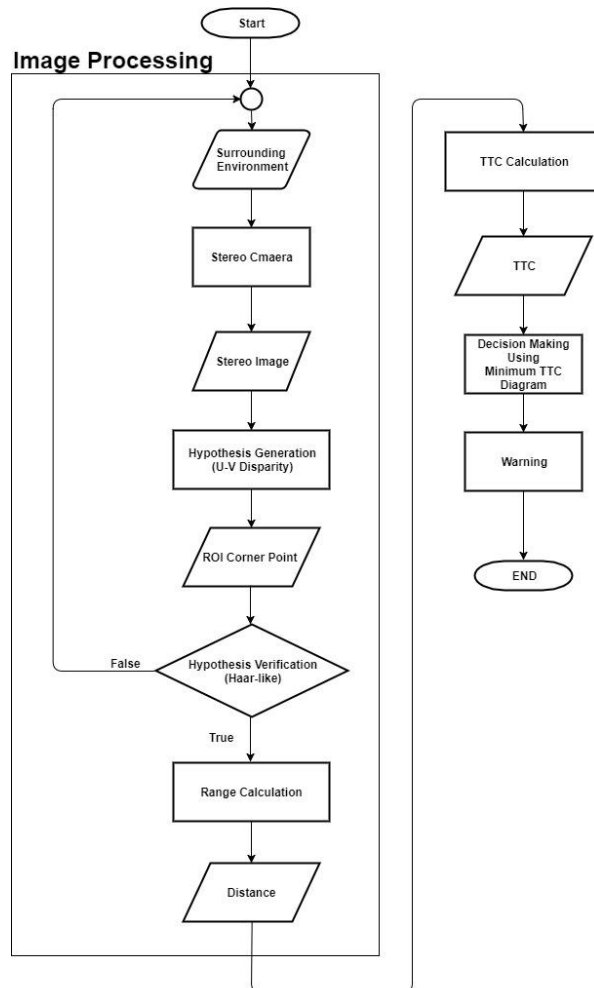
ในงานวิจัยที่ได้ศึกษาการคำนวณเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่มีการเลี้ยวเพื่อหลีกเลี่ยงการเกิดอุบัติเหตุมีการคำนวณในลักษณะของการใช้เรขาคณิตโดยการสมมุติเส้นทางการเคลื่อนที่ของรถยนต์เป็นรูปร่างทางเรขาคณิตโดยไม่มีการนำ Dynamic Model ของรถยนต์เข้ามามีในการคำนวณเส้นทาง โดยการใช้เรขาคณิตสมมุติเส้นทางการเคลื่อนที่ของรถยนต์นั้นง่ายที่จะควบคุม และทำตามด้วยคอนโทรลเลอร์ แต่เนื่องจากงานวิจัยนี้มีจุดประสงค์ที่จะพัฒนาระบบเตือนการชนด้านหน้าเนื่องจากสามารถนำมาติดตั้งรถยนต์ทั่วไปตามท้องถนนได้ ในขณะเดียวกัน ระบบที่มีลักษณะเป็นการหลีกเลี่ยงการชนโดยอัตโนมัติต้องการรถยนต์ที่มีระบบควบคุม เบรกหรือระบบควบคุม หรือช่วยเลี้ยวติดตั้งอยู่แล้วจึงจะสามารถนำมาติดตั้งเพื่อใช้งานได้ นอกจากระบบเตือนการชนด้านหน้าจะไม่ต้องการระบบที่ถูกติดตั้งอยู่ในรถยนต์แล้ว ตัวระบบนี้เองยังสามารถติดตั้งกับรถยนต์ได้ง่ายสืบเนื่องมาจากสาเหตุเดียวกันก็คือ ระบบไม่ต้องการระบบอื่นที่ติดตั้งกับรถยนต์มาทำงานร่วมกัน ดังนั้นการหลีกเลี่ยงการชนจึงจะถูกควบคุมด้วยผู้ขับขี่เท่านั้น ซึ่ง Geometrical Approach ทำให้การควบคุมรถยนต์ตามเส้นทางที่ได้จากวิธีนี้โดยผู้ขับขี่เป็นไปได้ยาก ดังนั้นในงานวิจัยนี้จึงนำ Dynamic Model มาใช้ในการคำนวณหาเส้นทางการหลีกเลี่ยงการชน

บทที่ 3

การออกแบบระบบเตือนการชนท้าย และการทดสอบ

3.1 การออกแบบระบบตรวจจับรถยนต์ด้านหน้า

อย่างที่ได้อธิบายไปในบทที่ 2 งานวิจัยนี้ใช้กล้องสเตอริโอ เป็นอุปกรณ์ตรวจจับรถยนต์ด้านหน้า โดยภาพที่ได้จากกล้องสเตอริโอ จะถูกนำมาประมวลผลด้วยวิธีการที่เรียกว่า Image Processing เพื่อเปลี่ยนข้อมูลจากรูปภาพเป็นข้อมูลที่ต้องการ เพื่อนำไปใช้ในขั้นตอนถัดไป ซึ่งกระบวนการประมวลผลภาพมีขั้นตอนดังที่แสดงใน รูปที่ 7



รูปที่ 7 แผนผังของระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรก ร่วมกับการเลี้ยวหลบ

การทำ Image Processing ถูกแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ Hypothesis Generation (HG) และ Hypothesis Verification (HV) ซึ่งกระบวนการ Hypothesis Generation เป็นกระบวนการสร้างตำแหน่งที่สนใจในภาพ (Region of Interest : ROI) จากการตั้งสมมุติฐานว่า

ตำแหน่งนั้นมีวัตถุที่ต้องการตรวจจับอยู่ โดยวิธีที่ใช้ในงานวิจัยนี้คือการใช้แผนภาพ U-V Disparity ส่วนกระบวนการ Hypothesis Verification เป็นการนำ ROI มาทำการยืนยันว่าใน ROI นั้นมีวัตถุที่เราต้องการอยู่ โดย HAAR-Like Feature ถูกนำมาใช้ในงานวิจัยนี้

3.1.1 กล้องสเตอริโอวิชัน

กล้องที่ใช้ในการตรวจจับรถยนต์ในงานวิจัยนี้คือ กล้องสเตอริโอวิชัน ซึ่งมีข้อดี คือ สามารถตรวจจับลักษณะของวัตถุ และระยะทางของวัตถุได้เนื่องจากการใช้กล้องจำนวน 2 ตัว ถ่ายภาพวัตถุเดียวกัน เพื่อสร้างภาพที่เป็น 3 มิติขึ้นมา โดยระยะลึกของภาพเกิดจากการที่กล้องทั้งสองตัวถูกติดตั้งโดยมีระยะห่างระหว่างกล้องสองตัว ดังแสดงใน รูปที่ 8 ซึ่งเป็นการจำลองการรับรู้ภาพของมนุษย์โดยที่ตาสองข้างมีระยะระหว่างตาทั้งคู่เช่นกัน



รูปที่ 8 การติดตั้งกล้อง 2 ตัว แบบ Stereo Camera

ซึ่งการวางตำแหน่งกล้องในลักษณะนี้ ทำให้ตำแหน่งของวัตถุในภาพที่ปรากฏจากกล้องแต่ละตัว อยู่ในตำแหน่งที่ต่างกัน โดยถ้ากำหนดคู่อันดับของเซนเซอร์ของกล้องเป็น (u,v) โดยที่ u เป็นคู่อันดับในแนวนอน และ v เป็นคู่อันดับในแนวตั้ง ของเซนเซอร์ ตำแหน่งที่ต่างกันจะเป็นตำแหน่งในคู่อันดับ u ซึ่งเรียกค่าที่แตกต่างกันนี้ว่า Disparity Value เป็นค่าที่แสดงให้เห็นถึงตำแหน่งที่ต่างกันของพิกเซลในกล้องทั้งสองตัว ว่าอยู่ต่างกันอยู่ที่พิกเซล ซึ่งสามารถนำค่าของ Disparity Value ในตำแหน่งต่าง ๆ มาสร้างเป็นภาพ Gray Scale ได้ เรียกว่า Disparity Map ซึ่งการที่จะได้ค่า Disparity Value มานั้น สิ่งที่ต้องกระทำเป็นอย่างแรกคือ การถ่ายภาพวัตถุที่อยู่ในโลกจริง ให้อยู่ในระนาบเซนเซอร์ของกล้องแต่ละตัว ทำได้โดยการแปลงตำแหน่งเวกเตอร์ของคู่อันดับในโลกจริง $(X,Y,Z,1)^T$ ให้อยู่ในคู่อันดับของระนาบเซนเซอร์ของกล้อง $(u,v,1)^T$ โดยอาศัย Transformation Matrix (M_{lr}) ดังแสดงในสมการที่ (1) โดยที่ l และ r แสดงถึงกล้องด้าน ซ้าย และ ขวา ตามลำดับ

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = M_{l,r} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = PD_{l,r} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

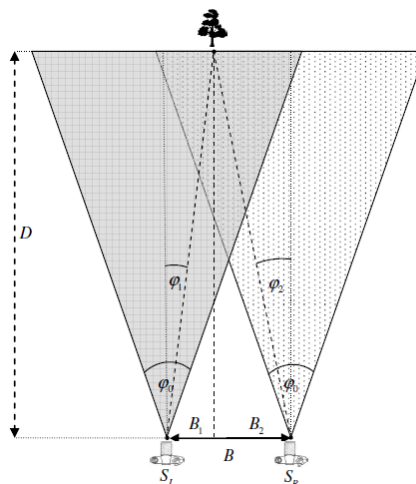
โดยที่ P เป็นเมทริกซ์ที่ประกอบไปด้วยตัวแปรต่าง ๆ ภายในกล่องดังแสดงในสมการที่ (2)

$$P = \begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2)$$

โดยที่ f คือความยาวโฟกัสของกล้อง และ (u_0, v_0) คือจุดศูนย์กลางของเซนเซอร์ และ D เป็นเมทริกซ์ที่ประกอบไปด้วยตัวแปรภายนอกดังแสดงในสมการที่ (3)

$$D_{l,r} = [R_{l,r} | T_{l,r}] = \begin{bmatrix} 1 & 0 & 0 & \pm \frac{b}{2} \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3)$$

โดยที่ R คือ Rotation Matrix T คือ Translation Matrix θ คือมุมที่เกิดขึ้นเมื่อกล้องมีการหมุนรอบแกน X และ b คือระยะห่างระหว่างกล้องทั้งสองตัว โดยตรงแปรทั้งหมดถูกแสดงอยู่ใน รูปที่ 9



รูปที่ 9 ตัวแปรต่าง ๆ ในการติดตั้งกล้อง 2 ตัว แบบ Stereo Camera [11]

จากสมการที่ (1) (2) และ (3) เราสามารถคำนวณหาตำแหน่งของวัตถุในคู่อันดับ (u,v) ได้ ดังแสดงในสมการที่ (4)

$$u_{l,r} = u_0 + f \frac{X \pm b/2}{Y \sin \theta + Z \cos \theta}$$

$$v = v_0 + f \frac{Y \cos \theta - Z \sin \theta}{Y \sin \theta + Z \cos \theta}$$
(4)

หลังจากนั้นค่า Disparity value (Δ) โดยที่ $\Delta = u_l - u_r$ จะถูกนำมาสร้างเป็นแผนภาพที่เรียกว่า Disparity Map ดังแสดงในรูปที่ 10 โดยแผนภาพจะมีขนาดเท่ากับภาพต้นฉบับและค่าความเข้มในพิกเซลมีค่าเท่ากับ Disparity Value นั้นตำแหน่งนั้น ๆ โดยตำแหน่งที่มีค่า Disparity สูง จะมีสีขาว ซึ่งแสดงให้เห็นว่าวัตถุมีตำแหน่งที่อยู่ใกล้กับกล้อง ในทางกลับกันตำแหน่งที่มีสีเข้ม เป็นตำแหน่งที่ค่า Disparity ต่ำ แสดงให้เห็นว่า วัตถุมีตำแหน่งที่อยู่ไกลจากกล้องออกไป



รูปที่ 10 แผนภาพ Disparity Map
CHULALONGKORN UNIVERSITY

3.1.2 แผนภาพ U Disparity และ V Disparity

หลังจากสร้างแผนภาพ Disparity Map เสร็จแล้ว แผนภาพ Disparity Map จะถูกนำมาสร้างแผนภาพอีก 2 แผนภาพชื่อว่า U Disparity และ V Disparity โดยแผนภาพ U Disparity และ V Disparity ในส่วนของการตรวจจบบรรณดที่เรียกว่า Hypothesis Generation ถ้ามองภาพเป็นเมทริกซ์ 2 มิติ (Gray Scale) โดยมีค่าในแต่ละ อิลิเมนต์ในเมทริกซ์เป็นค่าความลึกของภาพ V-Disparity Map จะเป็นเมทริกซ์ 2 มิติ ที่มีจำนวนแถวเท่ากับ จำนวนแถวของ Disparity Map และจำนวนหลักเท่ากับ ค่า Disparity Value สูงสุดของ Disparity Map และในทำนองเดียวกัน U-Disparity Map จะมีจำนวนหลักเท่ากับ จำนวนหลักของ Disparity Map และจำนวนแถวเท่ากับ

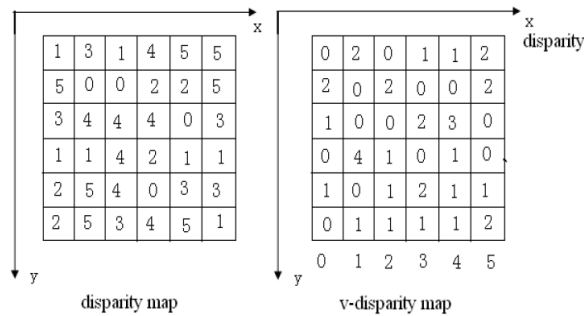
จำนวนหลักของ V-Disparity Map หรือ ค่า Disparity Value สูงสุดของ Disparity Map ซึ่งสามารถอธิบายโดยใช้สมการทางคณิตศาสตร์ได้ ดังแสดงในสมการที่ (5) และ (6) ตามลำดับ

$$v_{dj} = \sum_{i=0}^{cols} \Delta_{ij}, \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad (5)$$

เมื่อ

d = ค่า Disparity Value ของ Disparity Map ที่หลัก i และแถว j

v_{dj} = ค่าในอิลิเมนต์ของเมทริกซ์ V-Disparity ที่หลัก d และแถว j ซึ่งแสดงถึงความถี่ของ Disparity Value d ที่แถว j ใน Disparity Map



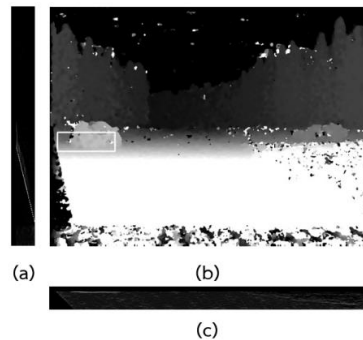
รูปที่ 11 ภาพการสร้างเมทริกซ์ของ V-Disparity Map จากเมทริกซ์ของ Disparity Map [12]

$$u_{id} = \sum_{j=0}^{rows} \Delta_{ij}, \Delta_{ij} = \begin{cases} 1, & disp_{ij} = d \\ 0, & otherwise \end{cases} \quad (6)$$

เมื่อ

d = ค่า Disparity Value ของ Disparity Map ที่หลัก i และแถว j

u_{id} = ค่าในอิลิเมนต์ของเมทริกซ์ U-Disparity ที่หลัก i และแถว d ซึ่งแสดงถึงความถี่ของ Disparity Value d ที่หลัก i ใน Disparity Map



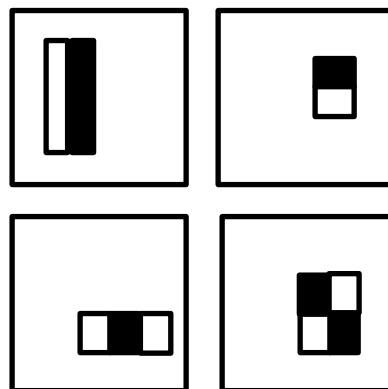
รูปที่ 12 (a) V-Disparity Map (b) Disparity Map และ (c) U-Disparity Map

โดยตำแหน่ง ROI จะถูกคำนวณมาจากตำแหน่งของเส้นตรงแนวตั้งในแผนภาพ V-Disparity และตำแหน่งของเส้นตรงแนวนอนในแผนภาพ U-Disparity

3.1.3 Haar-like feature

Haar-like feature เป็นวิธีการทาง Digital image ที่ถูกใช้ในวัตถุประสงค์เพื่อ ตรวจสอบวัตถุที่ต้องการ โดยเป็นวิธีที่ถูกพัฒนามาจาก Haar wavelet จึงมีชื่อเรียกว่า Haar-like features โดยเริ่มแรกถูกใช้ในการตรวจจับใบหน้าบุคคล โดยงานวิจัยนี้จะนำ Haar-like Feature มาใช้ในกระบวนการ Hypothesis Verification ในการตรวจจับรถยนต์

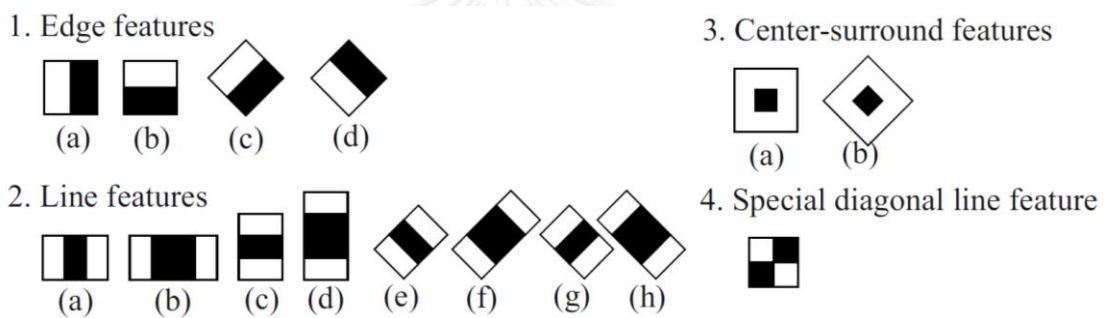
วิธีการ Haar-like feature จะพิจารณาบริเวณสี่เหลี่ยมที่อยู่ติดกัน ดังรูปที่ 13 ณ ตำแหน่งใด ๆ ของหน้าต่างที่ใช้ในการตรวจจับ หลังจากนั้น ทำการรวมค่าความลึกของพิกเซลในแต่ละบริเวณ และทำการคำนวณค่าความต่างกันของค่าในแต่ละบริเวณโดยให้ค่าในบริเวณสีขาวเป็นบวก และบริเวณสีดำเป็นลบ เมื่อได้ค่าความต่างกันมาแล้ว ค่านี้จะถูกใช้ในการแบ่งคุณลักษณะของบริเวณย่อย ๆ ต่าง ๆ ในวัตถุ เพื่อนำคุณลักษณะนี้ไปพิจารณาในการตรวจหาวัตถุที่ต้องการ



รูปที่ 13 ตัวอย่างของสี่เหลี่ยมลักษณะต่าง ๆ ในหน้าต่างที่ใช้ในการตรวจจับ

ในขั้นตอนการตรวจจับ หน้าต่างที่แสดงในรูปที่ 13 จะเคลื่อนที่ไปรอบ ๆ รูปที่ต้องการตรวจหาวัตถุ และคุณลักษณะของบริเวณย่อย ๆ ที่หน้าต่างเคลื่อนที่ไปจะถูกคำนวณ หลังจากนั้นค่าความต่างที่ได้จะถูกนำไปเปรียบเทียบกับขอบเขตที่ได้จากการเรียนรู้ของโปรแกรมจากภาพตัวอย่าง เพื่อแยกบริเวณที่เป็นวัตถุที่ต้องการ และไม่ใช่วัตถุที่ต้องการออกจากกัน เนื่องจากวิธีการนี้เป็นการเรียนรู้จากตัวอย่างแบบอ่อน (Weak Sample) ซึ่งเป็นตัวอย่างที่ไม่ได้บรรจุข้อมูลจำเป็นสำหรับการตรวจจับวัตถุชนิดใดชนิดหนึ่งไว้ 100% เช่น การตรวจจับใบหน้า เนื่องจากใบหน้าของคนแต่ละคนมีลักษณะแตกต่างกัน ดังนั้นการที่จะตรวจจับใบหน้าโดยทั่วไปจึงไม่สามารถนำภาพใบหน้าตัวอย่างเพียงตัวอย่างเดียวใช้ในการสอนได้ จากสาเหตุนี้ การเรียนรู้ของโปรแกรมจากตัวอย่างแบบอ่อน ต้องมีการหาตัวอย่างที่เป็นวัตถุชนิดเดียวกัน แต่มีลักษณะปลีกย่อยต่างกันไป จำนวนมากมาใช้ในการสอนโปรแกรม

หน้าต่างที่ใช้ในการตรวจจับสามารถแบ่งออกได้เป็นหลายลักษณะดังนี้



รูปที่ 14 หน้าต่างที่ใช้ในการตรวจจับในรูปแบบต่าง ๆ [13]

หน้าต่างในรูปที่ 14 แบบ 1c 1d 2e 2f 2g 2h และ 3b ถูกพัฒนาขึ้นมาทีหลังเพื่อเพิ่มมิติในการตรวจจับ และเพิ่มประสิทธิภาพในการตรวจจับวัตถุ แต่ข้อเสียคือทำให้เวลาในการคำนวณเพิ่มมากขึ้นอีกด้วย



รูปที่ 15 การนำ Haar-like Feature มาใช้ในการตรวจจับรถยนต์

3.1.4 กระบวนการ Machine Learning แบบ AdaBoost

การใช้ Haar-like Feature นั้นจะต้องผ่านกระบวนการที่เรียกว่า Machine Learning เพื่อสอนให้ระบบทราบถึงตัวลักษณะของวัตถุที่เราจะต้องการตรวจจับซึ่งในที่นี้คือ รถยนต์ หลังจากการทำ Machine Learning นั้นจะได้ข้อมูลอ้างอิงเพื่อใช้ในการตรวจจับรถยนต์ได้ โดยวิธีที่ใช้ในกระบวนการ Machine Learning คือ AdaBoost

AdaBoost เป็นวิธีการ Machine Learning โดยการนำชุดข้อมูลที่ต้องการฝึก $(x_1, y_1), \dots, (x_m, y_m)$ เมื่อ x_i คือตัวอย่างของวัตถุ และ y_i เป็นคลาสของวัตถุที่ต้องการจำแนก โดยสมมติให้คลาสมีทั้งหมด 2 คลาส คือ 1 และ -1 เพื่อความสะดวก โดยหลักการของอัลกอริทึมนี้ คือการกระจายน้ำหนักไปทั่วชุดข้อมูล โดยการวิธีการทำซ้ำที่เรียกว่า weak learner โดยทำซ้ำเป็นขั้นจำนวน T ชั้น โดยทำการทำซ้ำเพื่อลดความคาดเคลื่อนของชุดข้อมูลลง วิธีการนี้เรียกว่า Boost

AdaBoost มาจากคำว่า Adaptive Boost โดยเป็นการเปลี่ยนแปลงน้ำหนัก (น้ำหนักของตัวอย่างลำดับที่ i คือ $D_t(i)$) ของแต่ละข้อมูลในชุดข้อมูลจนทำให้ความคาดเคลื่อนของชุดข้อมูลลดลง เช่น ตอนเริ่มต้นอัลกอริทึมให้น้ำหนักของข้อมูลแต่ละข้อมูลมีค่าเท่ากัน แต่เมื่อมีการฝึกไปที่ละขั้น ข้อมูลที่ถูกจัดคลาสไม่ถูกต้องจะถูกเพิ่มน้ำหนักขึ้น เมื่อการฝึกเข้าสู่ขั้นต่อไปข้อมูลที่มีน้ำหนักมากจะถูกโฟกัสมากกว่าข้อมูลข้อมูลอื่น โดยหน้าที่ของ weak learner คือหา Classifier $h_t = X \rightarrow \{-1, 1\}$ ที่ทำให้เกิดความคาดเคลื่อน (ϵ_t) น้อยที่สุดเนื่องจากการกระจายของน้ำหนัก D_t เมื่อ

$$\varepsilon_t = \sum_{i=1}^l \text{weight}(h_t(i)) \begin{cases} \text{if}(h_t(i) \neq y_i) \\ \text{else}(0) \end{cases} \quad (7)$$

โดยที่ทุก h_t จะมีน้ำหนักซึ่งแปรผกผันกับ ε_t โดยที่ถ้า ε_t ลดลง น้ำหนัก α_t จะเพิ่มขึ้น และจะมีค่าสุดท้ายดังแสดงในสมการที่ (8)

$$\alpha_t = \ln\left(\frac{1 - \varepsilon_t}{\varepsilon_t}\right) \quad (8)$$

และในขั้นตอนสุดท้ายของอัลกอริทึมทุก Classifier h_t จะมีน้ำหนักเท่ากับ α_t และหลังจากนั้นข้อมูลที่ฝึกไว้จะถูกเก็บไว้เพื่ออ้างอิงในขั้นตอนการตรวจจับวัตถุต่อไป ซึ่งเป็นข้อมูล (X) ที่ถูกจัดคลาสหลังจากถ่วงน้ำหนัก α_t ค่าสุดท้ายแล้ว จะถูกเรียกว่า Strong classifier ($H(x)$)

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right) \quad (9)$$

Algorithm AdaBoost

Input: sequence of N labeled examples $((x_1, y_1), \dots, (x_N, y_N))$
distribution D over the N examples
weak learning algorithm **WeakLearn**
integer T specifying number of iterations

Initialize the weight vector: $w_i^1 = D(i)$ for $i = 1, \dots, N$.

Do for $t = 1, 2, \dots, T$

1. Set

$$\mathbf{p}^t = \frac{\mathbf{w}^t}{\sum_{i=1}^N w_i^t}$$

2. Call **WeakLearn**, providing it with the distribution \mathbf{p}^t ; get back a hypothesis $h_t : X \rightarrow [0, 1]$.

3. Calculate the error of h_t : $\varepsilon_t = \sum_{i=1}^N p_i^t |h_t(x_i) - y_i|$.

4. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$.

5. Set the new weights vector to be

$$w_i^{t+1} = w_i^t \beta_t^{1 - |h_t(x_i) - y_i|}$$

Output the hypothesis

$$h_f(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^T \left(\log \frac{1}{\beta_t}\right) h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \log \frac{1}{\beta_t} \\ 0 & \text{otherwise} \end{cases}$$

รูปที่ 16 อัลกอริทึมของการเรียนรู้โดย AdaBoost [14]

โดยการทำให้ Machine Learning จำเป็นที่จะต้องใช้องค์ข้อมูลตัวอย่างจำนวนมากในกระบวนการนี้ โดยข้อมูลที่ใช้มี 2 ประเภทคือ

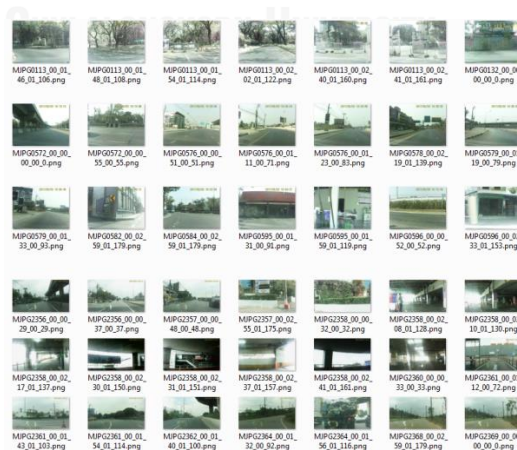
1. Positive Sample [15],[16]คือ ข้อมูลรูปภาพที่ประกอบไปด้วยรูปภาพวัตถุที่ต้องการตรวจจับ ในที่นี้คือ ภาพท้ายรถยนต์ ดังแสดงในรูปที่ 17

2. Negative Sample [17] คือ ข้อมูลรูปภาพที่ไม่มีภาพของ Positive Sample อยู่ในข้อมูลชุดนั้นเลย ดังแสดงในรูปที่ 18

เมื่อได้ข้อมูลตัวอย่างแล้ว ข้อมูลทั้งหมดจะถูกทำ Machine Learning ด้วย คำสั่งในโปรแกรม Matlab โดยจะได้ข้อมูลอ้างอิงเพื่อใช้ในการตรวจจับรถยนต์ดังแสดงในรูปที่ 19



รูปที่ 17 Positive Samples



รูปที่ 18 Negative Samples

```

<?xml version="1.0"?>
<opencv_storage>
  <!-- Created using Computer Vision System Toolbox(tm) for MATLAB(R) -->
  <!-- Version 8.5.0.197613 (R2015a) -->
  <!-- Compatible with OpenCV 2.4 -->
  - <cascade>
    <stageType>BOOST</stageType>
    <featureType>HAAR</featureType>
    <height>32</height>
    <width>39</width>
    - <stageParams>
      <boostType>GAB</boostType>
      <minHitRate>9.9500000476837158e-01</minHitRate>
      <maxFalseAlarm>5.00000000000000e-01</maxFalseAlarm>
      <weightTrimRate>9.49999999999999e-01</weightTrimRate>
      <maxDepth>1</maxDepth>
      <maxWeakCount>100</maxWeakCount>
    </stageParams>
    - <featureParams>
      <maxCatCount>0</maxCatCount>
      <featSize>1</featSize>
      <mode>BASIC</mode>
    </featureParams>
    <stageNum>13</stageNum>
  - <stages>
    - <!-- stage 0 -->
    - <-->
      <maxWeakCount>6</maxWeakCount>
      <stageThreshold>-1.0587797164916992e+00</stageThreshold>
    - <weakClassifiers>
      - <-->
        <internalNodes>0 -1 61 -2.1516026929020882e-02</internalNodes>
        <leafValues>7.6892429590225220e-01 -5.1914036273956299e-01</leafValues>
      </-->
    </-->
  </stages>
</opencv_storage>

```

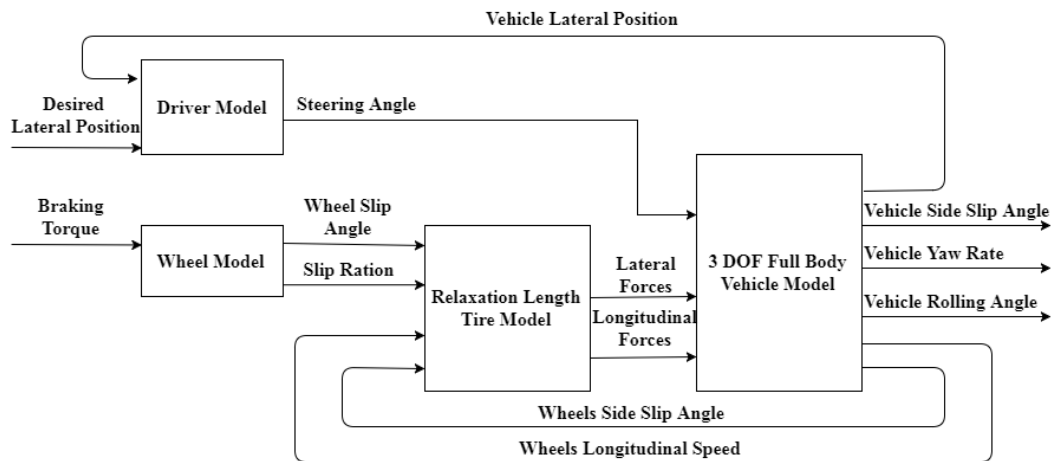
รูปที่ 19 ข้อมูลอ้างอิงที่ใช้ในการตรวจจ็บบรอยนต์หลังจากการทำ Machine Learning

3.2 Model Simulation

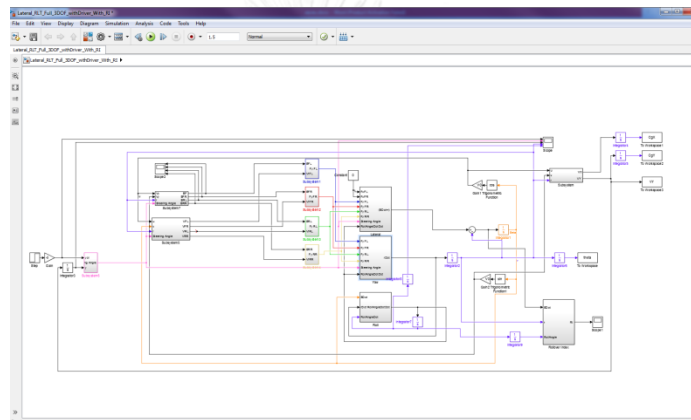
เพื่อให้เส้นทางการเคลื่อนที่ของรถยนต์ มีความใกล้เคียงกับเส้นทางที่จะเกิดขึ้นจริงเมื่อผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมากที่สุด หลักการพลศาสตร์ยานยนต์จึงถูกนำมาใช้ในรูปแบบของแบบจำลองทางพลศาสตร์ยานยนต์ โดยแบบจำลองที่ใช้นั้นมีทั้งสิ้น 4 แบบจำลอง ดังแสดงในรูปที่ 20 คือ

1. 3 Degree of Freedom Full Body Vehicle Dynamics Model
2. Relaxation Length Tire Model
3. Wheel Model
4. Driver Model

โดยการสร้างแบบจำลอง และคำนวณเส้นทางการเคลื่อนที่ของรถยนต์ถูกทำขึ้นมาด้วยโปรแกรม Matlab Simulink ดังแสดงในรูปที่ 21



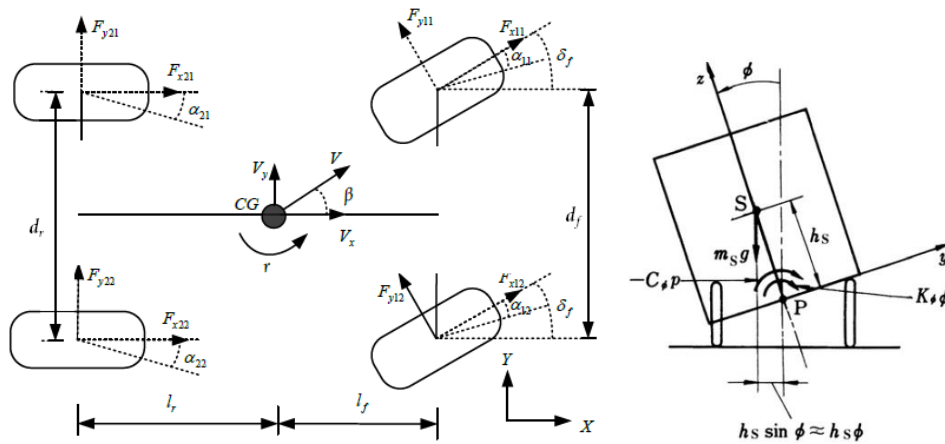
รูปที่ 20 แผนภาพ แบบจำลองทางพลศาสตร์ยานยนต์



รูปที่ 21 การจำลอง Vehicle Dynamics Model ผ่านโปรแกรม Matlab Simulink

3.2.1 3 Degree of Freedom Full Body Vehicle Dynamics Model

3 DoF Full Body Vehicle Dynamics Model ถูกนำมาใช้ในงานวิจัยชิ้นนี้ โดย 3 Degree of Freedom (3DoF) ประกอบไปด้วย Vehicle Side Slip Angle, Yaw Rate และ Rolling Angle ซึ่งสามารถนำมาคำนวณหาค่า Roll Over Index ที่แสดงถึงโอกาสในการพลิกคว่ำของรถยนต์ได้ในขณะที่ผู้ขับขี่ควบคุมรถยนต์ด้วยการเลี้ยวหลบ นอกจากนี้รถยนต์จริงยังเกิดการถ่วงน้ำหนักเนื่องจากการ Roll ของรถยนต์ซึ่งต้องใช้ Full Body Model ในการจำลองผล และเพียงพอที่จะใช้ในการหาเส้นทางการเคลื่อนที่ของรถยนต์ด้วยเช่นกัน ซึ่ง 3 DoF Full Body Vehicle Dynamics Model สามารถคำนึงถึงผลนั้นได้ต่างจาก Bicycle Model



รูปที่ 22 3 Degree of Freedom Full Body Vehicle Model [18]

3 Dof Full Body Vehicle Dynamics Model (รูปที่ 22) สามารถเขียนแสดงได้ด้วยสมการทั้งสี่ 3 สมการดังแสดงในสมการที่ (10) (11) และ (12)

$$(F_{x11} + F_{x12}) \sin \delta + (F_{y11} + F_{y12}) \cos \delta + F_{y21} + F_{y22} + \ddot{\phi} m_s h_s = mV(\dot{\beta} + r) \quad (10)$$

$$\left(F_{x11} \frac{D_f}{2} - F_{x12} \frac{D_f}{2} \right) \sin \delta + (F_{y11} L_f + F_{y12} L_r) \cos \delta - F_{y21} - F_{y22} + \dot{\phi} I_{xz} = r I_z \quad (11)$$

$$(\dot{\beta} + r) m_s h_s V - \dot{\phi} C_{roll} + I_{xz} \dot{r} + \phi (-K_{roll} + (m_s h_s g)) = \ddot{\phi} I_{roll} \quad (12)$$

โดยที่ F_x และ F_y คือ แรงในแนว Longitudinal และ Lateral ของยางตามลำดับ. V คือ ความเร็วของรถยนต์ r และ ϕ คือ yaw rate และ มุม roll ของรถยนต์ δ_f คือ มุมเอียงของรถยนต์. I_z และ m คือ โมเมนต์ความเฉื่อยในแนวแกน z และมวลของรถยนต์ I_{xz} และ I_{roll} คือ โมเมนต์ความเฉื่อยในระนาบ xz และในแนวการ roll ของรถยนต์ K_{roll} คือ C_{roll} Rolling Stiffness และ Rolling Damping ของรถยนต์ m_s และ h_s คือ มวล และความสูงของ Un-sprung Mass ตามลำดับ โดย แรงในแนวแกน Longitudinal และ Lateral สามารถหาได้จาก Relaxation Length Model ของยาง

3.2.2 Relaxation Length Tire Model

Relaxation Length Tire Model เป็น Tire Model ที่มีจำลองการหน่วงซึ่งมีสาเหตุจากการเสียดรูปของหน้าสัมผัสของยางกับพื้นถนน ที่เป็นผลมาจากการ เบรก และการเอียงของรถยนต์ โดย Relaxation Length Tire Model ที่ถูกนำมาใช้มีทั้งสิ้น 2 ชนิดตามทิศทางของแรงที่เกิดขึ้นจาก

ยาง คือ Longitudinal Relaxation Length Tire Model [19] และ Longitudinal Relaxation Length Tire Model [20] เนื่องจากการเปลี่ยนแปดการเคลื่อนที่ Relaxation Length Tire Model จะแสดงให้เห็นถึงผลของความหน่วงเนื่องจากคุณสมบัติของยางให้กับระบบได้

Longitudinal Relaxation Length Tire Model สามารถเขียนแสดงได้ดังสมการที่ (13)

$$F_{longitudinal} = C_s \frac{v_w}{s + v_w} \lambda \quad (13)$$

เมื่อ C_s คือ ค่า Longitudinal stiffness ของยาง ซึ่งเป็นฟังก์ชันของ Relaxation Length ของยาง ($C_s = f(\sigma)$), เมื่อ σ คือ Relaxation Length ของยาง v_w และ λ คือ ความเร็วในแนว Longitudinal และ Slip Ratio ของล้อ ตามลำดับ

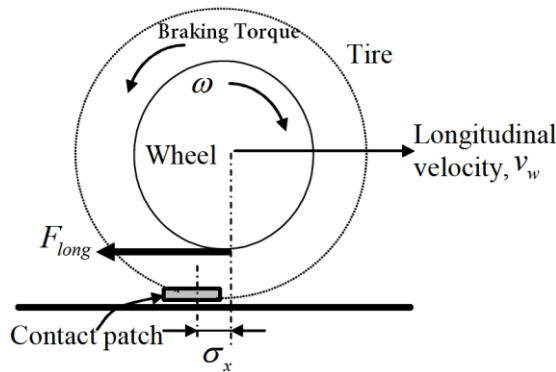
Lateral Relaxation Length Tire Model สามารถเขียนแสดงได้ดังสมการที่ (14)

$$F_{lateral} = C_\alpha \frac{v_r / \sigma}{s + (v_r / \sigma)} \alpha \quad (14)$$

เมื่อ v_r , C_α และ α คือ Rolling Speed ของล้อ, Cornering Stiffness ของยาง และ Side Slip Angle ตามลำดับ

3.2.3 Wheel Model

โมเดลของล้อรถยนต์ถูกนำมาใช้ด้วยเหตุผล 2 ข้อ คือ เพื่อจำลองพฤติกรรมของการไถลของล้อ และเพื่อคำนวณหาตัวแปรที่ต้องการเพื่อนำไปใช้ในโมเดลอื่น ๆ ซึ่งก็คือ Wheel Slip Ratio ในขณะที่รถยนต์กำลังเบรคนั้น การเสียดสีของหน้าสัมผัสระหว่างยางรถยนต์ และพื้นถนน สามารถถูกอธิบายได้ดังแสดงในรูปที่ 23 โดยที่การเสียดสีของยางนั้นเป็นสาเหตุทำให้เกิดความแตกต่างระหว่าง Rolling Speed ของล้อ และความเร็วในแนวแกน Longitudinal ของล้อ ซึ่งส่งผลให้เกิดพฤติกรรมการไถลขึ้น ซึ่ง Rolling Speed และ Slip Ratio ของล้อ สามารถเขียนอธิบายได้ดังสมการที่ (15) และ (16)



รูปที่ 23 กลศาสตร์ของยางขณะทำการเบรก [19]

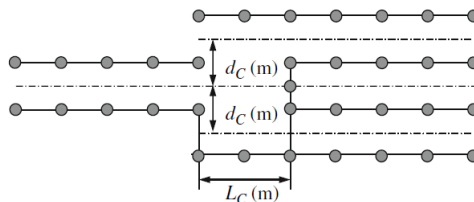
$$I_w \alpha = F_{longitudinal} R_w - \tau_{brake} \tag{15}$$

$$\lambda = \frac{(v_w - \omega R_w)}{v_w} \tag{16}$$

เมื่อ I_w , α , ω , R_w และ τ_{brake} คือ โมเมนต์ความเฉื่อย, ความเร่งเชิงมุม, ความเร็วเชิงมุม, รัศมี, และ แรงบิดการเบรก ของล้อตามลำดับ

3.2.4 Driver Model

Driver Mode (Human Controller [18]) เป็นแบบจำลองที่จำลองพฤติกรรมเปลี่ยนช่องทางจราจรของผู้ขับขี่ ซึ่งมีสถานการณ์ใกล้เคียงกับหลีกเลี่ยงการชนด้วยการเลี้ยวหลบ โดยที่ผู้ขับขี่ต้องการเปลี่ยนช่องทางจราจรในสถานที่ทดสอบแสดงในรูปที่ 24 ซึ่งมีสิ่งกีดขวางตั้งอยู่ที่ตำแหน่งด้านหน้ารถยนต์ โดยผู้ขับขี่ถูกกำหนดให้ทำการเปลี่ยนช่องทางจราจรเมื่อ Time Headway มีค่าเท่ากับ 1.5 วินาที หลังจากนั้นตำแหน่งของรถยนต์ถูกบันทึก วิเคราะห์ และทำการจำลองเป็นแบบจำลองผู้ขับขี่



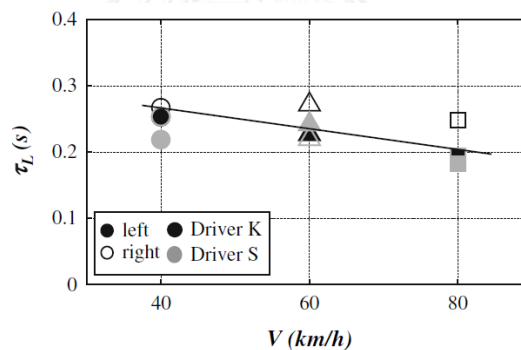
รูปที่ 24 ภาพจำลองสถานที่ทดสอบของ แบบจำลองผู้ขับขี่ [18]

โดยฟังก์ชันถ่ายโอนของแบบจำลองผู้ขับขี่ แสดงผ่านสมการที่ (17)

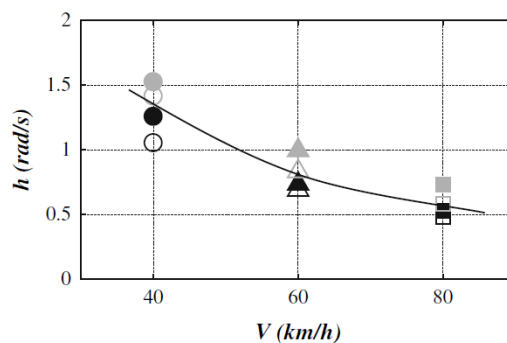
$$\delta(s) = -\frac{h}{1 + \tau_L s} \left\{ (1 + \tau_h s) y(s) - Y_{OL}(s) \right\} \quad (17)$$

เมื่อ y และ y_{OL} คือ ตำแหน่งในแนวแกน y ของรถยนต์ และตำแหน่งสุดท้ายในแนวแกน y ที่ต้องการ τ_L และ τ_h คือ ความล่าช้าที่เกิดจากการตอบสนองของผู้ขับขี่ และความล่าช้าที่เกิดจากพฤติกรรมของการมองทางด้านหน้าของผู้ขับขี่ ตามลำดับ h คือ Proportional Gain ในการเลี้ยว ซึ่งแสดงถึงความเร็วในการหมุนพวงมาลัยของผู้ขับขี่ขณะที่ผู้ขับขี่ทำการเปลี่ยนช่องทางจราจร

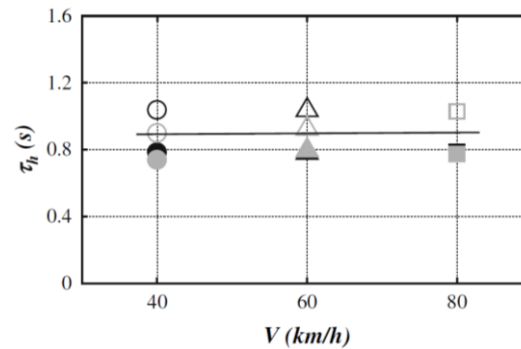
มีตัวแปร 2 ตัวในแบบจำลองผู้ขับขี่แปรผันตรงกับความเร็วของรถยนต์ คือ ความล่าช้าที่เกิดจากการตอบสนองของผู้ขับขี่ (τ_L) และความล่าช้าที่เกิดจากพฤติกรรมของการมองทางด้านหน้าของผู้ขับขี่ (τ_h) โดยค่าตัวแปรที่กล่าวมาถูกแสดงอยู่ในกราฟดังแสดงในรูปที่ 25 และ 26 แต่ตัวแปรความล่าช้าที่เกิดจากพฤติกรรมของการมองทางด้านหน้าของผู้ขับขี่ (τ_h) มีแนวโน้มที่จะคงที่ในทุกความเร็ว โดยมีค่าอยู่ระหว่าง 0.8-1 วินาที ดังแสดงในรูปที่ 27



รูปที่ 25 ความสัมพันธ์ระหว่างความล่าช้าที่เกิดจากการตอบสนองของผู้ขับขี่กับความเร็วรถยนต์ [18]



รูปที่ 26 ความสัมพันธ์ระหว่าง Proportional Gain ในการเลี้ยวกับความเร็วรถยนต์ [18]



รูปที่ 27 ความล่าช้าที่เกิดจากพฤติกรรมของการมองทางด้านหน้าของผู้ขับขี่ในการเกี่ยวข้องกับความเร็วรถยนต์ [18]

3.2.5 Time-To-Collision

ในสถานการณ์ที่เป็นอุบัติเหตุการชนระหว่างรถยนต์ 2 คัน เวลาที่เหลือก่อนชน หรือ Time-To-Collision (TTC) สามารถคำนวณได้จาก Headway Distance และความเร็วสัมพัทธ์ของรถยนต์ทั้งสองคัน ดังที่แสดงในสมการที่ (18)

$$TTC = \frac{H - l_{i-1}}{v_i - v_{i-1}} \quad (18)$$

เมื่อ v_i และ v_{i-1} คือความเร็วของรถยนต์ที่ผู้ขับขี่ขับอยู่ และรถยนต์ที่อยู่ด้านหน้าตามลำดับ l_{i-1} คือความยาวของรถยนต์ด้านหน้า Headway Distance (H) สามารถคำนวณได้จากสมการที่ (19) และ Time Headway สามารถคำนวณได้จากสมการที่ (20)

$$H = X_{i-1} - X_i \quad (19)$$

$$H_t = \frac{X_{i-1} - X_i}{v_i} = \frac{H}{v_i} \quad (20)$$

เมื่อ X_{i-1} และ X_i คือตำแหน่งของรถยนต์คันหน้า และรถยนต์ที่ผู้ขับขี่ขับอยู่ตามลำดับ ดังนั้น TTC และ Headway Distance จึงมีความสัมพันธ์กันตามแสดงในสมการที่ (21) [21]

$$TTC = \frac{v_i H_t - l_{i-1}}{v_i - v_{i-1}} \quad (21)$$

Headway Distance ที่สามารถวัดค่าได้บ่อยที่สุดในประเทศไทยมีค่าประมาณ 1 วินาที [22],[23] ในขณะที่ Time Headway ที่แนะนำให้ผู้ขับขี่รักษาไว้มีค่าอยู่ที่ 2 วินาที [21],[23] จาก

ข้อมูลดังกล่าวแสดงให้เห็นว่าผู้ขับขี่ส่วนใหญ่ในประเทศไทยมีการขับเข้าใกล้รถยนต์คันหน้ามากเกินไป ซึ่งทำให้เกิดความเสี่ยงที่จะเกิดอุบัติเหตุได้ นอกจากนี้ การขับรถใกล้รถยนต์คันหน้ามากเกินไปยังเป็นสาเหตุของการเกิดอุบัติเหตุแบบชนท้ายอีกด้วย ถ้ารถยนต์คันหน้าเกิดหยุดกะทันหันในกรณีที่เกิดเหตุฉุกเฉิน

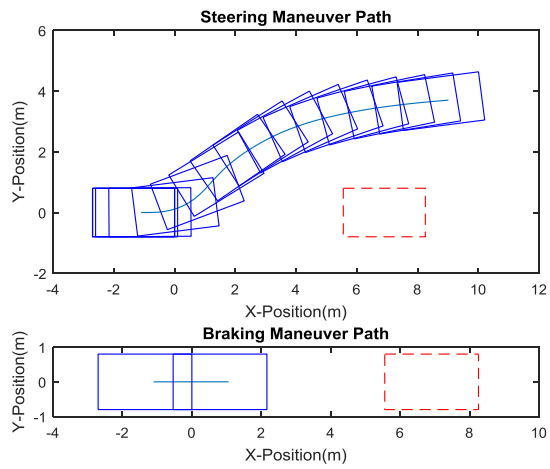
สมมติให้ $v_{i-1} = 0$ (จากการหยุดรถยนต์อย่างกะทันหัน) และ $H_t = 1s$. สมการที่ (4) จะกลายเป็น สมการที่ (22)

$$TTC = 1 - \frac{l_{i-1}}{v_i} \quad (22)$$

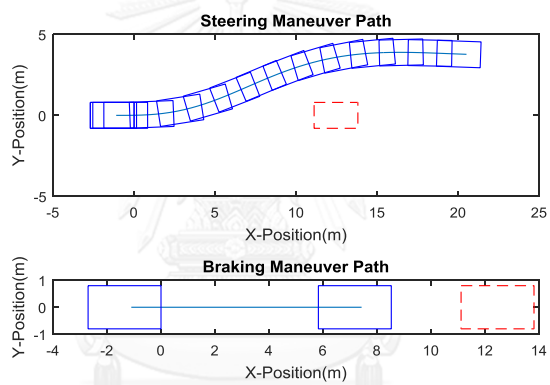
จากสมการที่ (22) การหยุดรถยนต์อย่างกะทันหันทำให้ TTC ของรถยนต์ที่ผู้ขับขี่ขับอยู่ที่มีค่า Time Headway 1 วินาที มีค่า TTC น้อยกว่า 1 วินาทีโดยทันที

3.2.6 การจำลองเส้นทางการหลีกเลี่ยงการชน

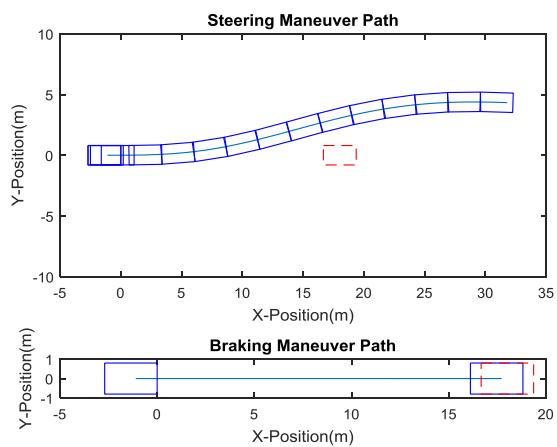
ในงานวิจัยนี้ใช้โปรแกรม Matlab Simulink ในการทำ Model Simulation โดยเหตุการณ์ที่ใช้ในการทำ Model Simulation คือ ผู้ขับขี่ขับรถยนต์ด้วยความเร็วคงที่เข้าสู่เป้าหมายที่หยุดนิ่ง จนกระทั่งมี TTC เท่ากับ 1 วินาที เส้นทางการหลีกเลี่ยงการชน จากทั้งการเบรก และการเลี้ยวหลบ จะถูกคำนวณโดยการใช้ Dynamics Model เส้นทางการหลีกเลี่ยงการชนใช้ในการวิเคราะห์หาความเป็นไปได้ที่จะเกิดการชน และวิธีการหลีกเลี่ยงที่เหมาะสม ในการทำ Model Simulation ใช้ TTC เท่ากับ 1 วินาที จากการนำค่า Time Headway ที่พบบ่อยที่สุดในประเทศไทย มาคำนวณหา TTC ตามสมการที่ (22) ซึ่งผลจากการคำนวณหาเส้นทางการหลีกเลี่ยงการชนแสดงอยู่ในรูปที่ 31, 32, 33 และ 34 [24]



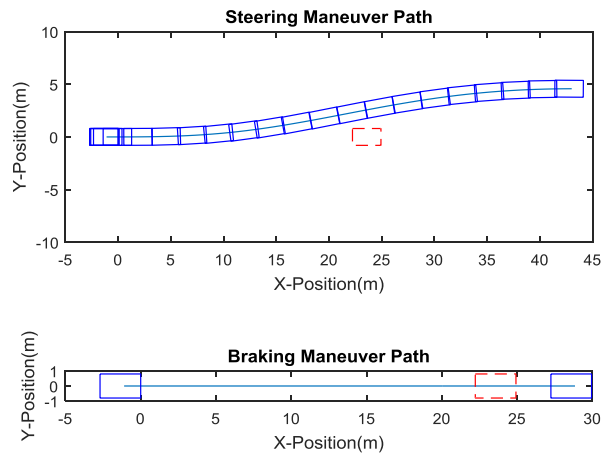
รูปที่ 28 เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 20 กิโลเมตรต่อชั่วโมง



รูปที่ 29 เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง



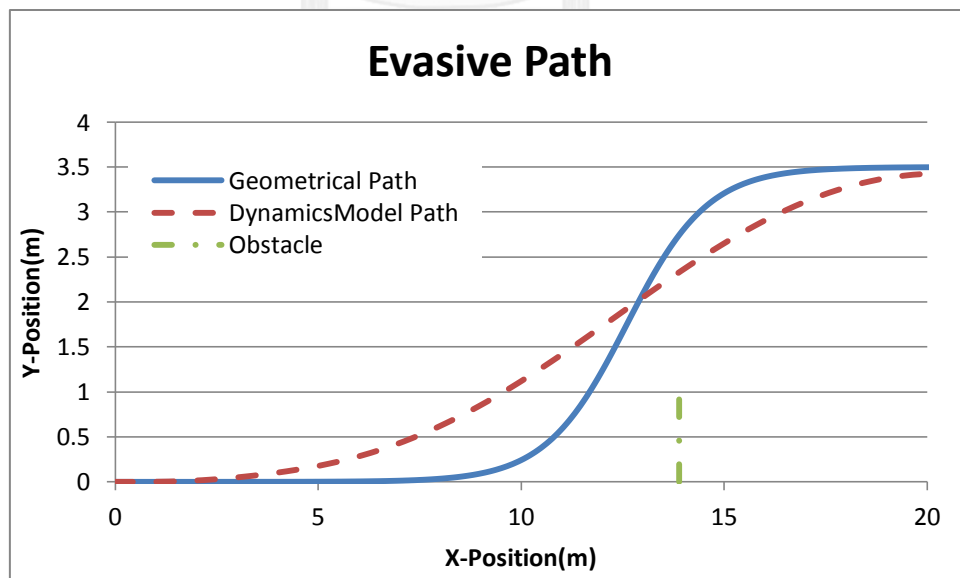
รูปที่ 30 เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 60 กิโลเมตรต่อชั่วโมง



รูปที่ 31 เส้นทางการหลีกเลี่ยงการชนเมื่อรถยนต์มีความเร็ว 80 กิโลเมตรต่อชั่วโมง

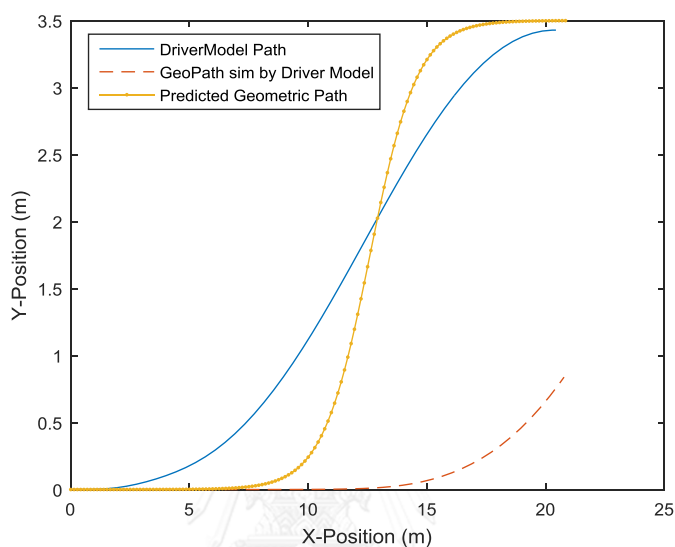
จากผลการคำนวณเส้นทางหลีกเลี่ยงการชนดังที่แสดงในรูปที่ 28, 29, 30 และ 31 จะเห็นได้ว่าที่ TTC เท่ากับ 1 วินาที การเลี้ยวหลบสามารถหลีกเลี่ยงการชนได้ที่ความเร็ว 20, 40, 60 และ 80 กิโลเมตรต่อชั่วโมง ในขณะเดียวกันการเบรกสามารถหลีกเลี่ยงการชนได้แค่เฉพาะที่ความเร็ว 20 และ 40 กิโลเมตรต่อชั่วโมง

3.2.7 การเปรียบเทียบเส้นทางหลีกเลี่ยงการชนที่ได้จาก Geometrical Approach และ Vehicle Dynamics Approach



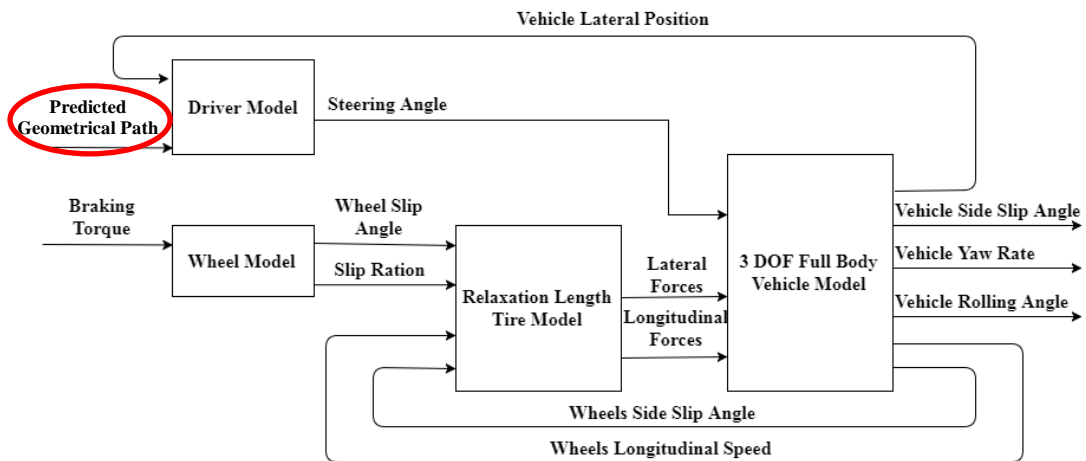
รูปที่ 32 เปรียบเทียบเส้นทางเคลื่อนที่ของรถยนต์ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

จากงานวิจัยที่ผ่านมาจะพบว่าเส้นทางการหลีกเลี่ยงการชนที่ถูกคำนวณด้วยวิธี Vehicle Dynamics Approach และ Geometrical Approach มีความแตกต่างกันดังแสดงในรูปที่ 32 โดยที่เส้นทางการเคลื่อนที่ที่ถูกคำนวณด้วยวิธี Geometrical Approach เริ่มเคลื่อนที่ไปในทิศทางในแนวแกน Y ช้ากว่าเส้นทางที่ได้จาก Vehicle Dynamics Approach และความเร็วในการเคลื่อนที่ของรถยนต์ในทิศทางแกน Y มีมากกว่า



รูปที่ 33 เปรียบเทียบเส้นทางการเคลื่อนที่ของรถยนต์ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

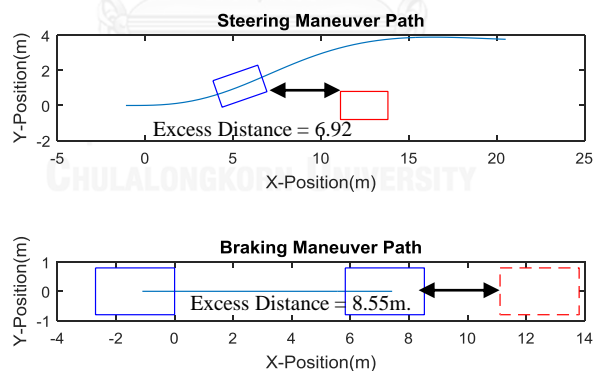
รูปที่ 33 เป็นการแสดงผลการนำเส้นทางการหลีกเลี่ยงการชนจาก Geometrical Approach มาทำการ Simulation โดยให้เส้นทางการหลีกเลี่ยงการชนจากวิธี Geometrical Approach เป็น Input ของระบบดังแสดงในรูปที่ 34 จะเห็นได้ว่าการ Simulation นั้น เส้นทางที่เกิดจาก Geometrical Approach จะเกิดความล่าช้าขึ้นเมื่อถูกควบคุมด้วยผู้ขับขี่ เพราะเส้นทางของวิธี Geometrical Approach นั้น มีความชันมาก จึงต้องการการตอบสนองที่เร็วในการที่จะควบคุมรถยนต์ให้เคลื่อนที่ตามเส้นทางนี้ แต่เนื่องจากผู้ขับขี่มีการตอบสนองไม่เร็วพอ ต่างจาก Controller ที่มีการตอบสนองที่เร็ว การที่จะควบคุมรถยนต์ตามเส้นทางที่ได้จาก Geometrical Approach โดยผู้ขับขี่จึงเป็นไปได้ยาก ส่งผลให้เกิดความล่าช้าขึ้นดังที่แสดงในรูปที่ 33 โดยถ้าจะนำวิธี Geometrical Approach ไปใช้คำนวณเส้นทางการหลีกเลี่ยงการชนเพื่อ ค่า TTC ที่ใช้ในการเตือนจะต้องมีค่ามากกว่า ระบบที่ใช้เส้นทางจากวิธี Vehicle Dynamics Approach



รูปที่ 34 แผนภาพ แบบจำลองทางพลศาสตร์ยานยนต์ ที่มี Input เป็น เส้นทางรถหลีกเลี่ยงการชน ที่ถูกคำนวณด้วยวิธี Geometrical Approach

3.3 แผนภาพ TTC ต่ำสุด

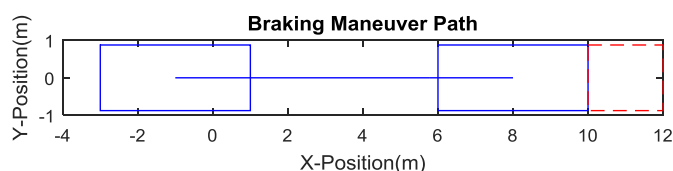
จากการคำนวณหาเส้นทางรถหลีกเลี่ยงการชนจากหัวข้อที่ 3.2.6 ผลการคำนวณจากรูปที่ 28, 29, 30 และ 31 จะเห็นได้ว่ายังมีระยะทางเหลือระหว่างเส้นทางเคลื่อนที่ของรถยนต์ และ สิ่งกีดขวางอยู่ดังแสดงในรูปที่ 35



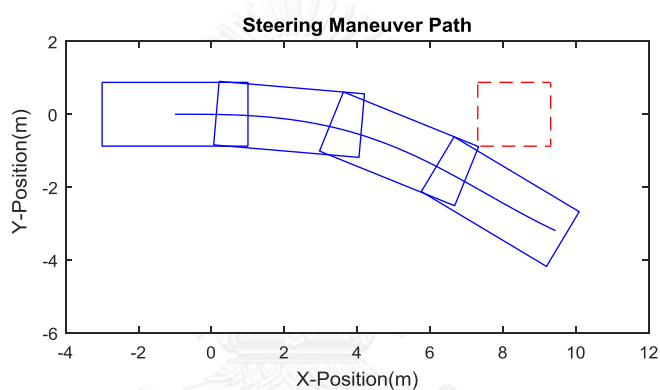
รูปที่ 35 ระยะทางเหลือระหว่างเส้นทางเคลื่อนที่ของรถยนต์ และสิ่งกีดขวาง

จากการที่มีระยะทางเหลือระหว่างเส้นทางเคลื่อนที่ของรถยนต์ และสิ่งกีดขวางดังแสดงในรูปที่ 35 แสดงให้เห็นว่าค่า TTC สามารถที่จะถูกลดลงให้น้อยกว่า 1 วินาทีได้ และอาจจะสามารถมีค่าแตกต่างกันได้ที่ความเร็วต่างกัน ดังนั้นจึงสามารถทำให้คำนวณค่า TTC ต่ำสุดที่สามารถทำให้รถยนต์หลีกเลี่ยงการชนได้ [24]

รูปที่ 36 และ 37 แสดงผลของการ simulation เส้นทางการเคลื่อนที่ของรถยนต์เมื่อใช้การเบรก และการเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ตามลำดับ เมื่อเตือนให้ผู้ขับขี่ทำการเบรก และเลี้ยวหลบที่ค่า TTC ต่ำสุด เท่ากับ 0.9 วินาที ในการเบรกเพื่อหลีกเลี่ยงการชน และ 0.66 วินาที ในการเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

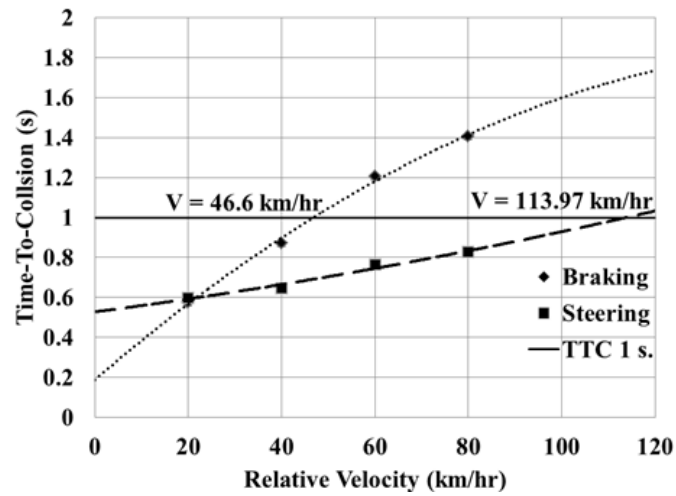


รูปที่ 36 เส้นทางการหลีกเลี่ยงการชนด้วยการเบรกเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง เมื่อเตือนผู้ขับขี่ที่ TTC ต่ำสุด เท่ากับ 0.9 วินาที

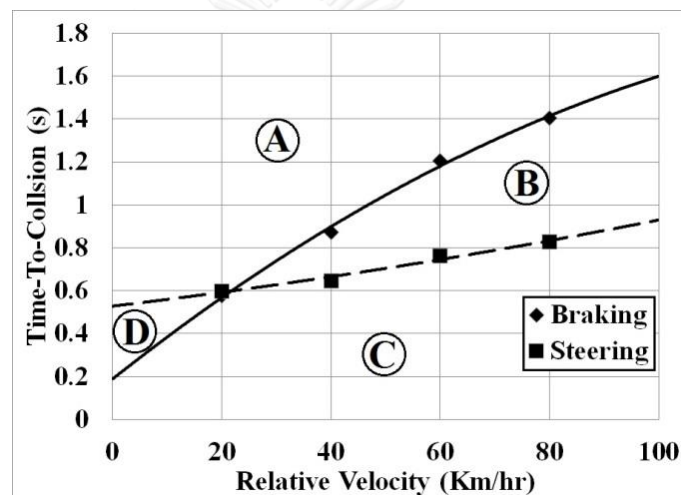


รูปที่ 37 เส้นทางการหลีกเลี่ยงการชนด้วยการเลี้ยวหลบเมื่อรถยนต์มีความเร็ว 40 กิโลเมตรต่อชั่วโมง เมื่อเตือนผู้ขับขี่ที่ TTC ต่ำสุด เท่ากับ 0.66 วินาที

หลังจากคำนวณค่า TTC ต่ำสุดของเส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบที่ความเร็ว 20, 40, 60 และ 80 กิโลเมตรต่อชั่วโมง ทำให้เห็นว่าค่า TTC ที่เตือนสามารถมีค่าแตกต่างกันในแต่ละความเร็ว ที่ความเร็ว 20, 40, 60 และ 80 กิโลเมตรต่อชั่วโมง ดังแสดงในรูปที่ 38 นอกจากนี้ ถ้าเตือนคนขับให้ทำการหลีกเลี่ยงการชนที่ TTC 1 วินาที การเลี้ยวหลบสามารถหลีกเลี่ยงการชนได้เกือบทุกความเร็ว แต่การเบรก สามารถหลีกเลี่ยงการชนได้ที่ความเร็วต่ำกว่า 46.6 กิโลเมตรต่อชั่วโมง ที่ความเร็วสูงกว่า 46.6 กิโลเมตรต่อชั่วโมง การเลี้ยวหลบเท่านั้นที่สามารถหลีกเลี่ยงการชนได้ จนที่ความเร็วสูงกว่า 113.97 กิโลเมตรต่อชั่วโมง ทั้งการเบรก และการเลี้ยวหลบไม่สามารถหลีกเลี่ยงการชนได้

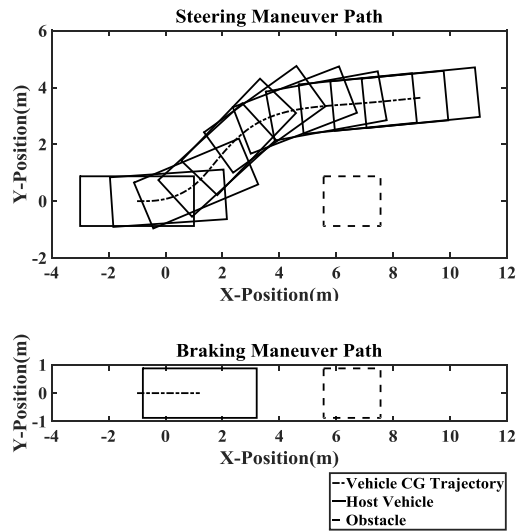


รูปที่ 38 ค่า TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชนที่ความเร็วต่าง ๆ

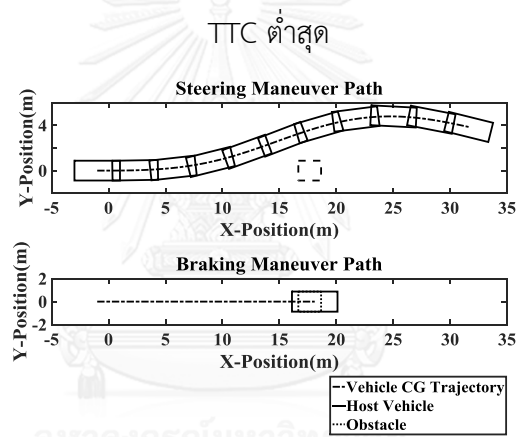


รูปที่ 39 บริเวณภายในแผนภาพ TTC ต่ำสุดที่ทำให้เกิดความสามารถในการหลีกเลี่ยงการชนต่างกัน

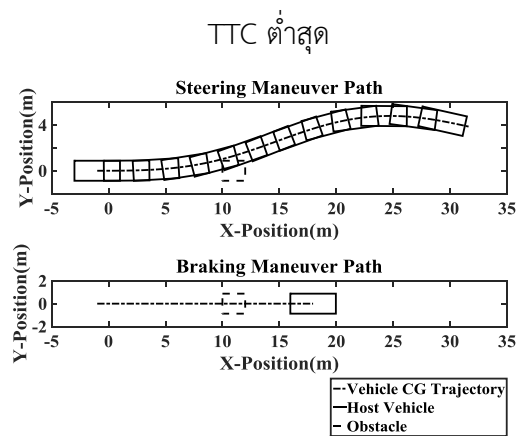
กราฟจากรูปที่ 38 สามารถแบ่งออกได้เป็น 4 บริเวณที่มีความสามารถในการหลีกเลี่ยงการชนแตกต่างกัน ดังแสดงในรูปที่ 39 โดยที่บริเวณ A การเบรก และการเลี้ยวหลบ สามารถหลีกเลี่ยงการชนได้ บริเวณ B การเลี้ยวหลบอย่างเดียวที่สามารถหลีกเลี่ยงการชนได้ บริเวณ C ทั้งการเบรก และการเลี้ยวหลบไม่สามารถหลีกเลี่ยงการชนได้ และสุดท้ายบริเวณ D การเบรกเพียงอย่างเดียวที่สามารถหลีกเลี่ยงการชนได้ ดังแสดงในรูปที่ 40, 41, 42 และ 43 ตามลำดับ



รูปที่ 40 เส้นทางรถหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ A ของแผนภาพ

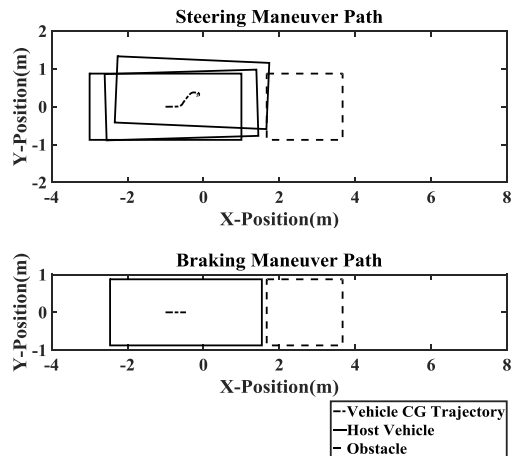


รูปที่ 41 เส้นทางรถหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ B ของแผนภาพ



รูปที่ 42 เส้นทางรถหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ C ของแผนภาพ

TTC ต่ำสุด



รูปที่ 43 เส้นทางการหลีกเลี่ยงการชนด้วยการเบรก และการเลี้ยวหลบในบริเวณ D ของแผนภาพ TTC ต่ำสุด

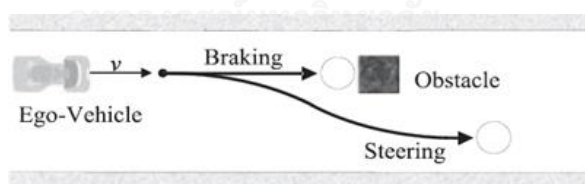
3.4 ผลของ Driver Reaction Time ต่อระบบ Forward Collision Warning System

เพื่อที่จะศึกษาการตอบสนองของผู้ขับขี่ต่อการเตือนด้วยระบบ Forward Collision Warning System โปรแกรม Carnet Soft Driving Simulator ถูกนำมาใช้ เนื่องจากการใช้ Driving Simulator ทำให้เราสามารถออกแบบ และควบคุม สถานการณ์ที่ใช้ในการทดสอบได้ ผลของการใช้ Driving Simulator สามารถแสดงออกถึงผลของการทดสอบจริงได้ [25],[26] สามารถเก็บค่าที่ต้องการได้ง่าย นอกจากนั้นยังช่วยลดความเสี่ยงของการทดสอบด้วยรถยนต์จริงได้

ในการทดสอบด้วย Driving Simulator มีตัวแปรที่ถูกเก็บค่าทั้งหมด 11 ตัวแปรดังแสดงในตารางที่ 4 โดยมีสถานการณ์จำลอง ดังแสดงในรูปที่ 42 ขับขี่ขั้บรถยนต์ด้วยความเร็วคงที่เข้าสู่เป้าหมายที่หยุดนิ่ง จนกระทั่งมี TTC เท่ากับ 1 วินาที ระบบ Forward Collision Warning System จะทำการเตือนผู้ขับขี่ให้เบรก หรือเลี้ยวหลบเพื่อหลีกเลี่ยงการชนตามแผนภาพ TTC ต่ำสุด หลังจากนั้นผู้ขับขี่จะทำการควบคุมรถตามการเตือน และค่าตัวแปรต่าง ๆ จะถูกเก็บแล้วนำไปวิเคราะห์

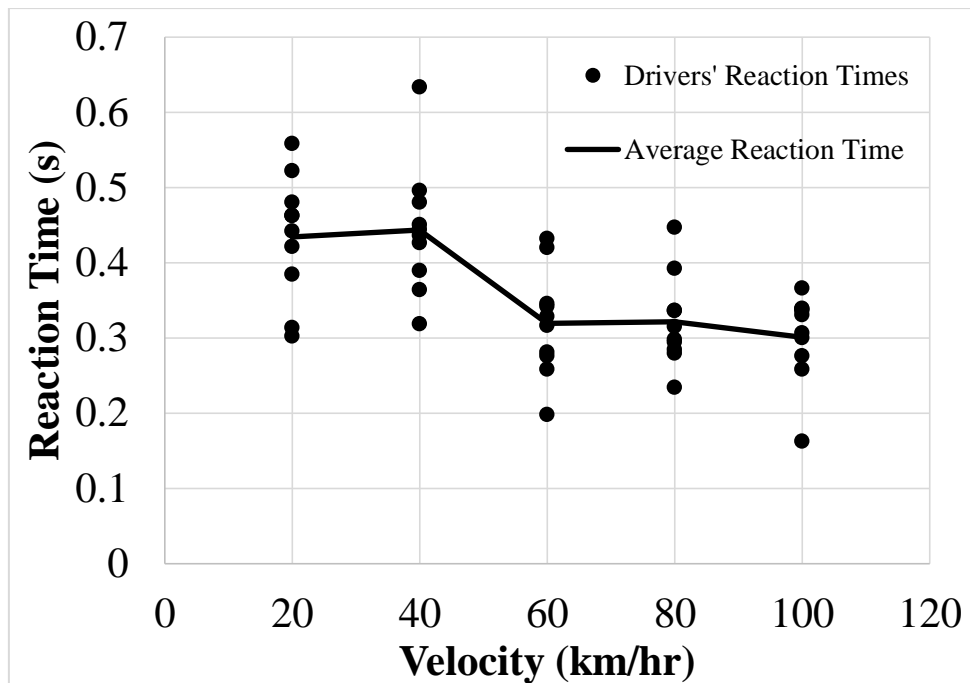
ตารางที่ 4 ตัวแปรที่ถูกเก็บค่าขณะการทดสอบด้วย Driving Simulator

Parameters	Unit
Longitudinal Velocity	m/s
Longitudinal Acceleration	m/s ²
Lateral Velocity	m/s
Lateral Acceleration	m/s ²
Yaw Rate	rad/s
Heading Angle	degree
Brake Pedal Position	percentage
Gas Pedal Position	percentage
Steering Wheel Angle	radians
Time-To-Collision	second
Reaction Time	second



รูปที่ 44 สถานการณ์จำลองที่ใช้ใน Driving Simulator [5]

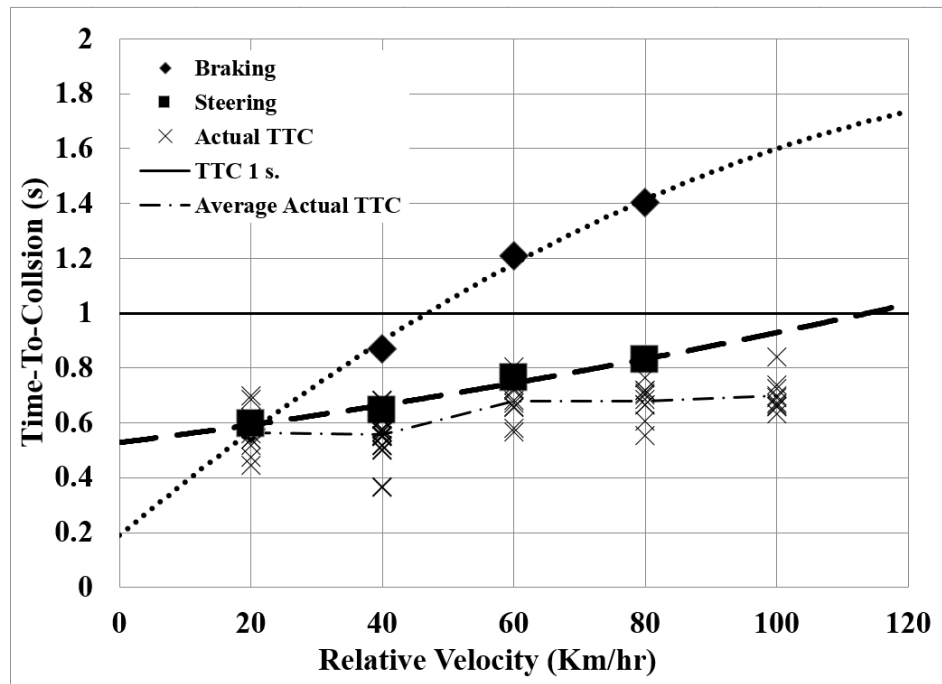
หลังจากผู้ทดสอบซึ่งมีคุณสมบัติดังนี้ 1. ผู้ขับขี่ต้องมีใบอนุญาตขับขี่ที่ยังไม่หมดอายุ 2. ผู้ขับขี่ต้องมีอายุระหว่าง 20 – 25 ปี 3. ผู้ขับขี่ต้องมีการขับขี่รถยนต์อย่างต่อเนื่องในช่วงระยะ 1 เดือนที่ผ่านมา จำนวน 10 ท่าน ได้ทำการทดสอบระบบ Forward Collision Warning System ด้วย Driving Simulator แล้ว ผลของ Driver Reaction Time ถูกแสดงในรูปที่ 44 [27]



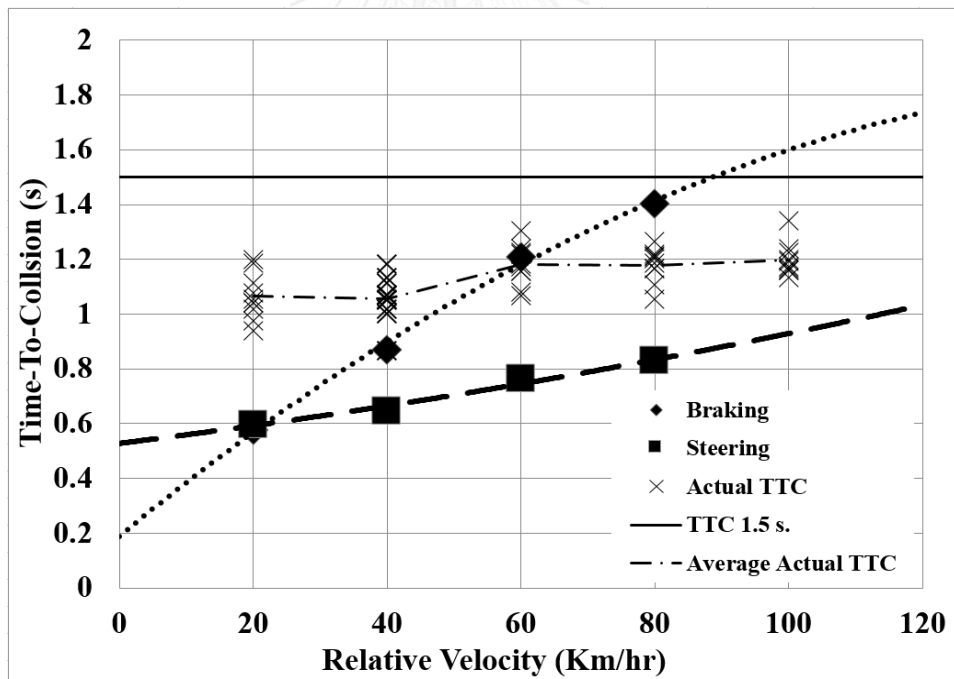
รูปที่ 45 ความสัมพันธ์ระหว่าง Driver Reaction Time ที่ความเร็วต่าง ๆ

จากรูปที่ 45 จะเห็นว่าค่าเฉลี่ยของ Driver Reaction Time ของการทดสอบด้วย Driving Simulation มีค่าที่แตกต่างกันอย่างเห็นได้ชัดอยู่ 2 ค่า คือ 0.44 วินาที ที่ความเร็ว 20 และ 40 กิโลเมตรต่อชั่วโมง และ 0.314 วินาที ที่ความเร็ว 60, 80 และ 100 กิโลเมตรต่อชั่วโมง โดยที่ความแตกต่างของค่า Driver Reaction Time ในแต่ละช่วงเกิดขึ้นมากจาก ความแตกต่างกันของการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน โดยในทางกลับกันค่าความเร็วของรถยนต์ไม่ส่งผลต่อ Driver Reaction Time

Driver Reaction Time นั้นส่งผลกระทบต่อความเป็นไปได้ในการหลีกเลี่ยงการชนอย่างมากดังแสดงในรูปที่ 46 จะเห็นว่าเนื่องจากความล่าช้าเนื่องจาก Driver Reaction Time ทำให้เวลาที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่าน้อยกว่าที่ต้องการ นอกจากนั้นจากการที่ตั้งค่า TTC ไว้ที่ 1 วินาที ความล่าช้าเนื่องจาก Driver Reaction Time ที่ได้จากการทดสอบด้วย Driving Simulator ทำให้โอกาสที่สามารถหลีกเลี่ยงการชน จากสามารถหลีกเลี่ยงการชนได้ เปลี่ยนไปเป็น ไม่สามารถหลีกเลี่ยงการชนได้ โดยถ้าต้องการให้เวลาที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 1 วินาที ค่า TTC ที่ใช้ในการเตือนผู้ขับขี่ต้องถูกชดเชยด้วย ความล่าช้าเนื่องจาก Driver Reaction Time ในกรณีนี้คือ 0.44 วินาที หรือเตือนผู้ขับขี่ที่ TTC 1.5 วินาที โดยประมาณ ซึ่งจะทำให้เวลาที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่าประมาณ 1 วินาทีตามต้องการ [27] ดังแสดงในรูปที่ 47



รูปที่ 46 กราฟแสดงผลกระทบของ Driver Reaction Time ที่ส่งผลต่อ TTC



รูปที่ 47 กราฟแสดง TTC ที่ถูกชดเชยด้วย Driver Reaction Time แล้ว

3.5 ขั้นตอนการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิด

การนำระบบเตือนการชนด้านหน้ามาใช้งานร่วมกับรถยนต์จริงอาจมีความแตกต่างกันไม่ว่าในเรื่องเส้นทางการเคลื่อนที่ของรถยนต์ กับแบบจำลองทางพลศาสตร์ยานยนต์ และการตอบสนองของผู้ขับขี่เองอาจจะแตกต่างจากโปรแกรมจำลองการขับขี่ จึงมีความจำเป็นที่ต้องทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิด เพื่อศึกษาการตอบสนองของผู้ขับขี่ต่อระบบเตือนการชนด้านหน้า และการทำงานของระบบเตือนการชนด้านหน้ากับรถยนต์จริง โดยการทดสอบนั้นถูกทดสอบในสภาวะแวดล้อมแบบปิดด้วยเหตุผลด้านความปลอดภัย และความสามารถในการติดตั้ง และทำซ้ำการทดสอบได้ง่าย

การทดสอบนั้นถูกออกแบบมาให้ผู้ขับขี่ขับรถยนต์เข้าหาเป้าหมายจำลองสภาพรถยนต์ที่หยุดนิ่งด้วยความเร็วคงที่ จนกระทั่งระบบเตือนการชนด้านหน้าเตือนให้ผู้ขับขี่ทำการเบรก หรือเลี้ยวหลบเพื่อหลีกเลี่ยงการชน จากนั้นผู้ขับขี่ถูกกำหนดให้ปฏิบัติตามระบบเตือนการชนด้านหน้า จนกระทั่งสามารถหลีกเลี่ยงการชนได้ โดยตัวแปรที่ถูกเก็บในการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิด แสดงในตารางที่ 5

ตารางที่ 5 ตัวแปรที่ถูกเก็บค่าขณะการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง

Parameters	Unit
Vehicle Velocity	m/s
Driver Reaction Time	s.
Vehicle Lat Position	Degree
Vehicle Lon Position	Degree
Processing Time	ms.
Actual TTC	s.

โดย Vehicle Velocity, Vehicle Lat Position และ Vehicle Lon Position สามารถเก็บค่าได้ผ่านทางอุปกรณ์ GPS ในส่วนของ Driver Reaction Time สามารถวิเคราะห์การตอบสนองผ่าน VDO ที่ถูกบันทึกภายในห้องขับขี่ และนำข้อมูลที่ได้มาคำนวณหาค่า Driver Reaction Time ได้ และค่า Processing Time และ Actual TTC เก็บค่าได้จากระบบเตือนการชนด้านหน้า ซึ่งการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริงในสภาวะแวดล้อมแบบปิด ดังแสดงในรูปที่ 48 และ 49 มีขั้นตอนการทดสอบดังนี้

1. ผู้ทดสอบขับรถยนต์ด้วยความเร็ว 20 กิโลเมตรต่อชั่วโมง มุ่งเข้าสู่เป้าหมายด้วยความเร็วคงที่
2. ระบบเตือนการชนด้านหน้า เตือนผู้ขับขี่ให้ทำการเบรกเพื่อหลีกเลี่ยงการชน เมื่อค่า TTC ต่ำกว่าค่าที่กำหนด
3. ผู้ทดสอบทำการควบคุมรถยนต์ตามที่ระบบเตือนการชนด้านหน้าได้ทำการแนะนำ
4. เส้นทางการเคลื่อนที่, ความเร็วของรถยนต์, TTC, Processing Time, และ Driver Reaction Time ถูกบันทึกไว้
5. ทำซ้ำข้อ 1 – 4 โดยระบบเตือนการชนด้านหน้าเตือนให้ผู้ขับขี่เลี้ยวหลบเพื่อหลีกเลี่ยงการชน
6. ทำซ้ำข้อ 1 – 5 ด้วยความเร็ว 30 และ 40 กิโลเมตรต่อชั่วโมง



รูปที่ 48 การทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิดขณะที่รถยนต์เคลื่อนที่เข้าใกล้เป้าหมาย



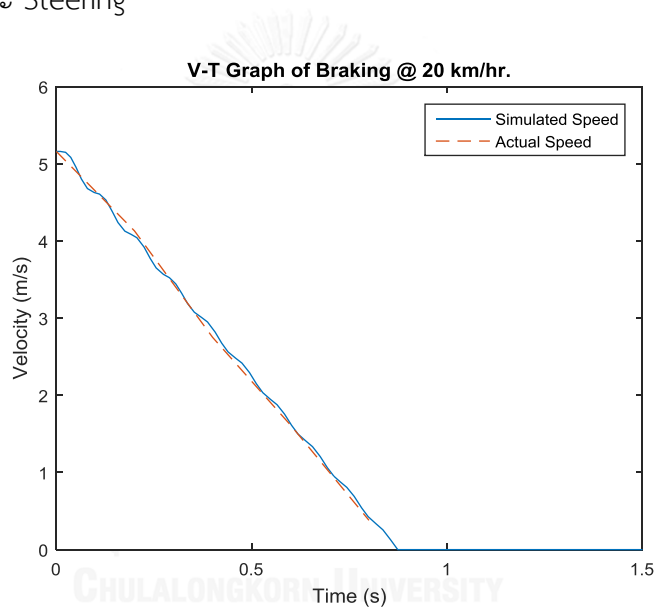
รูปที่ 49 ภาพขณะระบบเตือนการชนด้านหน้าเตือนให้ผู้ขับขี่เบรกเพื่อหลีกเลี่ยงการชนด้วย LED

บทที่ 4

ผลการทดสอบ

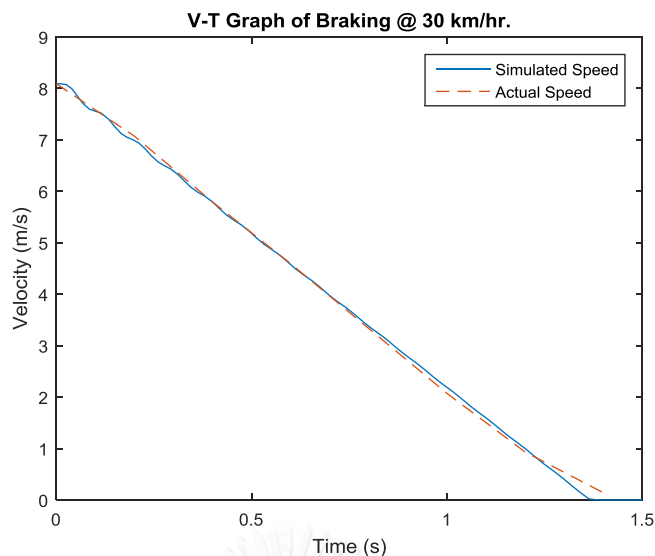
4.1 Model Evaluation

เนื่องจาก Dynamics model ถูกนำมาใช้ในการคำนวณค่า TTC ต่ำสุด ซึ่งจะถูกนำมาวิเคราะห์โอกาสในการเกิดการชนของรถยนต์ ในความเร็วต่าง ๆ ดังนั้น การทำ Model Evaluation จึงมีความจำเป็นในการยืนยันว่า Dynamics Model ที่ถูกนำมาใช้สามารถแสดงถึงผลที่เกิดขึ้นจริง และสามารถนำมาใช้ได้ตามที่วิจัยไว้ ซึ่งถูกแบ่งออกเป็น 2 วิธี ตามลักษณะวิธีการที่ใช้ในการหลีกเลี่ยงการชนคือ Braking และ Steering



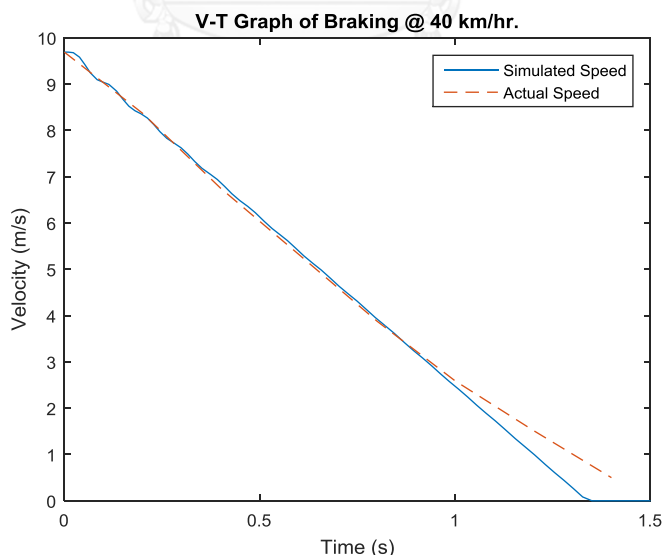
รูปที่ 50 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 20 กม./ชม.

รูปที่ 50 แสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 20 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า รถยนต์มีความเร็วเริ่มต้นที่ 5.16 เมตรต่อวินาที หลังจากทำการเบรกไป 0.8737 วินาที รถยนต์จึงหยุดนิ่ง



รูปที่ 51 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 30 กม./ชม.

รูปที่ 51 แสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า รถยนต์มีความเร็วเริ่มต้นที่ 8.09 เมตรต่อวินาที หลังจากทำการเบรกไป 1.385 วินาที รถยนต์จึงหยุดนิ่ง



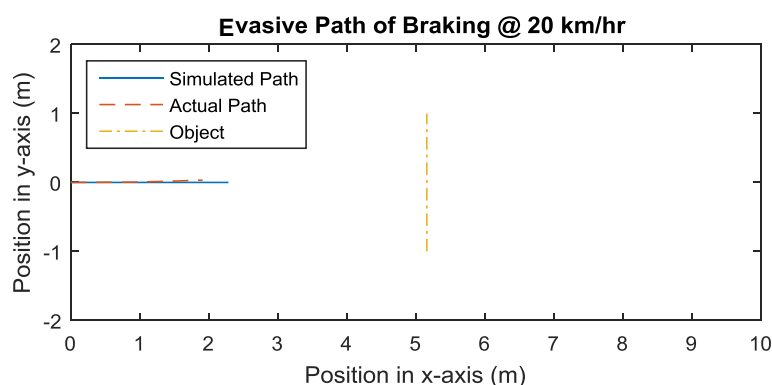
รูปที่ 52 กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 40 กม./ชม.

รูปที่ 52 แสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลาในขณะที่เบรกเพื่อหลีกเลี่ยงการชนที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า รถยนต์มีความเร็วเริ่มต้นที่ 9.69 เมตรต่อวินาที หลังจากทำการเบรกไป 1.35 วินาที รถยนต์จึงหยุดนิ่ง

โดยรูปที่ 50 – 52 แสดงให้เห็นว่า ลักษณะของความเร็วที่ลดลงมีแนวโน้มที่เหมือนกัน ทั้งจากการทดสอบ และจากการ Simulation แต่ในช่วงที่รถยนต์ใกล้จะหยุดนิ่ง ความเร็วที่ได้จากการทดสอบที่สภาวะความเร็ว 30 และ 40 กิโลเมตรต่อชั่วโมง อัตราการลดลงของความเร็วรถยนต์นั้นลดลง ทำให้กราฟความเร็วที่มาจาก การทดสอบมีลักษณะเป็น 2 ช่วงที่มีความหน่วงต่างกัน และนอกจากนั้น กราฟจากการทดสอบทั้ง 3 ความเร็ว มีข้อมูลช่วงสุดท้ายของกราฟขาดหายไปอีกด้วย

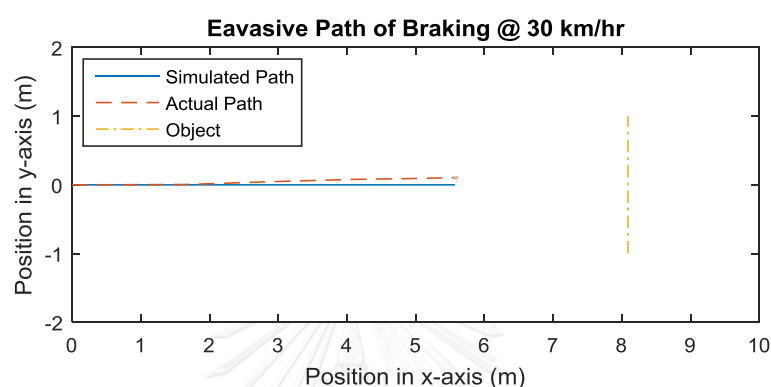
กราฟความเร็วที่ได้จากการทดสอบในรูปที่ 51 – 52 มีลักษณะเป็น 2 ช่วงที่มีความหน่วงต่างกัน มีสาเหตุมาจาก การทดสอบด้วยรถยนต์จริงนั้น เมื่อผู้ขับขี่ได้ทำการเบรกรถยนต์เพื่อหลีกเลี่ยงการชนแล้ว ในขณะที่รถยนต์มีความเร็วลดลง และผู้ขับขี่ได้พิจารณาแล้วว่าสามารถหลีกเลี่ยงการชนได้แล้ว ผู้ขับขี่จึงทำการเบรกรถยนต์น้อยลงให้แค่พอให้รถยนต์หยุดได้ นี่จึงเป็นสาเหตุของการที่กราฟความเร็วที่ได้จากการทดสอบมีลักษณะเป็น 2 ช่วงที่มีความหน่วงต่างกัน

ในส่วนสุดท้ายของกราฟความเร็วของรถยนต์ที่ได้จากการทดสอบในรูปที่ 50 – 52 ที่มีข้อมูลบางส่วนหายไปสืบเนื่องมาจากช่วงที่รถยนต์หยุดนิ่ง หรือมีความเร็วเป็น 0 นั้น ข้อมูลไม่ได้ถูกเก็บเนื่องจากอุปกรณ์เก็บข้อมูลมีค่าความถี่ในการเก็บข้อมูลอยู่ที่ 5 ข้อมูลทุก ๆ 1 วินาที จากเหตุผลนี้ทำให้ความเร็วที่เก็บได้มีค่าหายไปบางส่วนเมื่อขณะที่รถยนต์มีความเร็วเป็น 0 แต่อุปกรณ์เก็บข้อมูลยังไม่ได้เก็บข้อมูล จึงส่งผลถึงเส้นทางการหลีกเลี่ยงการชนจากการเบรกที่เกิดขึ้น



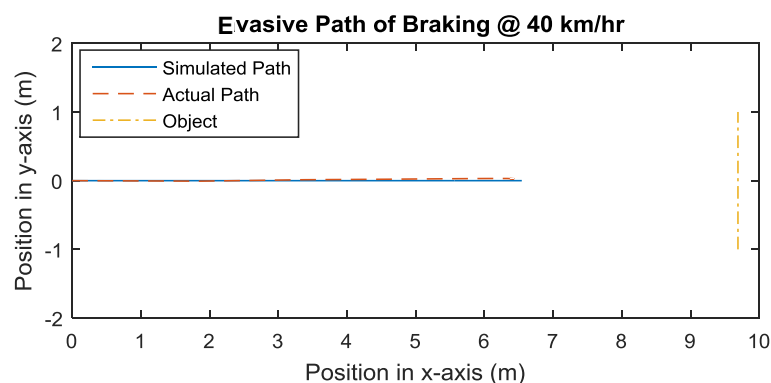
รูปที่ 53 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 20 กิโลเมตรต่อชั่วโมง

รูปที่ 53 แสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 20 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า จากการทำ Model Simulation รถยนต์จะเคลื่อนที่หลังจากเริ่มเบรกไปได้ 2.285 เมตร และจากการทดสอบ 1.912 เมตร ทำให้เมื่อรถยนต์หยุดนิ่ง รถยนต์มีระยะห่างจากวัตถุเป้าหมายในกรณีที่รถยนต์เริ่มเบรกเมื่อค่า TTC เท่ากับ 1 วินาที 2.875 เมตร และ 3.248 เมตร ตามลำดับ



รูปที่ 54 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง

รูปที่ 54 แสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า จากการทำ Model Simulation รถยนต์จะเคลื่อนที่หลังจากเริ่มเบรกไปได้ 5.574 เมตร และจากการทดสอบ 5.609 เมตร ทำให้เมื่อรถยนต์หยุดนิ่ง รถยนต์มีระยะห่างจากวัตถุเป้าหมายในกรณีที่รถยนต์เริ่มเบรกเมื่อค่า TTC เท่ากับ 1 วินาที 2.516 เมตร และ 2.361 เมตร ตามลำดับ

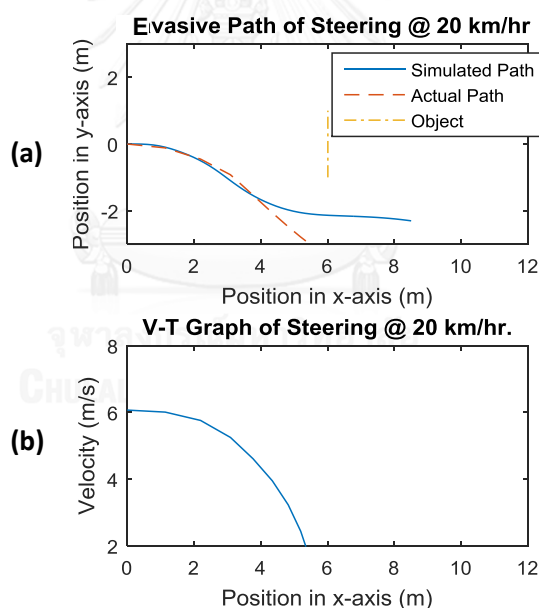


รูปที่ 55 กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง

รูปที่ 55 แสดงเส้นทางการเคลื่อนที่ของรถยนต์ขณะที่เบรกเพื่อหลีกเลี่ยงการชน จากการทำ Model Simulation และจากการทดสอบ ที่ความเร็ว 40 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า จากการทำ Model Simulation รถยนต์จะเคลื่อนที่หลังจากเริ่มเบรกไปได้ 6.433 เมตร และจากการทดสอบ 6.544 เมตร ทำให้เมื่อรถยนต์หยุดนิ่ง รถยนต์มีระยะห่างจากวัตถุเป้าหมายในกรณีที่รถยนต์เริ่มเบรกเมื่อค่า TTC เท่ากับ 1 วินาที 3.257 เมตร และ 3.146 เมตร ตามลำดับ

รูปที่ 53 – 55 แสดงให้เห็นถึง เส้นทางที่เกิดจากการทดสอบ ผู้ขับขี่มีการบังคับรถยนต์เบี่ยงออกด้านข้างเล็กน้อย และจากรูปที่ 50 – 52 ความเร็วในช่วงสุดท้ายข้อมูลหายไป ข้อมูลที่หายไปนั้นส่งผลถึงเส้นทางที่เกิดขึ้นทำให้เส้นทางสั้นกว่าที่เกิดจาก Model Simulation

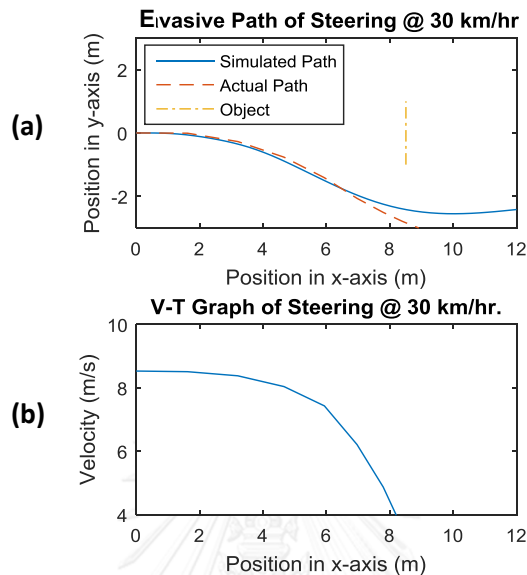
ผลจากการทดสอบแสดงให้เห็นว่าเส้นทางที่เกิดขึ้นมีความใกล้เคียงกันมากในการเบรกเพื่อหลีกเลี่ยงการชน ดังนั้น Model ที่ใช้ในการ Simulation เพื่อหาเส้นทางการเบรกเพื่อหลีกเลี่ยงการชนสามารถแสดงให้เห็นถึงผลที่จะเกิดจากการเบรกจริงได้



รูปที่ 56 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะที่เลี้ยวหลบเพื่อหลีกเลี่ยงการชนความเร็ว 20 กม./ชม.

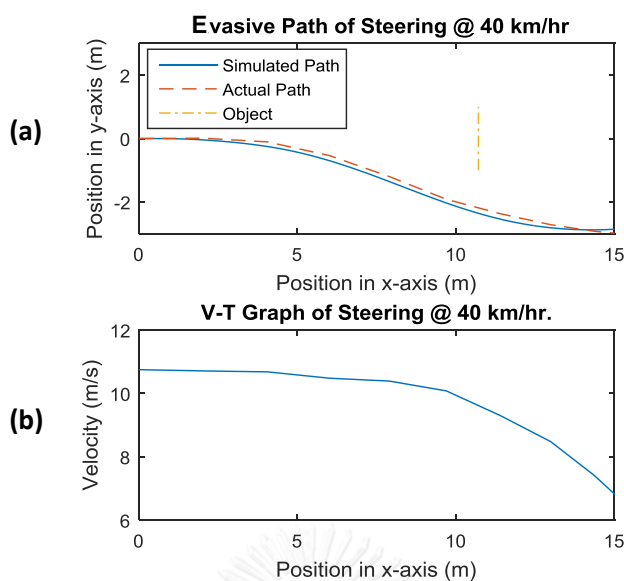
รูปที่ 56 (a) แสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ ขณะที่เลี้ยวหลบเพื่อหลีกเลี่ยงการชน ที่ความเร็ว 20 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า ในช่วงแรกจนเมื่อรถยนต์เคลื่อนที่ในแกน x ได้

ระยะทางประมาณ 2 เมตร เส้นทางการเคลื่อนที่ของรถยนต์ที่ได้จากการทดสอบเริ่มแตกต่างจากการทำ Model Simulation และในช่วงระยะเดียวกัน ความเร็วของรถยนต์เริ่มลดต่ำลงเนื่องจากการเบรกของผู้ขับขี่ ดังแสดงในรูปที่ 56 (b)



รูปที่ 57 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะที่เลี้ยวหลบเพื่อหลีกเลี่ยงการชนความเร็ว 30 กม./ชม.

รูปที่ 57 (a) แสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบขณะที่เลี้ยวหลบเพื่อหลีกเลี่ยงการชน ที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า ในช่วงแรกจนเมื่อรถยนต์เคลื่อนที่ในแกน x ได้ระยะทางประมาณ 6 เมตร เส้นทางการเคลื่อนที่ของรถยนต์ที่ได้จากการทดสอบเริ่มแตกต่างจากการทำ Model Simulation และเมื่อรถยนต์เคลื่อนที่ในแกน x ได้ระยะทางประมาณ 4.663 เมตร ความเร็วของรถยนต์เริ่มลดต่ำลงเนื่องจากการเบรกของผู้ขับขี่ ดังแสดงในรูปที่ 57 (b)



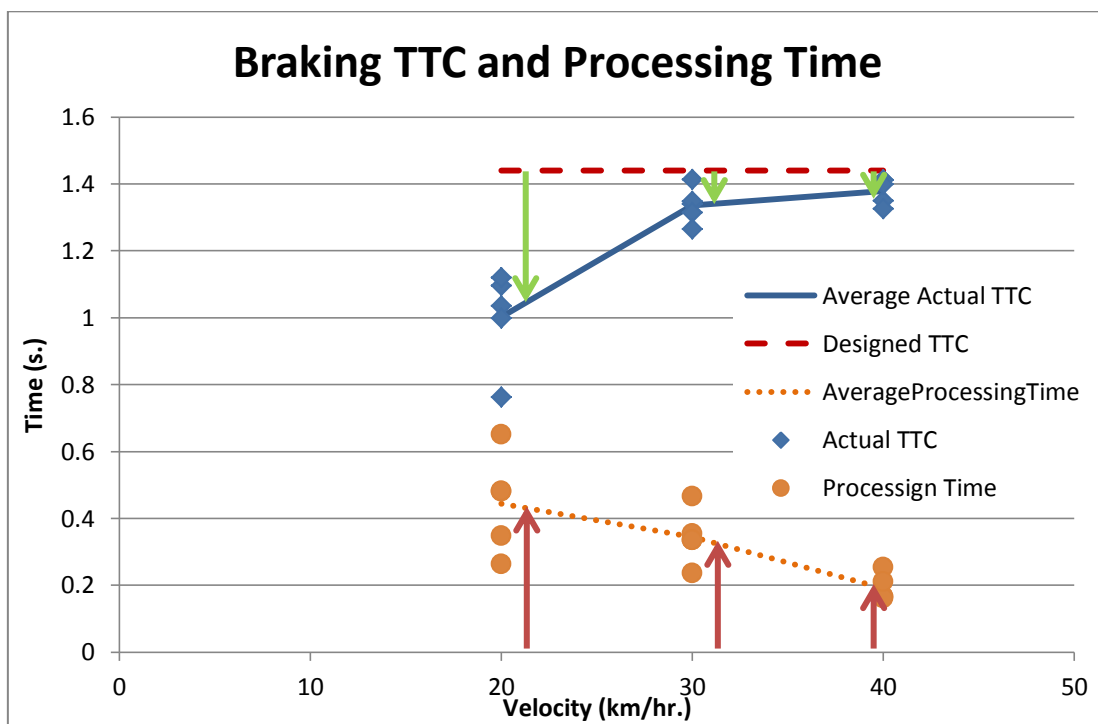
รูปที่ 58 (a) กราฟแสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบ (b) กราฟแสดงความสัมพันธ์ระหว่างความเร็วของรถยนต์กับเวลา ขณะที่ยellowหลบเพื่อหลีกเลี่ยงการชนความเร็ว 40 กม./ชม.

รูปที่ 58 (a) แสดงเส้นทางการเคลื่อนที่ของรถยนต์จากการทำ Model Simulation และจากการทดสอบขณะที่yellowหลบเพื่อหลีกเลี่ยงการชน ที่ความเร็ว 30 กิโลเมตรต่อชั่วโมง จาก ณ เวลาที่รถยนต์เริ่มเบรก จนถึงหยุดนิ่ง จากกราฟจะเห็นได้ว่า ในช่วงแรกจนเมื่อรถยนต์เคลื่อนที่ในแกน x ได้ระยะทางประมาณ 14 เมตร เส้นทางการเคลื่อนที่ของรถยนต์ที่ได้จากการทดสอบเริ่มแตกต่างจากการทำ Model Simulation และเมื่อรถยนต์เคลื่อนที่ในแกน x ได้ระยะทางประมาณ 9.697 เมตร ความเร็วของรถยนต์เริ่มลดต่ำลงเนื่องจากการเบรกของผู้ขับขี่ ดังแสดงในรูปที่ 58 (b)

ความแตกต่างของ เส้นทางการเคลื่อนที่ของรถยนต์ขณะที่yellowหลบเพื่อหลีกเลี่ยงการชน ระหว่างการทำ Model Simulation และการทดสอบ สืบเนื่องมาจากการที่ผู้ขับขี่ทำการเบรกรถยนต์เมื่อผู้ขับขี่พิจารณาแล้วว่าสามารถหลีกเลี่ยงการชนโดยการหลบได้แล้ว ผู้ขับขี่จึงทำการเบรกรถยนต์เพื่อให้รถยนต์หยุด ซึ่งแตกต่างจากตัว Model เนื่องจาก Model ที่ใช้นั้นมีการรักษาความเร็วของรถให้คงที่ และในส่วนของ Driver Model มีการกำหนดเป้าหมายแตกต่างจากการทดสอบ โดยการทดสอบด้วยรถยนต์จริงมีการกำหนดเป้าหมายแค่ให้yellowหลบวัตถุได้เท่านั้น แต่ตัว Driver Model มีการกำหนดให้รถยนต์ทำการเปลี่ยนช่องจราจร และคงตำแหน่งตามที่กำหนดไว้

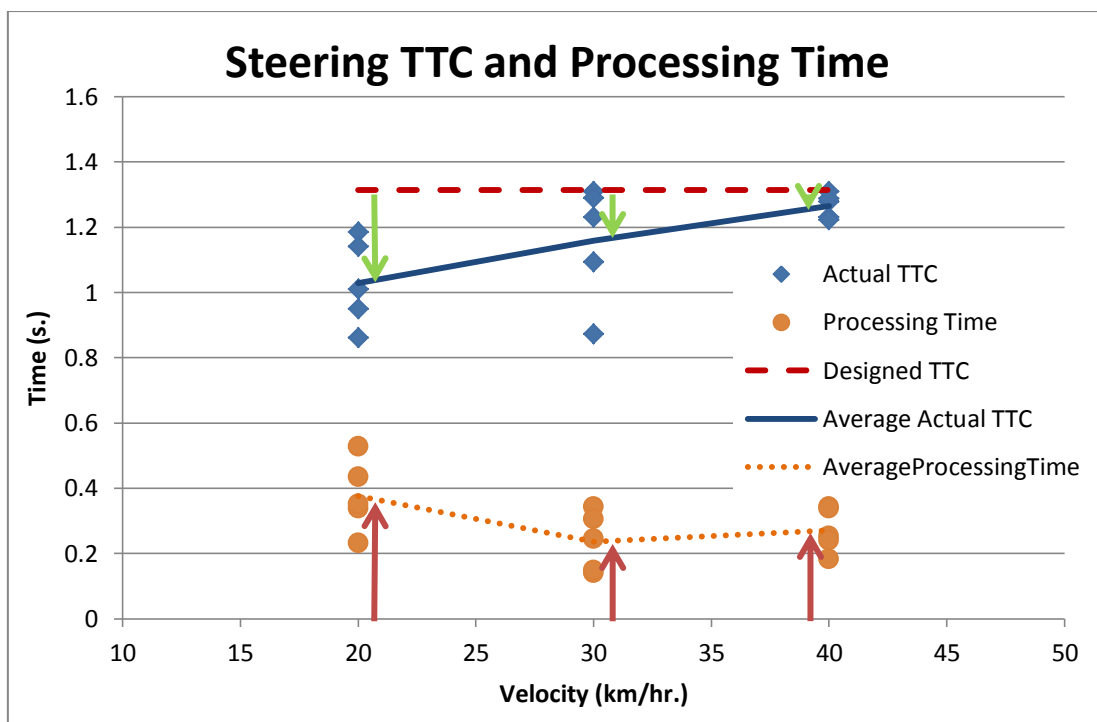
4.2. TTC ที่ได้จากการทดสอบ

หลังจากทำ Model Evaluation เพื่อให้ทราบถึงความสามารถในการจำลองสภาวะจริงของ Dynamics Model ที่นำมาใช้ ค่า TTC ที่ระบบเตือนในการทดสอบจริงถูกเก็บค่าเพื่อมาเปรียบเทียบกับค่า TTC ที่ระบบต้องเตือนตามที่กำหนดไว้



รูปที่ 59 กราฟแสดงความสัมพันธ์ระหว่าง TTC และ Processing Time กับความเร็วของรถยนต์ขณะเบรกเพื่อหลีกเลี่ยงการชน

รูปที่ 59 แสดงค่า TTC ที่ระบบเตือนเมื่อทดสอบด้วยรถยนต์ที่มีความเร็วต่าง ๆ โดยค่า TTC เฉลี่ยที่ระบบเตือนผู้ขับขี่เพื่อเตือนให้ผู้ขับขี่ทำการเบรกเพื่อหลีกเลี่ยงการชนของรถยนต์ที่มีความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง คือ 1.002, 1.336 และ 1.379 วินาทีตามลำดับ นอกจากนี้ค่าเวลาในการประมวลผลเฉลี่ยของระบบเพื่อเตือนให้ผู้ขับขี่ทำการเบรกเพื่อหลีกเลี่ยงการชนของรถยนต์ที่มีความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง คือ 0.444, 0.345 และ 0.191 วินาทีตามลำดับ



รูปที่ 60 กราฟแสดงความสัมพันธ์ระหว่าง TTC และ Processing Time กับความเร็วของรถยนต์ ขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

รูปที่ 60 แสดงค่า TTC ที่ระบบเตือนเมื่อทดสอบด้วยรถยนต์ที่ความเร็วต่าง ๆ โดยค่า TTC เฉลี่ยที่ระบบเตือนผู้ขับขี่เพื่อเตือนให้ผู้ขับขี่ทำการเบรกเพื่อหลีกเลี่ยงการชนของรถยนต์ที่มีความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง คือ 1.029, 1.159 และ 1.265 วินาทีตามลำดับ นอกจากนี้ค่า เวลาในการประมวลผลเฉลี่ยของระบบเพื่อเตือนให้ผู้ขับขี่ทำการเบรกเพื่อหลีกเลี่ยงการชนของรถยนต์ ที่มีความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง คือ 0.376, 0.236 และ 0.271 วินาทีตามลำดับ

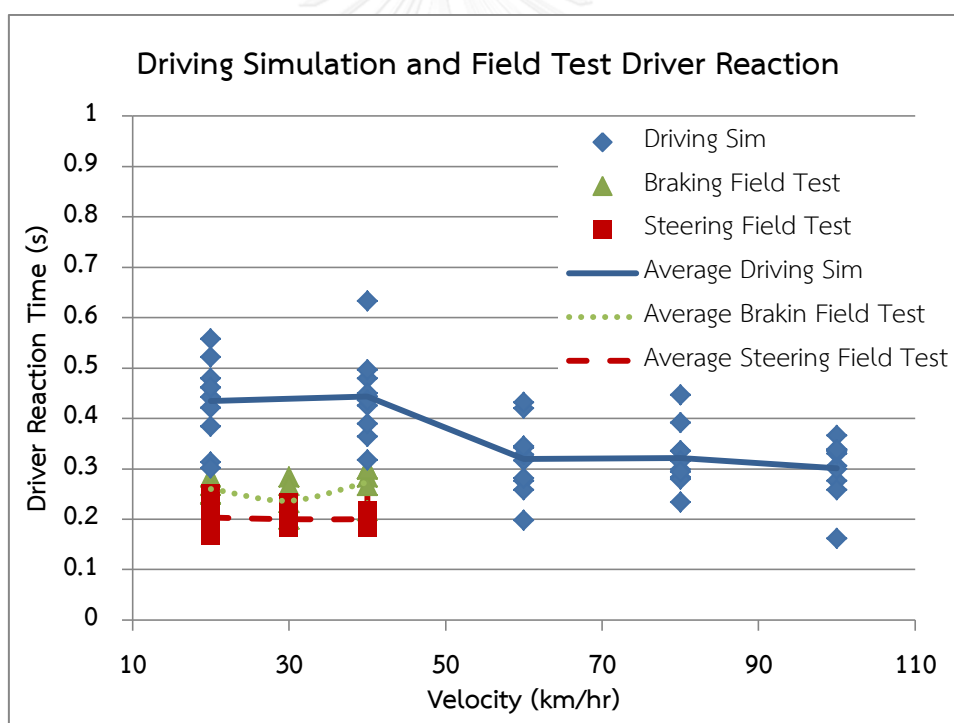
จากการทดสอบพบว่า TTC ที่ระบบเตือนในการทดสอบจริง มีค่าน้อยกว่า TTC ที่ระบบต้องเตือนตาม Model Simulation สืบเนื่องจากระบบมีความล่าช้าเนื่องจากเวลาในการประมวลผล เนื่องจากระบบได้รับข้อมูลสภาพแวดล้อมผ่านกล้องสเตอริโอในช่วงต้นของการประมวลผล แต่ระบบจะนำเตือนได้เมื่อผ่านการประมวลผลต่าง ๆ ไปแล้วดังที่แสดงในรูปที่ 7 ซึ่งจะเกิดความล่าช้า เนื่องจากเวลาในการประมวลผลขึ้น

ระบบจะมีความล่าช้าเพิ่มขึ้นเมื่อความเร็วในการเคลื่อนที่ของรถยนต์ลดลง สืบเนื่องมาจากระบบ ต้องใช้เวลาประมวลผลในการคำนวณระยะ ในระยะสั้น ๆ มากขึ้น โดยในทางกลับกันที่ความเร็วสูง ระบบสามารถประมวลผลได้เร็วจึงทำให้ TTC ที่เตือนมีค่าใกล้เคียงกับค่าที่ได้จากการทำ Simulation โดยเวลาในการประมวลผลนี้มีลักษณะเช่นเดียวกันทั้งการ Braking และ Steering ที่มีระยะเวลาในการประมวลผลที่ความเร็วต่ำๆมากขึ้น ซึ่งสาเหตุของความเร็วในการประมวลผลที่มากในการคำนวณ

ระยะทางสั้น ๆ สืบเนื่องมาจากการที่รถยนต์เข้าใกล้เป้าหมายในระยะใกล้ส่งผลทำให้ขนาดวัตถุที่บันทึกได้โดยกล้องสเตอริโอวิชันมีขนาดใหญ่ขึ้น และตำแหน่งในแนวนอนของภาพทั้ง 2 ภาพ ที่ได้จากกล้องสเตอริโอวิชันต่างกันมาก จึงส่งผลถึงความเร็วในการประมวลผลเพื่อสร้างแผนภาพ Disparity ของระบบเตือนการชนด้านหน้า ทำให้ในการใช้งานที่ความเร็วต่ำส่งผลทำให้ความเร็วในการประมวลผลของระบบมากขึ้น

4.3 ความล่าช้าเนื่องจาก Driver Reaction Time

จากการทำ Driving Simulation จะเห็นได้ว่า Delay ที่เกิดจาก Reaction time ของผู้ขับขี่ ส่งผลต่อระยะเวลาที่เหลือที่สามารถให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน โดยอาจจะมีโอกาสทำให้เกิดการชนได้ แม้ว่าการเตือนอยู่ในช่วงที่สามารถหลีกเลี่ยงการชนได้ ในการทดสอบระบบที่ใช้กับรถยนต์ในสภาวะปิดจึงมีการเก็บข้อมูลของ Driver Reaction Time ด้วย

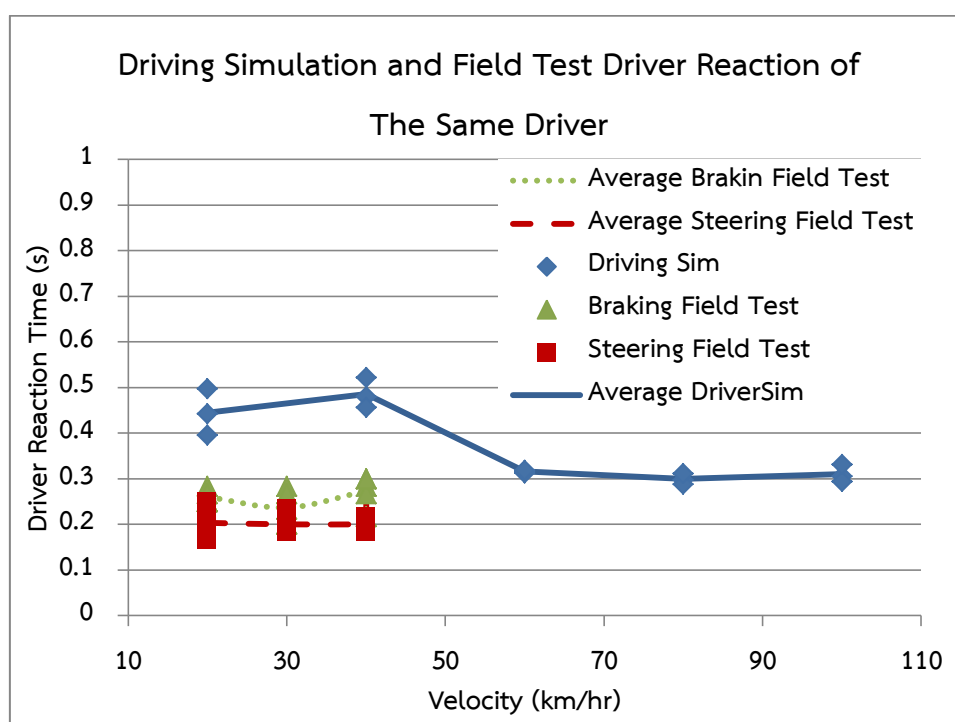


รูปที่ 61 กราฟแสดงความสัมพันธ์ระหว่าง Driver Reaction Time กับความเร็วของรถยนต์ ของ Driving Simulator และการทดสอบ

ค่าเฉลี่ยของ Driver Reaction Time ของการทดสอบด้วย Driving Simulation มีค่าที่แตกต่างกันอย่างเห็นได้ชัดอยู่ 2 ค่า คือ 0.44 วินาที ที่ความเร็ว 20 และ 40 กิโลเมตรต่อชั่วโมง และ 0.314 วินาที ที่ความเร็ว 60, 80 และ 100 กิโลเมตรต่อชั่วโมง โดยที่ความแตกต่างของค่า Driver Reaction

Time ในแต่ละช่วงเกิดขึ้นมาจาก ความแตกต่างกันของการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน โดยในทางกลับกันค่าความเร็วของรถยนต์ไม่ส่งผลต่อ Driver Reaction Time

ค่าเฉลี่ยของ Driver Reaction Time ของการทดสอบด้วยรถยนต์จริงในช่วงความเร็ว 20 30 และ 40 กิโลเมตรต่อชั่วโมง มีค่าที่แตกต่างกันตามการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน โดยค่า Driver Reaction Time เฉลี่ยเมื่อผู้ขับขี่เบรกรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 0.257 วินาที และค่า Driver Reaction Time เฉลี่ยเมื่อผู้ขับขี่เบรกรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 0.201 วินาที ซึ่งความเร็วของรถยนต์ไม่ส่งผลต่อ Driver Reaction Time

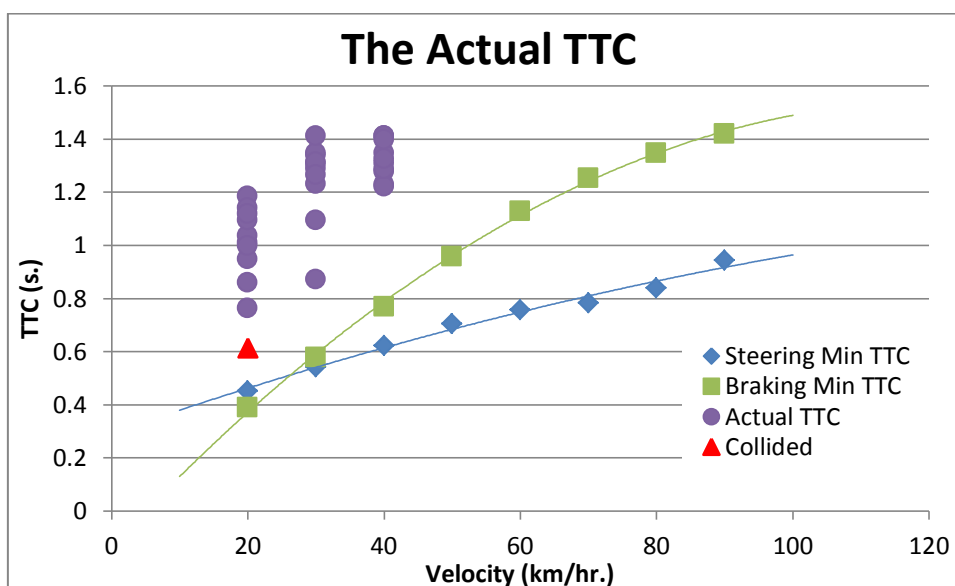


รูปที่ 62 กราฟแสดงความสัมพันธ์ระหว่าง Driver Reaction Time กับความเร็วของรถยนต์ ของ Driving Simulator และ การทดสอบจริง ของผู้ขับขี่คนเดียวกัน

จากรูปที่ 62 เป็นการเปรียบเทียบค่า Driver Reaction Time ที่ได้จากการทดสอบด้วย Driving Simulator และจากการทดสอบด้วยรถยนต์จริงที่ขับด้วยผู้ขับขี่คนเดียวกัน ผลการทดสอบแสดงให้เห็นว่า ค่าเฉลี่ย Driver Reaction Time จากการทดสอบด้วย Driving Simulation จากการ เบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ซึ่งมีค่าเท่ากับ 0.466 และ 0.308 วินาทีตามลำดับ มีค่ามากกว่า ค่า Driver Reaction Time จากการ เบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ที่ได้จากการทดสอบด้วยรถยนต์จริง ซึ่งมีค่าเท่ากับ 0.257 และ 0.201 วินาทีตามลำดับ

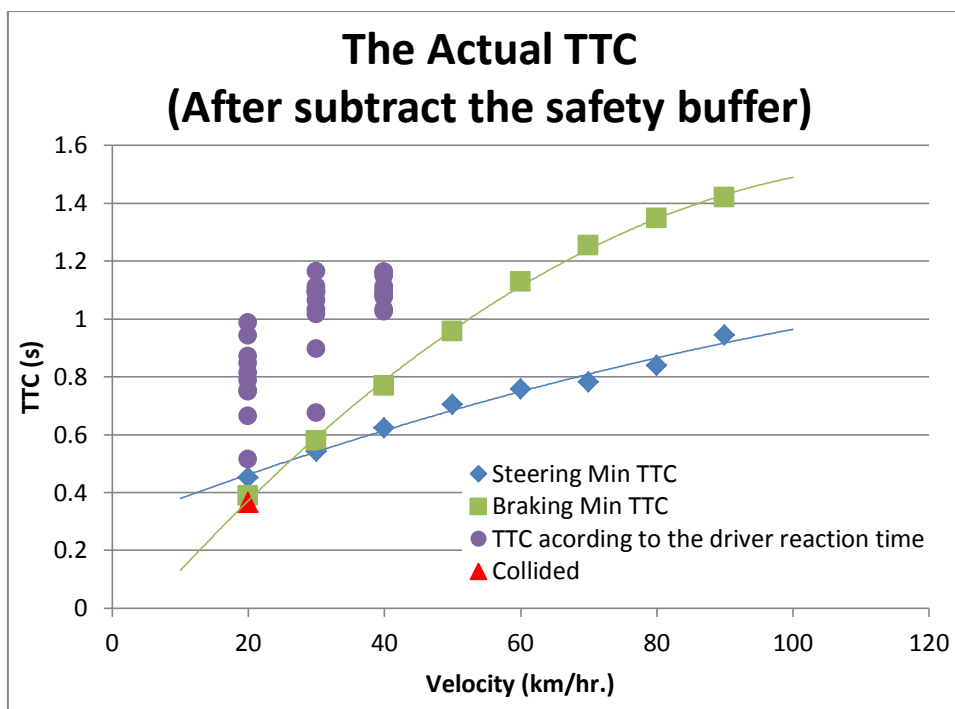
เนื่องจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง ในสภาวะแวดล้อมแบบปิดนั้น ผู้ทดสอบใช้รถยนต์ส่วนตัวที่ติดตั้งระบบเตือนการชนด้านหน้าในการทดสอบ เพื่อความปลอดภัยในการทดสอบ ทำให้ผู้ขับขี่มี ความคุ้นเคยในการควบคุมรถยนต์ทดสอบ ดังนั้น Driver Reaction Time ที่เกิดขึ้นจากการทดสอบด้วยรถยนต์จริงมีค่าน้อยกว่าค่าที่ได้จาก Driver Reaction ที่ได้จาก Driving Simulation สืบเนื่องมาจาก ความคุ้นเคยกับรถยนต์ที่ใช้ในการทดสอบมากกว่า จากการใช้ Driving Simulator

4.4 เปรียบเทียบ Actual Warning TTC กับ Min TTC Diagram



รูปที่ 63 กราฟเปรียบเทียบค่า TTC ที่ระบบเตือน กับ แผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

จากรูปที่ 63 ค่า TTC ที่ระบบเตือนจริง (รวมจุดจุดสามเหลี่ยมที่เกิดการชนขึ้น) อยู่ในโซน A ของแผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ดังแสดงใน รูปที่ 39 ซึ่งเป็นโซนที่ผู้ขับขี่สามารถใช้การเบรก หรือการเลี้ยวหลบในการหลีกเลี่ยงการชนได้



รูปที่ 64 กราฟเปรียบเทียบค่า TTC ที่เหลือให้ผู้ขับขี่ได้ควบคุมรถจริง กับ แผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน

สืบเนื่องจาก Driver Reaction Time เวลา TTC ที่เหลือให้ผู้ขับขี่ได้ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน จึงน้อยลงตาม Driver Reaction Time ที่เกิดขึ้น และทำให้ตำแหน่งของ TTC อยู่ต่ำกว่าตำแหน่งที่ต้องการ ดังแสดงในรูปที่ 64 จะเห็นว่าค่า TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน หลังจากหักค่า Driver Reaction ที่ได้จากการทดสอบแล้ว ยังอยู่ในโซนที่สามารถหลีกเลี่ยงการชนได้ ยกเว้นค่าที่ได้เมื่อรถยนต์เกิดการชนขึ้น ค่า TTC มีค่าเท่ากับ 0.365 วินาที ซึ่งเป็นตำแหน่งที่อยู่ในบริเวณ D ของแผนภาพ TTC ต่ำสุดของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ซึ่งเป็นบริเวณที่การเบรก และการเลี้ยวหลบ ไม่สามารถหลีกเลี่ยงการชนได้ ซึ่งตรงกับผลที่เกิดจากการทำ Model Simulation

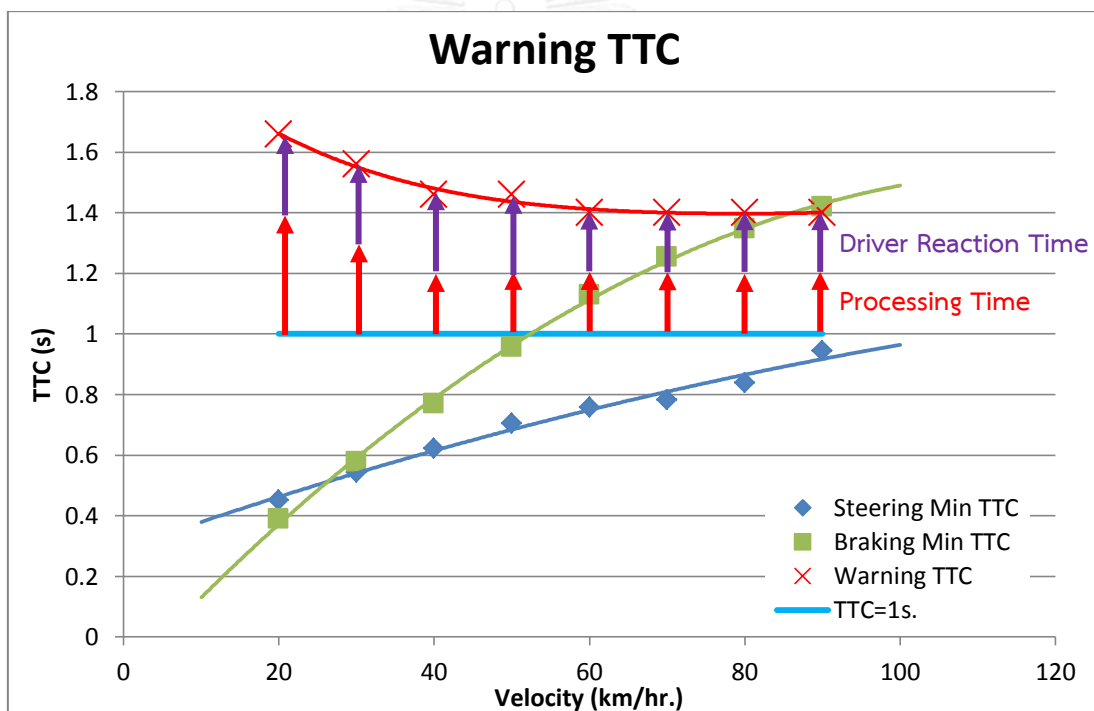
4.5 ตัวอย่างการออกแบบค่า TTC ที่ใช้ในการเตือนผู้ขับขี่

จากรูปที่ 65 จะเห็นว่าถ้าหากต้องการให้ระบบเตือนการชนด้านหน้าเตือนให้ผู้ขับขี่ทำการควบคุมรถยนต์ให้หลีกเลี่ยงการชน โดยให้ผู้ขับขี่มีเวลาที่เหลือให้กระทำการควบคุมรถยนต์ 1 วินาทีพอดี ค่า TTC ที่เตือนผู้ขับขี่ให้ทำการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน ต้องมีค่ามากกว่า 1 วินาที โดยที่ความเร็วต่ำต้องมากกว่า ความเร็วสูง สืบเนื่องมาจาก เวลาในการประมวลผลของระบบ และ

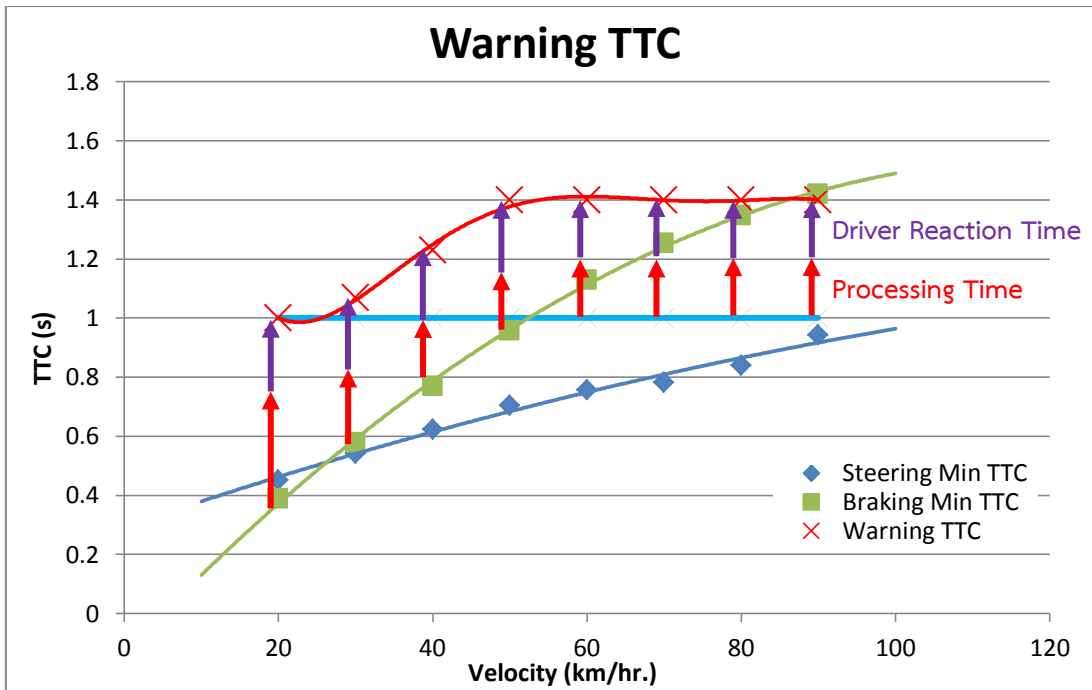
ความล่าช้าเนื่องจาก Driver Reaction Time ซึ่งจะเห็นได้ว่าในความเร็วต่ำยังมีพื้นที่เหลือใต้เส้น $TTC = 1$ วินาที แสดงให้เห็นว่ายังสามารถเตือนผู้ขับขี่ได้ที่ TTC ที่น้อยกว่านี้ โดยยิ่งความเร็วน้อยลง ยิ่งเตือนได้ที่ TTC น้อยลง

โดยรูปที่ 66 แสดงให้เห็นถึงตัวอย่างการกำหนดค่า TTC ที่ใช้ในการเตือนในกรณีนี้ ค่า TTC ที่ใช้ในการเตือนแปรผันกับความเร็ว ซึ่งในตัวอย่างพิจารณาให้การหลีกเลี่ยงการชนด้วยวิธีการเบรกเตือนผู้ใช้ที่ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์ใกล้กับเส้น TTC ต่ำสุดมากที่สุด ส่วนการหลีกเลี่ยงการชนด้วยการเลี้ยวหลบ ออกแบบให้เหลือเวลาให้ผู้ขับขี่ทำการควบคุมรถที่ TTC 1 วินาที

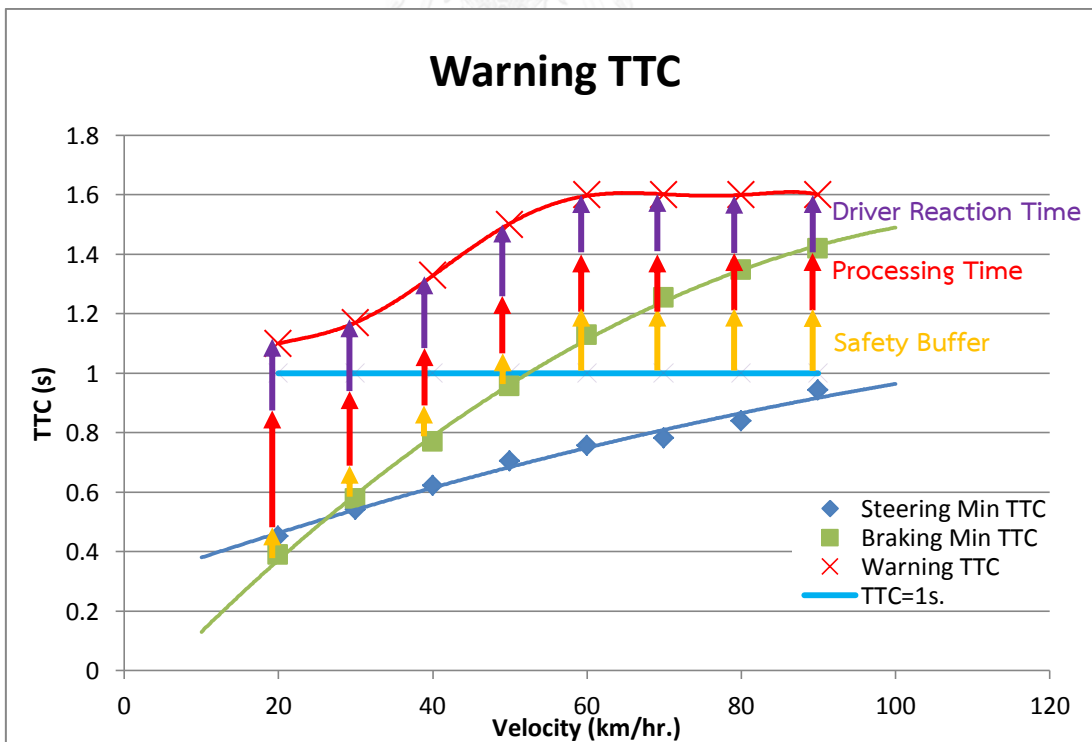
ส่วนในรูปที่ 67 เป็นการแสดงตัวอย่าง TTC ที่ระบบต้องเตือนโดยใช้หลักการการออกแบบ เช่นเดียวกับรูปที่ 66 แต่มีการเพิ่มการ Buffer เข้าไปเพื่อความปลอดภัยโดยค่า Buffer ในการ เบรก เพื่อหลีกเลี่ยงการชนมีค่าน้อยกว่าค่า Buffer ของการเลี้ยวหลบเพื่อหลีกเลี่ยงการชน



รูปที่ 65 แสดงค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนมีค่า 1 วินาที



รูปที่ 66 ตัวอย่างค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนแปรผันตามความเร็ว



รูปที่ 67 ตัวอย่างค่า TTC ที่ระบบต้องเตือนเมื่อต้องการให้ TTC ที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนแปรผันตามความเร็ว

ค่าความล่าช้าเฉลี่ยจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริงสามารถถูกนำไปใช้ในการชดเชยค่า TTC ที่ต้องเตือนผู้ขับขี่เพื่อหลีกเลี่ยงการชน เมื่อนำระบบเตือนการชนด้านหน้าไปใช้งานกับรถยนต์จริง โดยค่า TTC ที่เตือนผู้ขับขี่เพื่อหลีกเลี่ยงการชน ต้องถูกชดเชยด้วยค่า Average Driver Reaction Time และ Average Processing Time ทั้งสองค่า โดยมีเงื่อนไขว่าค่าที่จะนำไปชดเชยต้องคำนึงถึงความเร็ว และวิธีการที่ใช้ในการหลีกเลี่ยงการชนดังที่แสดงในตารางที่ 6

ตารางที่ 6 ค่าความล่าช้าเฉลี่ยจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง

Velocity (km/hr)	Steering		Braking	
	Average Driver Reaction Time (s)	Average Processing Time (s)	Average Driver Reaction Time (s)	Average Processing Time (s)
20	0.2	0.4	0.25	0.4
30	0.2	0.3	0.25	0.3
40	0.2	0.2	0.25	0.2
50	0.2	0.2	0.25	0.2
60	0.2	0.2	0.25	0.2
70	0.2	0.2	0.25	0.2
80	0.2	0.2	0.25	0.2
90	0.2	0.2	0.25	0.2
100	0.2	0.2	0.25	0.2

บทที่ 5

สรุปผล และข้อเสนอแนะ

5.1 สรุปผลงานวิจัย

งานวิจัยนี้ได้ทำการออกแบบระบบตรวจจับรถยนต์ด้านหน้าด้วยวิธีการ Image Processing โดยการรับภาพภายนอกด้วยกล้องสเตอริโอ ผ่านการทำแผนภาพ Disparity เพื่อใช้ในการหาระยะ และจำแนกวัตถุออกจากผิวถนนด้วยแผนภาพ U-V Disparity หลังจากนั้นทำการยืนยันตัววัตถุว่าเป็นรถยนต์หรือไม่ผ่านกระบวนการ Cascade Classification ที่เรียกว่า Haar-like Feature ผ่านการทำ Machine Learning เพื่อทำการตรวจจับรถยนต์ และระยะห่างของรถยนต์ด้านหน้า แล้วนำมาคำนวณหาค่า TTC เพื่อแนะนำให้ผู้ขับขี่ทำการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน

งานวิจัยนี้ใช้หลักการของแบบจำลองทางพลศาสตร์ยานยนต์ เพื่อคำนวณหาเส้นทางการเคลื่อนที่ของรถยนต์ในขณะที่ผู้ขับขี่ทำการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนด้วยการเบรก หรือการเลี้ยวหลบ โดยแบบจำลองทางพลศาสตร์ยานยนต์ที่ใช้คือ 3Degree of Freedom Full Body Vehicle Dynamics Model โดยรับค่าแรงกระทำทิศ Longitudinal และ Lateral มาจาก Tire Model ที่มีการจำลองพฤติกรรมความหน่วงเนื่องจากการเสียดรูปของยางชื่อว่า Relaxation Length Tire Model โดยมี Wheel Model ในการจำลองพฤติกรรมการไถล และมุม Side Slip ของล้อซึ่งมีความจำเป็นในการสร้างแรงจาก Relaxation Length Tire Model นอกจากนี้ยังมี Driver Model เพื่อจำลองพฤติกรรมการเลี้ยวเพื่อเปลี่ยนช่องทางจราจรของผู้ขับขี่ ซึ่งมีลักษณะเช่นเดียวกับการเลี้ยวเพื่อหลีกเลี่ยงการชน นอกจากการคำนวณหาเส้นทางการเคลื่อนที่ของรถยนต์แล้ว แบบจำลองทางพลศาสตร์ยานยนต์ยังสามารถนำมาใช้ในการสร้าง แผนภาพค่า TTC ต่ำสุดที่ผู้ขับขี่สามารถหลีกเลี่ยงการชนได้ด้วยการเบรก และการเลี้ยวหลบ ที่ความเร็วต่าง ๆ

หลังจากการคำนวณหาเส้นทางการเคลื่อนที่ของรถยนต์ เพื่อหาค่า TTC ต่ำสุดที่ผู้ขับขี่สามารถหลีกเลี่ยงการชนได้ด้วยการเบรก และการเลี้ยวหลบ Driving Simulator ได้ถูกนำมาใช้เพื่อศึกษาผลการตอบสนองของผู้ขับขี่ และความเป็นไปได้ในการหลีกเลี่ยงการชน ด้วยการเตือนที่ TTC 1 วินาที ที่ความเร็ว 20 ถึง 100 กิโลเมตรต่อชั่วโมง ผลการศึกษาแสดงให้เห็นว่าผู้ขับขี่มีเวลาในการตอบสนองต่อการเตือนที่ต่างกันเมื่อ เบรก และเลี้ยวหลบ เพื่อหลีกเลี่ยงการชน แต่ในการหลีกเลี่ยงการชนด้วยวิธีเดียวกัน เวลาในการตอบสนองของผู้ขับขี่มีค่าคงที่ที่ความเร็วต่างกัน

จากการทำ Model Evaluation ผลจากการทดสอบแสดงให้เห็นว่าเส้นทางที่เกิดขึ้นในช่วงแรกที่เป็นช่วงที่รถยนต์เคลื่อนที่เพื่อหลบวัตถุที่มีความใกล้เคียงกันมาก ดังนั้น Vehicle Dynamics

Model ที่ใช้ในการ Simulation เพื่อหาเส้นทางการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน สามารถนำมาใช้คำนวณเพื่อหาเส้นทางการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชนโดยผู้ขับขี่ได้ ซึ่งจะถูกนำมาคำนวณ และสร้างแผนภาพ TTC ต่ำสุด ของการเบรก และเลี้ยวหลบเพื่อหลีกเลี่ยงการชน ที่ความเร็วต่าง ๆ เพื่อนำมาใช้เป็นเกณฑ์ในการเตือนผู้ขับขี่ เมื่อระบบถูกนำมาติดตั้งในรถยนต์จริงได้

ผลจากการทดสอบระบบเตือนการชนด้านหน้าด้วยรถยนต์จริง สถานที่ทดสอบที่มีสภาวะแบบปิด ที่ความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง แสดงให้เห็นว่าระบบเตือนการชนด้านหน้าสามารถใช้งานกับรถยนต์จริงได้ โดยมีอัตราการเตือนอยู่ที่ 96.77 เปอร์เซ็นต์ โดยข้อผิดพลาดเกิดจากความล่าช้าเนื่องจากเวลาในการประมวลผล ในขณะที่ความสามารถในการหลีกเลี่ยงการชนตรงกับการใช้แผนภาพ TTC ต่ำสุด โดยมีความล่าช้าเนื่องจาก Driver Reaction Time ซึ่งมีแนวโน้มเช่นเดียวกับ ความล่าช้าเนื่องจาก Driver Reaction Time ที่ทดสอบด้วย Driving Simulator ซึ่งค่าความล่าช้าในการเบรกมากกว่าการเลี้ยวหลบ และในการหลีกเลี่ยงการชนด้วยวิธีเดียวกัน เวลาในการตอบสนองของผู้ขับขี่มีค่าคงที่ที่ความเร็วต่างกัน นอกจากนั้น ค่า TTC ที่ระบบเตือนผู้ขับขี่จากการทดสอบด้วยรถยนต์จริง มีค่าน้อยกว่าที่กำหนดมากขึ้นที่ความเร็วน้อยลงสืบเนื่องมาจากเวลาในการประมวลผลของระบบเตือนการชนด้านหน้า ซึ่งค่าความล่าช้าที่กล่าวมานั้นส่งผลให้เวลาที่เหลือให้ผู้ขับขี่ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน มีค่าน้อยกว่าที่ต้องการ และระบบหลีกเลี่ยงการชนสามารถใช้งานได้ที่ความเร็วสูงสุด 40 กิโลเมตรต่อชั่วโมงในสภาพแวดล้อมของการทดสอบ สืบเนื่องมาจากที่ระบบไม่สามารถแยกฉากด้านข้างออกจากเป้าหมาย และฉากหลังได้ จึงทำให้ระบบไม่สามารถแยกตัวเป้าหมายออกจากฉากหลังในกระบวนการทำแผนภาพ Disparity ได้

ในการนำระบบเตือนการชนด้านหน้าไปใช้งานในรถยนต์จริง ค่า TTC ที่ใช้ในการเตือนผู้ขับขี่ให้ทำการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนต้องทำการชดเชยด้วยค่าความล่าช้าเนื่องจากเวลาในการประมวลผล และความล่าช้าเนื่องจาก Driver Reaction Time ซึ่งต้องชดเชยค่าความล่าช้าเนื่องจากเวลาในการประมวลผลที่ความเร็วต่ำมากกว่า ที่ความเร็วสูง และต้องชดเชยค่าความล่าช้าเนื่องจาก Driver Reaction Time ขณะเบรกเพื่อหลีกเลี่ยงการชน มากกว่าขณะเลี้ยวหลบเพื่อหลีกเลี่ยงการชน โดยถ้าต้องการให้ผู้ขับขี่มีเวลาเหลือในการเลี้ยวหลบที่ 1 วินาที ตามค่า TTC ที่คำนวณได้จากสมการที่ (22) ระบบต้องเตือนผู้ขับขี่ให้ควบคุมรถยนต์เพื่อหลีกเลี่ยงการชนที่ TTC 1.7 วินาที ถ้าต้องการเตือนผู้ขับขี่ที่ TTC คงที่ 1 วินาที หรือถ้าต้องการเปลี่ยนแปลงค่า TTC ที่ใช้เตือนผู้ขับขี่ตามความเร็วรถยนต์ ต้องชดเชย TTC 0.7 วินาทีจากค่า TTC ต่ำสุดที่ผู้ขับขี่สามารถหลีกเลี่ยงการชนได้ด้วยการเบรกในการเตือนผู้ขับขี่ให้หลีกเลี่ยงการชนด้วยการเบรก และต้องชดเชย TTC 0.7 วินาที จากค่า TTC คงที่ 1 วินาที ในการเตือนผู้ขับขี่ให้หลีกเลี่ยงการชนด้วยการเลี้ยวหลบ

5.2 ข้อเสนอแนะ

กระบวนการ Image Processing ควรถูกปรับปรุงให้มีความเร็ว และมีความแม่นยำมากขึ้น โดยระบบในวิทยานิพนธ์นี้ ถูกจำกัดด้านความเร็วสืบเนื่องมาจากความเร็วในการประมวลผลด้าน Image Processing ในส่วนของการทำแผนภาพ Disparity และ Cascade Detection นอกจากนี้ การทำ Machine ให้สามารถตรวจจับวัตถุให้มีความแม่นยำมากขึ้น และตรวจจับวัตถุที่มีลักษณะแตกต่างกันมากขึ้น เป็นข้อจำเป็นในการนำระบบเตือนการชนด้านหน้าด้วยเช่นกัน

ในงานวิจัยนี้ทดสอบระบบด้วยรถยนต์จริงที่ความเร็ว 20, 30 และ 40 กิโลเมตรต่อชั่วโมง สืบเนื่องมาจากเหตุผลทางด้านความปลอดภัย และความสามารถในการประมวลผลของระบบ การศึกษาในอนาคตควรมีการออกแบบการทดลองเพื่อให้สามารถทดสอบที่ความเร็วที่สูงขึ้นได้ นอกจากนั้น คุณสมบัติของอุปกรณ์ในการทดสอบควรถูกพิจารณาเรื่องความถี่ในการเก็บข้อมูล ให้สอดคล้องกับข้อมูลที่เกิดขึ้นขณะทำการทดลอง นอกจากนั้นความสามารถในการระบุตำแหน่ง หรือระยะห่างระหว่างรถยนต์ กับวัตถุควรถูกพิจารณาด้วยเช่นกันเพื่อที่จะทำให้สามารถทราบถึงความสามารถในการคำนวณตำแหน่ง ระยะห่าง และ TTC ของรถยนต์กับวัตถุได้

ในวิทยานิพนธ์นี้ไม่ได้คำนึงถึงพฤติกรรมของผู้ขับขี่ที่มีความแตกต่างกันในผู้ขับขี่แต่ละคน เช่น พฤติกรรมการขับขี่ของผู้ขับขี่ ลักษณะการควบคุมรถยนต์เพื่อหลีกเลี่ยงการชน และความล่าช้าเนื่องจาก Driver Reaction Time ที่แตกต่างกัน เป็นต้น เมื่อปัจจัยเหล่านี้ควรถูกพิจารณาในการทำงานวิจัยในอนาคต จะทำให้ระบบสามารถใช้งานได้กับผู้ขับขี่ได้ในวงที่กว้างขึ้น รวมถึงความแตกต่างของสภาพแวดล้อมภายนอกรถยนต์ เช่น สัมประสิทธิ์แรงเสียดทานที่ต่างกันบนถนนแต่ละประเภท ที่ส่งผลต่อความสามารถในการหลีกเลี่ยงการชนของรถยนต์ด้วย

รายการอ้างอิง

1. Organization, W.H., *Global status report on road safety 2015*. 2015.
2. Sattrawut Ponboon, et al., *Contributing Factors of Road Crashes in Thailand: Evidences from the Accident In-Depth Study*. Journal of the Eastern Asia Society for Transportation Studies 2010. **8**.
3. Wassim G. Najm, et al., *Description of light-vehicle pre-crash scenarios for safety application based on vehicle-to-vehicle communication*. 2013: USA.
4. AShinichiro Horiuchi, et al., *Optimal steering and braking control in emergency obstacle avoidance*. Transactions of the japan society of mechanical engineers series C, 2006. **72**: p. 3250-3255.
5. Ryuzo Hayashi, et al., *Autonomous collision avoidance system by combined control of steering and braking using geometrically optimized vehicular trajectory*. Vehicle system dynamics: international journal of vehicle mechanics and mobility, 2012. **50**: p. 151-168.
6. R. Isermann, M. Schorn, and U. Stahlin, *Anticollision system PRORETA with automatic braking and steering*. Vehicle system dynamics, 2009. **46**: p. 683-694.
7. A. Mukhtar, L. Xia, and T.B. Tang, *Vehicle detection techniques for collision avoidance systems: a review*. IEEE Transactions on intelligent transportation systems, 2015. **16**(5): p. 2318-2338.
8. R. Labayrade, D. Aubert, and L. P. Tarel, *Real time obstacle detection in stereovision on non flat road geometry through "V-Disparity" representation*, in *Intelligent vehicle symposium*. 2002, IEEE. p. 646-651.
9. Z. Hu and K. Uchimura. *U-V-Disparity: an efficient algorithm for Stereovision based scene analysis*. in *Intelligent vehicles symposium*. 2005. IEEE.
10. X. Wen and Yuhui Zheng, *An improved algorithm based on AdaBoost for vehicle recognition*, in *The 2nd international conference on information science and engineering*. 2010: Hangzhou, China. p. 981-984.

11. Jernej Mrovlje and Damir Vrancic, *Distance measuring based on stereoscopic pictures*, in *9th international PhD workshop on systems and control: young generation viewpoint*. 2008.
12. Min Zhang, et al., *An obstacle detection algorithm based on U-V disparity map analysis*, in *International conference on information theory and information security (ICITIS)*. 2010, IEEE: Beijing. p. 763-766.
13. Rainer Lienhart, A.K., Vadim Pisarevsky, *Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection*, in *Joint Pattern Recognition Symposium*. 2003: Magdeburg, Germany. p. 297-304.
14. Yoav Freund, R.E.S., *A Decision-Theoretic Generalization of on-Line Learning and an Application to Boosting* *Journal of Computer and System Sciences*, 1995. **55**(1): p. 119-139.
15. Shivani Agarwal, A.A., and Dan Roth, *Learning to detect objects in images via a sparse, part-based representation*. . *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004. **26**(11): p. 1475-1490.
16. Shivani Agarwal and Dan Roth. *Learning a sparse representation for object detection*. in *he Seventh European Conference on Computer Vision*. 2002. Copenhagen, Denmark.
17. G. Overett, L.P., N. Brewer, L. Andersson and N. Pettersson. *A New Pedestrian Dataset for Supervised Learning*. in *IEEE Intelligent Vehicles Symposium*. 2008.
18. Masato, A., *Vehicle Handling Dynamics: Theory and Application*. 2009, Oxford, UK.: Butterworth-Heinemann,.
19. Shiang Lung, K., T. Han-Shue, and Y. Masayoshi. *Impact of Tire Compliance Behavior to Vehicle Longitudinal Dynamics and Control*. in *The 2007 American Control Conference*. 2007. New York, USA.
20. Shiang-Lung, K., T. Han-Shue, and Y. Masayoshi. *An Improved Tire Model for Vehicle Lateral Dynamics and Control*. in *The 2006 American Control Conference*. 2006. Minnesota, USA.
21. Katja, V., *A Comparison of Headway and Time-to-Collision as Safety Indicators*., *Accident Analysis and Prevention*, 2002. **35**: p. 427-433.

22. V. Lertsilpachalearn, et al. *Study on Driving Behavior in Follow State Driving Condition in Bangkok*, in *The Second TSME International Conference on Mechanical Engineering*,. 2011. Krabi, Thailand.
23. Sarawut, J. and N. Sorawit. *Assessment of Area Traffic Control System in Bangkok by The Microscopic Simulation Model*,. in *The Eastern Asia Society for Transportation Studies*,. 2005.
24. S. Larbwisuthisaroj and N. Noomwongs. *The Vehicle Steering and Braking Guidance using Vehicle Dynamics Approach for Forward Collision Warning System*. in *9th Atrans Symposium on Transportation for a Better Life: Safe and Smart City*,. 2016. Bangkok, Thailand.
25. Ngersukphaiboon, T., S. Chantranuwathaan, and N. Noomwongs. *A Study on Car Following and Cognitive Ability of Elderly Drivers by Using Driving Simulator*. in *12th International Conference on Automotive Engineering*,. 2016. Nonthaburi, Thailand.
26. Hoe C. Lee, Don Cameron, and Andy H. Lee, *Assessing the driving performance of older adult drivers: on-road versus simulated driving*,. *Accident Analysis & Prevention*,, 2003. **35**(5): p. 797-803.
27. S. Larbwisuthisaroj and N. Noomwongs. *Study on The Effect of Driver Reaction Times to the Collision Possibilit and Evasive Ability due to The Forward Collision Warning System using Driving Simulator*. in *The 7th TSME International Conference on Mechanical Engineering*,. 2016. Chaing Mai, Thailand.



ภาคผนวก

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



ภาคผนวก ก

อุปกรณ์ที่ใช้ในการทดสอบ

จุฬาลงกรณ์มหาวิทยาลัย

CHULALONGKORN UNIVERSITY

1. กล้อง IDS 1220LE (รูปที่ 68) ใช้ในการตรวจจับสภาพแวดล้อมเพื่อใช้ในการประมวลผลร่วมกับเลนส์ที่มีระยะโฟกัส 8 มิลลิเมตร



รูปที่ 68 กล้อง IDS 1220LE พร้อมเลนส์ที่มีระยะโฟกัส 8 มิลลิเมตร

ตารางที่ 7 คุณสมบัติของกล้อง IDS 1220LE

Sensor Specification	Detail
Sensor type	CMOS
Shutter system	Electronic global shutter
Characteristic	Linear with knee point
Readout Mode	Progressive scan
Resolution class	WVGA
Resolution	0.36 MPix
Resolution (w x v)	752 x 480 pixels
Aspect ratio	5:4
ADC	10 bits
Optical sensor class	1/3 inch
Optical area	4.512 mm x 2.880 mm
Optical sensor diagonal	5.35 mm (1/2.99 inch)
Pixel size	6.00 μ m, square
Micro lens shift	0°
Manufacturer	Aptina
Sensor name, monochrome	MT9V032C12STM
Sensor name, color	MT9V032C12STC

2. คอมพิวเตอร์ (รูปที่ 69) ใช้สำหรับประมวลผลระบบเตือนการชนท้าย



รูปที่ 69 คอมพิวเตอร์ที่ใช้ในการประมวลผล

ตารางที่ 8 คุณสมบัติของคอมพิวเตอร์ที่ใช้ในการประมวลผล

Specification	Detail
Manufacturer	Fujitsu
Model	LH531
CPU	Intel Core i5 2340M/ 2.4Ghz
Ram	2x4GB/1333 MHz
Display Resolution	1366 x 768
HDD	500 GB 5400 rpm
VGA	Intel HD Graphics 3000 / NVIDIA GeForce 410M
Operatioin System	Window 7 64 bits

3. Microcontroller Arduino Uno (รูปที่ 70) ใช้สำหรับรับข้อมูลจากคอมพิวเตอร์เพื่อเตือนผู้ขับขี่



รูปที่ 70 Microcontroller Arduino Uno

ตารางที่ 9 คุณสมบัติของ Microcontroller Arduino Uno

Specification	Detail
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (Limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

4. Neo Pixel LED Stick (รูปที่ 71) เพื่อใช้ในการเตือนผู้ขับขี่



รูปที่ 71 Neo Pixel LED Stick

5. กล้อง GoPro Hero 5 Black (รูปที่ 72) ใช้สำหรับบันทึกวิธีการทดสอบ นอกจากนี้ยังสามารถใช้หา Driver Reaction Time ได้อีกด้วย



รูปที่ 72 กล้อง GoPro Hero 5 Black

ตารางที่ 10 คุณสมบัติของกล้อง Gopro Hero 5 Black

Specification	Detail
Weight	118 g with battery; 90g without battery
Dimension	62 mm x 44.6 mm x 33.7 mm
Battery	Removable Li-ion or LiPolymer Capacity: 1220mAh, 3.85V, 4636 mWhr
Lens	6-element aspherical all-glass lens
Ports	USB type C HDMI
Storage	MicroSD card minimum class 10 or UHS-1 up to 64 capacity
Audio	Internal mono microphone AAC (advanced audio codec) compression with AGC (auto gain control)
LCD	Touch interface Sleep/ wake and on/off setting
WiFi / Bluetooth	802.11 b/g/n/a/ac WLAN + BT4.1 + BLE

6. รถยนต์ทดสอบ (รูปที่ 73) ใช้สำหรับติดตั้งระบบเตือนการชนท้ายเพื่อทดสอบ



รูปที่ 73 รถยนต์ที่ใช้ในการทดสอบ

7. ภาพรถยนต์จำลอง (รูปที่ 74) ภาพจำลองรถยนต์ ขนาด 2 x 2 เมตร ทำจากโพลีเอทิลีนสำหรับเป็นเป้าในการทดลอง เพื่อความปลอดภัย



รูปที่ 74 ภาพรถยนต์จำลอง

8. GPS พร้อมเสาอากาศ (รูปที่ 75) ใช้ในการเก็บข้อมูล ตำแหน่ง และความเร็วของรถยนต์



รูปที่ 75 GPS พร้อมเสาอากาศ

ตารางที่ 11 คุณสมบัติ GPS ที่ใช้ในการเก็บข้อมูล ตำแหน่ง และความเร็วของรถยนต์

Specification	Detail
Chip	u-blox UBX-M8030
Frequency	L1, 1575.42MHz
Protocol	NMEA 0183 v2.3 GALILEO L1 open service (with upgrade) Default: GGA,GSA,GSV,RMC Support: VTG, GLL, TXT u-blox binary and NMEA Command
Available Baud	1200,4800,9600,19200,38400,57600,112500 bps
Channel	56
Internal ROM	3Mbit of ROM and 2Mbit of RAM
Sensitivity	Tracking:-164dBm Capturing:-159dBm Cold Start:-147dBm
Cold Start	26 seconds on average
Warm Start	24 seconds on average
Hot Start	1 second on average
Accuracy	Horizontal Position: Autonomous<2.5m on average, SBAS < 2.0m on average Velocity: 0.1 m/s Time pulse signal: RMS 30 ns
Sampling Rate	5 Hz
Max Height	50000 meters
Max Speed	515 m/s
Max Accelerate	<= 4G



ภาคผนวก ข
ระบบเตือนการชนด้านหน้า

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ระบบเตือนการชนด้านหน้าถูกพัฒนาขึ้นด้วยภาษา C# และใช้ Library Image Processing ของ EMguCV ซึ่งพัฒนามาบนโปรแกรม Microsoft Visual Studio ซึ่งทั้งหมดเป็นโปรแกรม Open Source และ Freeware ที่ไม่มีค่าใช้จ่ายในการนำมาใช้



รูปที่ 76 ตัวอย่างภาพระบบเตือนการชนด้านหน้าขณะใช้งาน

```

using System;
using System.Collections.Generic;
using System.Drawing;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;
using System.IO.Ports;

using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;

namespace Real_All
{
    public partial class Real_All : Form
    {
        private SerialPort myPort;
        private int warningStatus;

        Capture zCapture1;
        Capture zCapture2;
        Image<Bgr, byte> zLImg = new Image<Bgr, byte>(640, 480);
        Image<Bgr, byte> zRImg = new Image<Bgr, byte>(640, 480);
        Mat zLGray = new Mat();
        Mat zRGray = new Mat();
    }
}

```



```

Mat zLGray2 = new Mat();
Mat zRGray2 = new Mat();
Mat zLGray3 = new Mat();
Mat zRGray3 = new Mat();
Mat zLGrayRec = new Mat();
Mat zRGrayRec = new Mat();
Image<Gray, short> zDisparity = new Image<Gray, short>(640, 480);
Image<Gray, short> zDisparity2 = new Image<Gray, short>(640, 480);

Int32 minDisparity = 0;
Int32 numDisparities = 32;
Int32 SADWindowSize = 5;
Int32 P1 = 5*5*15;
Int32 P2 = 5*5*5*23;
Int32 disp12MaxDiff = 50;
Int32 preFilterCap = 5;
Int32 UniquenessRatio = 0;
Int32 speckleWindowSize = 1;
Int32 speckleRange = 1;

short[, ,] zDispData;
Byte[, ,] zVDispData;
Byte[, ,] zUDispData;
Image<Gray, Byte> zReCreateV, zReCreateV8, zReCreateU, zReCreateU8;

LineSegment2D[] lines;
LineSegment2D[] lines2;
Image<Gray, Byte> lineImage;
Image<Gray, Byte> lineImage2;

int rhoV = 1;
int thetaV = 1;
int thresV = 1;
int minV = 1;
int maxV = 1;

int rhoU = 1;
int thetaU = 1;
int thresU = 1;
int minU = 1;
int maxU = 1;

int zRangeP;

Rectangle zROI = new Rectangle();
bool HgChk = false;
bool HvChk = false;

CascadeClassifier car = new CascadeClassifier("cascade.xml");
Image<Bgr, Byte> zCroppedImg;
List<Rectangle> cars = new List<Rectangle>();

Double zRangeM = 0;
Double zSensorwidth = 3.838;
Double zFocal = 8;
Double zCamBase = 0.25;

Double zTTC = 0;

```

```

Stopwatch watch = Stopwatch.StartNew();

public Real_All()
{
    InitializeComponent();

    warningStatus = 0;
    myPort = new SerialPort();
    myPort.BaudRate = 9600;
    myPort.PortName = "COM3";
    myPort.Open();

    try
    {
        zCapture1 = new Capture(1);
        zCapture2 = new Capture(2);
        zCapture1.Start();
        zCapture2.Start();
        Application.Idle += ProcessFrame;
    }
    catch (NullReferenceException excpt)
    {
        MessageBox.Show(excpt.Message);
    }
}

private void ProcessFrame(object sender, EventArgs arg)
{
    watch.Restart();
    zCapture1.FlipVertical = true;
    zCapture2.FlipVertical = true;
    zCapture1.Retrieve(zLImg);
    zCapture2.Retrieve(zRImg);
    imageBox3.Image = zLImg;
    imageBox4.Image = zRImg;

    Compute_Disparity();

    zDispData = new short[zDisparity.Height, zDisparity.Width, 1];
    zDispData = zDisparity.Data;
    zVDispData = new Byte[zDisparity.Height, numDisparities, 1];
    zUDispData = new Byte[numDisparities, zDisparity.Width, 1];

    Compute_UV();

    zReCreateV = new Image<Gray, Byte>(zVDispData);
    zReCreateV8 = new Image<Gray, Byte>(zReCreateV.Size);
    zReCreateU = new Image<Gray, Byte>(zUDispData);
    zReCreateU8 = new Image<Gray, Byte>(zReCreateU.Size);

    Compute_Canny();
    Compute_HT_V();
    Compute_HT_U();
    HgChk = Compute_ROI();
    zCroppedImg = zLImg.Copy(zROI);
    HvChk = Compute_HV();
}

```

```

        if (HvChk == true)
        {
            zRangeM = Compute_Range();
            label1.Text = HvChk.ToString();
            label2.Text = zRangeM.ToString();
        }
        else
        {
            zRangeM = 0;
            label1.Text = HvChk.ToString();
            label2.Text = "none";
        }

        zTTC = Compute_TTC(zRangeM);
        warning(zTTC);
    }

    public void Compute_Disparity()
    {
        CvInvoke.CvtColor(zLImg, zLGray, ColorConversion.Rgb2Gray);
        CvInvoke.CvtColor(zRImg, zRGray, ColorConversion.Rgb2Gray);

        CvInvoke.GaussianBlur(zLGray, zLGray2, new Size(0, 0), 4);
        CvInvoke.GaussianBlur(zRGray, zRGray2, new Size(0, 0), 4);

        CvInvoke.EqualizeHist(zLGray2, zLGray3);
        CvInvoke.EqualizeHist(zRGray2, zRGray3);

        using (StereoSGBM stereoSolver = new StereoSGBM(minDisparity,
            numDisparities, SADWindowSize, P1, P2, disp12MaxDiff, preFilterCap,
            UniquenessRatio, speckleWindowSize, speckleRange, 0))
        { stereoSolver.Compute(zLGray3, zRGray3, zDisparity); }
    }

    public void Compute_UV()
    {
        Parallel.For(0, zDisparity.Height, i =>
        {
            Parallel.For(0, zDisparity.Width, k =>
            {
                int d = Math.Abs(zDispData[i, k, 0] / 16);
                zVDispData[i, d, 0]++;
                zUDispData[d, k, 0]++;
            });
        });

        Parallel.For(0, 10, i =>
        {
            Parallel.For(0, zDisparity.Width, k =>
            {
                zUDispData[i, k, 0] = 0;
            });
            Parallel.For(0, zDisparity.Height, k =>
            {
                zVDispData[k, i, 0] = 0;
            });
        });
    }
}

```

```

public void Compute_Canny()
{
    zReCreateV8 = zReCreateV.Canny(130, 170);
    zReCreateU8 = zReCreateU.Canny(60, 100);
}

public void Compute_HT_V()
{
    lines = CvInvoke.HoughLinesP(zReCreateV8, rhoV, Math.PI /
thetaV, thresV, minV, maxV);
    lineImage = new Image<Gray, Byte>(zReCreateV8.Size);
    foreach (LineSegment2D line in lines)
    {
        lineImage.Draw(line, new Gray(255), 2);
    }
}

public void Compute_HT_U()
{
    lines2 = CvInvoke.HoughLinesP(zReCreateU8, rhoU, Math.PI /
thetaU, thresU, minU, maxU);
    lineImage2 = new Image<Gray, Byte>(zReCreateU8.Size);
    foreach (LineSegment2D line in lines2)
    {
        lineImage2.Draw(line, new Gray(255), 2);
    }
}

public bool Compute_ROI()
{
    bool chk1 = false;
    bool chk2 = false;
    //----P1----//
    int zLineCnt1 = 0;
    foreach (LineSegment2D line in lines)
    {
        foreach (LineSegment2D line2 in lines2)
        {
            int zDispValV = line.P1.X;
            int zDispValU = line2.P1.Y;
            if (zDispValV == zDispValU)
            {
                zLineCnt1++;
            }
        }
    }

    int[] P1X = new int[zLineCnt1];
    int[] P1Y = new int[zLineCnt1];
    int[] zDispRange = new int[zLineCnt1];

    if (zLineCnt1 != 0)
    {
        chk1 = true;
        zLineCnt1 = 0;
        foreach (LineSegment2D line in lines)
        {
            foreach (LineSegment2D line2 in lines2)

```

```

        int zDispValV = line.P1.X;
        int zDispValU = line2.P1.Y;
        if (zDispValV == zDispValU)
        {
            zDispRange[zLineCnt1] = zDispValV;
            P1Y.SetValue(line.P1.Y, zLineCnt1);
            P1X.SetValue(line2.P1.X, zLineCnt1);
            zLineCnt1++;
        }
    }
}
else
{
    chk1 = false;
}

Console.WriteLine(zLineCnt1);
//----P2----//

int zLineCnt2 = 0;
for (int i = 0; i < lines.Length; i++)
{
    for (int j = 0; j < lines2.Length; j++)
    {
        int zDispValV = lines[i].P2.X;
        int zDispValU = lines2[j].P2.Y;
        if (zDispValV == zDispValU)
        {
            zLineCnt2++;
        }
    }
}

int[] P2X = new int[zLineCnt2];
int[] P2Y = new int[zLineCnt2];

if (zLineCnt2 != 0)
{
    chk2 = true;
    zLineCnt2 = 0;
    for (int i = 0; i < lines.Length; i++)
    {
        for (int j = 0; j < lines2.Length; j++)
        {
            int zDispValV = lines[i].P2.X;
            int zDispValU = lines2[j].P2.Y;
            if (zDispValV == zDispValU)
            {
                P2Y.SetValue(lines[i].P2.Y, zLineCnt2);
                P2X.SetValue(lines2[j].P2.X, zLineCnt2);
                zLineCnt2++;
            }
        }
    }
}
}

```

```

else
{
    chk2 = false;
}

zDisparity2 = zDisparity.Clone();

if (chk1 == true && chk2 == true)
{
    int[] Xmax = new int[2] { P1X.Max(), P2X.Max() };
    int[] Xmin = new int[2] { P1X.Min(), P2X.Min() };
    int[] Ymax = new int[2] { P1Y.Max(), P2Y.Max() };
    int[] Ymin = new int[2] { P1Y.Min(), P2Y.Min() };
    zROI = new Rectangle(Xmin.Min(), Ymin.Min(), Xmax.Max() -
Xmin.Min(), Ymax.Max() - Ymin.Min());
    zDisparity2.Draw(zROI, new Gray(255), 2);
    imageBox1.Image = zDisparity2;
    zRangeP = zDispRange.Max();
    return (true);
}
else
{
    imageBox1.Image = zDisparity2;
    return (false);
}
}

public bool Compute_HV()
{
    using (UMat ugray = new UMat())
    {
        CvInvoke.CvtColor(zCroppedImg, ugray,
ColorConversion.Bgr2Gray);
        CvInvoke.EqualizeHist(ugray, ugray);
        Rectangle[] carDetected = car.DetectMultiScale(ugray, 1.1,
10, Size.Empty);
        cars.AddRange(carDetected);
        foreach (Rectangle c in carDetected)
        {
            CvInvoke.Rectangle(zCroppedImg, c, new
Bgr(Color.Red).MCvScalar, 2);
        }
        imageBox2.Image = zCroppedImg;
    }
    if (cars.Count() != 0) { return(true); }[1]
    else { return(false); }
}

public double Compute_Range()
{
    double Range;
    Range = (zCamBase * zRImg.Width) / ((zSensorwidth / zFocal) *
zRangeP);
    return (Range);
}

```

```

public double Compute_TTC(double S)
{
    double TTC = 0;
    double V = 20 / 3.6;
    TTC = S / V;
    return(TTC);
}

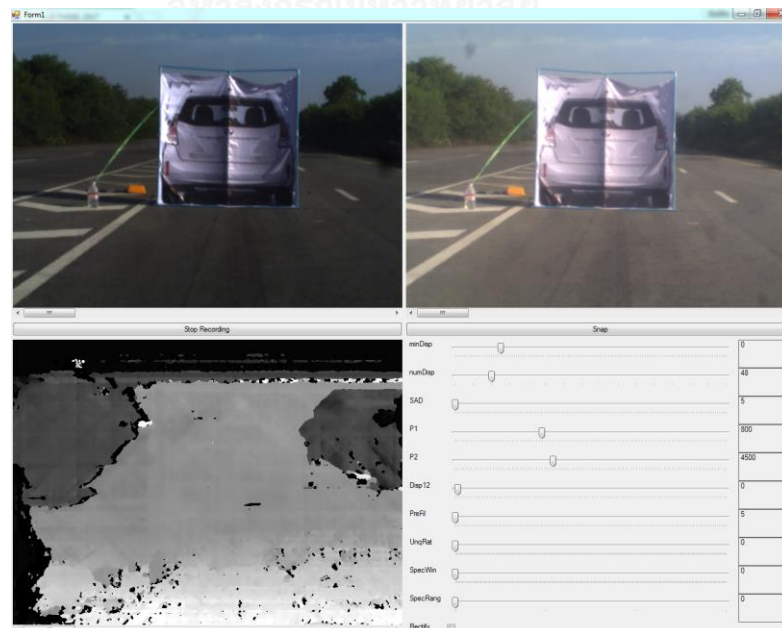
public void warning(double TTC)
{
    double AllTTC = TTC - (watch.ElapsedMilliseconds / 1000);

    if (AllTTC <= 1.94 && warningStatus == 0)
    {
        warningStatus = 1;
    }
    else if (AllTTC >= 2.24 && warningStatus == 1)
    {
        warningStatus = 0;
    }
    Arduino_Com(warningStatus);
}

public void Arduino_Com(int state)
{
    myPort.WriteLine(state.ToString());
}
}
}

```

รูปที่ 77 แสดง Code ที่ใช้ในการพัฒนาระบบ เตือนการชนด้านหน้า



รูปที่ 78 ภาพแสดงโปรแกรมที่ใช้ในการปรับแต่งค่าเพื่อสร้าง Disparity Map

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Diagnostics;

using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;

namespace Stereo_Capture
{
    public partial class Form1 : Form
    {
        private Capture zCapture1;
        private Capture zCapture2;
        private VideoWriter zVideo1;
        private VideoWriter zVideo2;
        private bool zRec;
        Stopwatch watch = new Stopwatch();
        int FileNo = 0;
        private String ImgPath1;
        private String ImgPath2;
        Mat Frame1 = new Mat();
        Mat Frame2 = new Mat();
        Image<Gray, short> zDisparity = new Image<Gray, short>(640, 480);

        Int32 minDisparity = 0;
        Int32 numDisparities = 48;
        Int32 SADWindowSize = 5;
        Int32 P1 = 0;
        Int32 P2 = 0;
        Int32 disp12MaxDiff = 20;
        Int32 preFilterCap = 5;
        Int32 UniquenessRatio = 0;
        Int32 speckleWindowSize = 0;
        Int32 speckleRange = 0;

        public Form1()
        {
            InitializeComponent();

            zminDispBar.Value = minDisparity;
            znumDispBar.Value = numDisparities / 16;
            zSADBar.Value = (SADWindowSize - 5) / 2;
            zP1Bar.Value = (P1/SADWindowSize)/SADWindowSize;
            zP2Bar.Value = (P2 / SADWindowSize) / SADWindowSize;
            zDisp12Bar.Value = disp12MaxDiff;
            zPreFilBar.Value = (preFilterCap - 5) / 2;
            zUnqRatBar.Value = UniquenessRatio;
            zSpecWinBar.Value = speckleWindowSize;
            zSpecRangBar.Value = speckleRange;
        }
    }
}

```



```

zminDispLbl.Text = minDisparity.ToString();
znumDispLbl.Text = numDisparities.ToString();
zSADLbl.Text = SADWindowSize.ToString();
zP1Lbl.Text = P1.ToString();
zP2Lbl.Text = P2.ToString();
zDisp12Lbl.Text = disp12MaxDiff.ToString();
zPreFilLbl.Text = preFilterCap.ToString();
zUnqRatLbl.Text = UniquenessRatio.ToString();
zSpecWinLbl.Text = speckleWindowSize.ToString();
zSpecRangLbl.Text = speckleRange.ToString();

try
{
    zCapture1 = new Capture(1);
    zCapture2 = new Capture(2);
    zCapture1.Start();
    zCapture2.Start();
    zCapture1.ImageGrabbed += ProcessFrame;
    Application.Idle += ProcessFrame;
}
catch (NullReferenceException excpt)
{
    MessageBox.Show(excpt.Message);
}
}

private void Form1_Load(object sender, EventArgs e)
{
    Console.WriteLine(zRec);
    String Path1 = "testL.avi"; //File Name For Left Cam
    String Path2 = "testR.avi"; //File Name For Right Cam
    ImgPath1 = "testL.bmp";
    ImgPath2 = "testR.bmp";
    int Fps = 30;
    Size zSize = new Size(640, 480);
    zVideo1 = new VideoWriter(Path1, Fps, zSize, true);
    zVideo2 = new VideoWriter(Path2, Fps, zSize, true);
    zRec = false;
}

private void ProcessFrame(object sender, EventArgs arg)
{
    zCapture1.FlipVertical = true;
    zCapture2.FlipVertical = true;
    zCapture1.Retrieve(Frame1);
    zCapture2.Retrieve(Frame2);
    if (zRec == true)
    {
        zVideo1.Write(Frame1);
        zVideo2.Write(Frame2);
    }
    imageBox1.Image = Frame1;
    imageBox2.Image = Frame2;
    Compute_Dispatch();
}

```

```

private void button1_Click(object sender, EventArgs e)
{
    if (zRec == false)
    {
        zVideo1.Dispose();
        zVideo2.Dispose();
        zRec = !zRec;
        button1.Text = "Stop Recording";
    }
    else
    {
        zRec = !zRec;
    }
}

private void button2_Click(object sender, EventArgs e)
{
    string ImgPathNew1 = ImgPath1.Insert(5, FileNo.ToString());
    string ImgPathNew2 = ImgPath2.Insert(5, FileNo.ToString());
    Frame1.Save(ImgPathNew1);
    Frame2.Save(ImgPathNew2);
    FileNo++;
}

private void zminDispBar_Scroll(object sender, EventArgs e)
{
    minDisparity = zminDispBar.Value;
    zminDisplbl.Text = minDisparity.ToString();
    Compute_Disp();
}

private void znumDispBar_Scroll(object sender, EventArgs e)
{
    numDisparities = znumDispBar.Value * 16;
    znumDisplbl.Text = numDisparities.ToString();
    Compute_Disp();
}

private void zSADBar_Scroll(object sender, EventArgs e)
{
    SADWindowSize = (zSADBar.Value * 2) + 5;
    zSADLbl.Text = SADWindowSize.ToString();
    Compute_Disp();
}

private void zP1Bar_Scroll(object sender, EventArgs e)
{
    P1 = zP1Bar.Value * SADWindowSize * SADWindowSize;
    zP1Lbl.Text = P1.ToString();
    Compute_Disp();
}

private void zP2Bar_Scroll(object sender, EventArgs e)
{
    P2 = zP2Bar.Value * SADWindowSize * SADWindowSize *
SADWindowSize;
    zP2Lbl.Text = P2.ToString();
    Compute_Disp();
}

```

```

private void zDisp12Bar_Scroll(object sender, EventArgs e)
{
    disp12MaxDiff = zDisp12Bar.Value;
    zDisp12Lbl.Text = disp12MaxDiff.ToString();
    Compute_Disp();
}

private void zPreFilBar_Scroll(object sender, EventArgs e)
{
    preFilterCap = (zPreFilBar.Value*2)+5;
    zPreFilLbl.Text = preFilterCap.ToString();
    Compute_Disp();
}

private void zUnqRatBar_Scroll(object sender, EventArgs e)
{
    UniquenessRatio = zUnqRatBar.Value;
    zUnqRatLbl.Text = UniquenessRatio.ToString();
    Compute_Disp();
}

private void zSpecWinBar_Scroll(object sender, EventArgs e)
{
    speckleWindowSize = zSpecWinBar.Value;
    zSpecWinLbl.Text = speckleWindowSize.ToString();
    Compute_Disp();
}

private void zSpecRangBar_Scroll(object sender, EventArgs e)
{
    speckleRange = zSpecRangBar.Value;
    zSpecRangLbl.Text = speckleRange.ToString();
    Compute_Disp();
}

void Compute_Disp()
{
    Image<Gray, byte> zLGray = new Image<Gray, byte>(640, 480);
    Image<Gray, byte> zRGray = new Image<Gray, byte>(640, 480);
    Mat zLGray2 = new Mat();
    Mat zRGray2 = new Mat();
    Mat zLGray3 = new Mat();
    Mat zRGray3 = new Mat();

    if (zRecChkBox.Checked == true)
    {
        CvInvoke.CvtColor(Frame1, zLGray,
ColorConversion.Rgb2Gray);
        CvInvoke.CvtColor(Frame2, zRGray,
ColorConversion.Rgb2Gray);
        CvInvoke.GaussianBlur(zLGray, zLGray2, new Size(0, 0), 1);
        CvInvoke.GaussianBlur(zRGray, zRGray2, new Size(0, 0), 1);
        CvInvoke.EqualizeHist(zLGray2, zLGray3);
        CvInvoke.EqualizeHist(zRGray2, zRGray3);

        CvInvoke.Remap(zLGray3, zLGray3, zMap1L, zMap2L, 0);
        CvInvoke.Remap(zRGray3, zRGray3, zMap1R, zMap2R, 0);
    }
}

```

```

        imageBox1.Image = zLGray3;
        imageBox2.Image = zRGray3;
    }
    else if (zRecChkBox.Checked == false)
    {
        CvInvoke.CvtColor(Frame1, zLGray,
ColorConversion.Rgb2Gray);
        CvInvoke.CvtColor(Frame2, zRGray,
ColorConversion.Rgb2Gray);

        CvInvoke.GaussianBlur(zLGray, zLGray2, new Size(0, 0), 4);
        CvInvoke.GaussianBlur(zRGray, zRGray2, new Size(0, 0), 4);

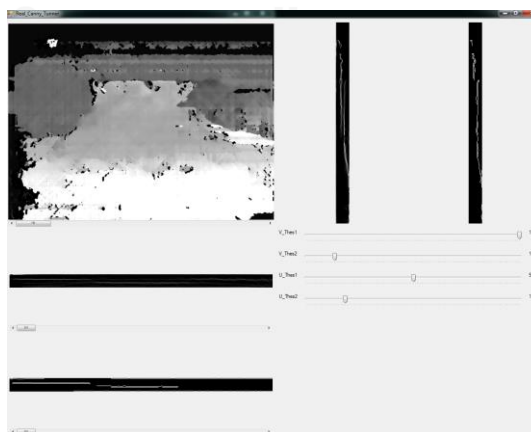
        CvInvoke.EqualizeHist(zLGray2, zLGray3);
        CvInvoke.EqualizeHist(zRGray2, zRGray3);
    }
    using (StereoSGBM stereoSolver = new StereoSGBM(minDisparity,
numDisparities, SADWindowSize, P1, P2, disp12MaxDiff, preFilterCap,
UniquenessRatio, speckleWindowSize, speckleRange, 0))
    { stereoSolver.Compute(zLGray3, zRGray3, zDisparity); }
    zDispImgBox.Image = zDisparity;

    PointF zCenterPt = new PointF(zDisparity.Width / 2,
zDisparity.Height / 2);
    CircleF zCir = new CircleF(zCenterPt, 1);

    zDisparity.Draw(zCir, new Gray(255));
    zDispImgBox.Image = zDisparity;
    }
}
}
}

```

รูปที่ 79 ภาพแสดงCodeโปรแกรมที่ใช้ในการปรับแต่งค่าเพื่อสร้าง Disparity Map



รูปที่ 80 ภาพแสดงโปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Canny Edge Detection

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.Util;

namespace Real_Canny_Tunner
{
    public partial class Canny : Form
    {
        Capture zCapture1;
        Capture zCapture2;

        Image<Bgr, byte> zLImg = new Image<Bgr, byte>(640, 480);
        Image<Bgr, byte> zRImg = new Image<Bgr, byte>(640, 480);
        Mat zLGray = new Mat();
        Mat zRGray = new Mat();
        Mat zLGray2 = new Mat();
        Mat zRGray2 = new Mat();
        Mat zLGray3 = new Mat();
        Mat zRGray3 = new Mat();
        Mat zLGrayRec = new Mat();
        Mat zRGrayRec = new Mat();
        Image<Gray, short> zDisparity = new Image<Gray, short>(640, 480);

        Int32 minDisparity = 0;
        Int32 numDisparities = 32;
        Int32 SADWindowSize = 5;
        Int32 P1 = 800;
        Int32 P2 = 4500;
        Int32 disp12MaxDiff = 20;
        Int32 preFilterCap = 5;
        Int32 UniquenessRatio = 0;
        Int32 speckleWindowSize = 0;
        Int32 speckleRange = 0;
        short[, ] zDispData;
        Byte[, ] zVDispData;
        Byte[, ] zUDispData;
        Image<Gray, Byte> zReCreateV, zReCreateV8, zReCreateU, ReCreateU8;

        int V_Thes1 = 0;
        int V_Thes2 = 0;
        int U_Thes1 = 0;
        int U_Thes2 = 0;

        public Canny()
        {

```

```

        InitializeComponent();
    try
    {
        zCapture1 = new Capture(1);
        zCapture2 = new Capture(2);
        zCapture1.Start();
        zCapture2.Start();
        Application.Idle += ProcessFrame;
    }
    catch (NullReferenceException excpt)
    {
        MessageBox.Show(excpt.Message);
    }
}

private void ProcessFrame(object sender, EventArgs arg)
{
    zCapture1.FlipVertical = true;
    zCapture2.FlipVertical = true;

    zCapture1.Retrieve(zLImg);
    zCapture2.Retrieve(zRImg);

    Compute_Disparity();

    zDispData = new short[zDisparity.Height, zDisparity.Width, 1];
    zDispData = zDisparity.Data;
    zVDispData = new Byte[zDisparity.Height, numDisparities, 1];
    zUDispData = new Byte[numDisparities, zDisparity.Width, 1];

    Compute_UV();

    zReCreateV = new Image<Gray, Byte>(zVDispData);
    zReCreateV8 = new Image<Gray, Byte>(zReCreateV.Size);
    imageBox2.Image = zReCreateV;

    zReCreateU = new Image<Gray, Byte>(zUDispData);
    zReCreateU8 = new Image<Gray, Byte>(zReCreateU.Size);
    imageBox3.Image = zReCreateU;

    Compute_Canny();
}

public void Compute_Disparity()
{
    CvInvoke.CvtColor(zLImg, zLGray, ColorConversion.Rgb2Gray);
    CvInvoke.CvtColor(zRImg, zRGray, ColorConversion.Rgb2Gray);

    CvInvoke.GaussianBlur(zLGray, zLGray2, new Size(0, 0), 4);
    CvInvoke.GaussianBlur(zRGray, zRGray2, new Size(0, 0), 4);

    CvInvoke.EqualizeHist(zLGray2, zLGray3);
    CvInvoke.EqualizeHist(zRGray2, zRGray3);
}

```

```

        using (StereoSGBM stereoSolver = new StereoSGBM(minDisparity,
numDisparities, SADWindowSize, P1, P2, disp12MaxDiff, preFilterCap,
UniquenessRatio, speckleWindowSize, speckleRange, 0))
        { stereoSolver.Compute(zLGray3, zRGray3, zDisparity); }
        imageBox1.Image = zDisparity;
    }
    public void Compute_UV()
    {
        Parallel.For(0, zDisparity.Height, i =>
        {
            Parallel.For(0, zDisparity.Width, k =>
            {
                int d = Math.Abs(zDispData[i, k, 0] / 16);
                zVDispData[i, d, 0]++;
                zUDispData[d, k, 0]++;
            });
        });
    }

    public void Compute_Canny()
    {
        zReCreateV8 = zReCreateV.Canny(V_Thes1, V_Thes2);
        imageBox4.Image = zReCreateV8;
        zReCreateU8 = zReCreateU.Canny(U_Thes1, U_Thes2);
        imageBox5.Image = zReCreateU8;
    }

    private void trackBar1_Scroll(object sender, EventArgs e)
    {
        V_Thes1 = trackBar1.Value*10;
        V1.Text = V_Thes1.ToString();
        Compute_Canny();
    }

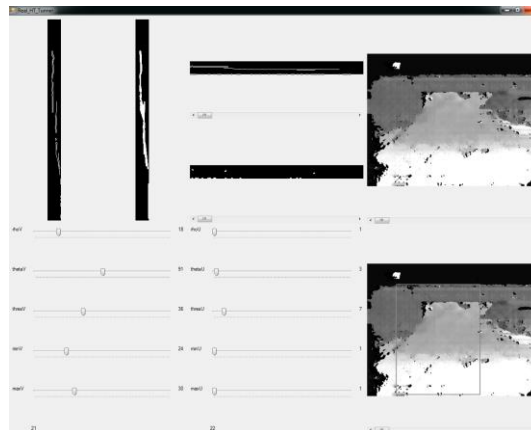
    private void trackBar2_Scroll(object sender, EventArgs e)
    {
        V_Thes2 = trackBar2.Value * 10;
        V2.Text = V_Thes2.ToString();
        Compute_Canny();
    }

    private void trackBar3_Scroll(object sender, EventArgs e)
    {
        U_Thes1 = trackBar3.Value * 10;
        U1.Text = U_Thes1.ToString();
        Compute_Canny();
    }

    private void trackBar4_Scroll(object sender, EventArgs e)
    {
        U_Thes2 = trackBar4.Value * 10;
        U2.Text = U_Thes2.ToString();
        Compute_Canny();
    }
}
}

```

รูปที่ 81 ภาพแสดง Code โปรแกรมที่ใช้ในการปรับแต่ค่าในการทำ Canny Edge Detection



รูปที่ 82 ภาพแสดง โปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Hough Transformation

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.Util;

namespace Real_HT_Tunner
{
    public partial class HT : Form
    {
        Capture zCapture1;
        Capture zCapture2;

        Image<Bgr, byte> zLImg = new Image<Bgr, byte>(640, 480);
        Image<Bgr, byte> zRImg = new Image<Bgr, byte>(640, 480);
        Mat zLGray = new Mat();
        Mat zRGray = new Mat();
        Mat zLGray2 = new Mat();
        Mat zRGray2 = new Mat();
        Mat zLGray3 = new Mat();
        Mat zRGray3 = new Mat();
        Mat zLGrayRec = new Mat();
        Mat zRGrayRec = new Mat();
        Image<Gray, short> zDisparity = new Image<Gray, short>(640, 480);
        Image<Gray, short> zDisparity2 = new Image<Gray, short>(640, 480);

        Int32 minDisparity = 0;
        Int32 numDisparities = 144;
        Int32 SADWindowSize = 5;
        Int32 P1 = 5*5*15;
    }
}

```



```

Int32 P2 = 5*5*5*23;
Int32 disp12MaxDiff = 50;
Int32 preFilterCap = 5;
Int32 UniquenessRatio = 0;
Int32 speckleWindowSize = 1;
Int32 speckleRange = 1;

short[,] zDispData;
Byte[,] zVDispData;
Byte[,] zUDispData;
Image<Gray, Byte> zReCreateV, zReCreateV8, zReCreateU, zReCreateU8;

LineSegment2D[] lines;
LineSegment2D[] lines2;
Image<Gray, Byte> lineImage;
Image<Gray, Byte> lineImage2;

int rhoV = 1;
int thetaV = 1;
int thresV = 1;
int minV = 1;
int maxV = 1;
int rhoU = 1;
int thetaU = 1;
int thresU = 1;
int minU = 1;
int maxU = 1;

public HT()
{
    InitializeComponent();

    try
    {
        zCapture1 = new Capture(2);
        zCapture2 = new Capture(1);
        zCapture1.Start();
        zCapture2.Start();
        Application.Idle += ProcessFrame;
    }
    catch (NullReferenceException excpt)
    {
        MessageBox.Show(excpt.Message);
    }
}

private void ProcessFrame(object sender, EventArgs arg)
{
    Console.WriteLine("1");
    zCapture1.FlipVertical = true;
    zCapture2.FlipVertical = true;
    zCapture1.Retrieve(zLImg);
    zCapture2.Retrieve(zRImg);
    Console.WriteLine("1");
    Compute_Disp();

    zDispData = new short[zDisparity.Height, zDisparity.Width, 1];
    zDispData = zDisparity.Data;
}

```

```

zVDispData = new Byte[zDisparity.Height, numDisparities, 1];
zUDispData = new Byte[numDisparities, zDisparity.Width, 1];
Console.WriteLine("1");
Compute_UV();

zReCreateV = new Image<Gray, Byte>(zVDispData);
zReCreateV8 = new Image<Gray, Byte>(zReCreateV.Size);
zReCreateU = new Image<Gray, Byte>(zUDispData);
zReCreateU8 = new Image<Gray, Byte>(zReCreateU.Size);
Console.WriteLine("1");
Compute_Canny();
Compute_HT_V();
Compute_HT_U();
Compute_ROI();
}

public void Compute_Disp()
{
    CvInvoke.CvtColor(zLImg, zLGray, ColorConversion.Rgb2Gray);
    CvInvoke.CvtColor(zRImg, zRGray, ColorConversion.Rgb2Gray);
    CvInvoke.GaussianBlur(zLGray, zLGray2, new Size(0, 0), 4);
    CvInvoke.GaussianBlur(zRGray, zRGray2, new Size(0, 0), 4);
    CvInvoke.EqualizeHist(zLGray2, zLGray3);
    CvInvoke.EqualizeHist(zRGray2, zRGray3);
    using (StereoSGBM stereoSolver = new StereoSGBM(minDisparity,
numDisparities, SADWindowSize, P1, P2, disp12MaxDiff, preFilterCap,
UniquenessRatio, speckleWindowSize, speckleRange, 0))
    { stereoSolver.Compute(zLGray3, zRGray3, zDisparity); }
    imageBox5.Image = zDisparity;
}

public void Compute_UV()
{
    Parallel.For(0, zDisparity.Height, i =>
    {
        Parallel.For(0, zDisparity.Width, k =>
        {
            int d = Math.Abs(zDispData[i, k, 0] / 16); // If Using
16 Bit Disparity map for real Disparity Value hav to devied by 16...
Disparity Map ที่ใช้เป็น 16 bit ค่า Pixel Value = Disp Val * 16
            zVDispData[i, d, 0]++;
            zUDispData[d, k, 0]++;
        });
    });

    Parallel.For(0, 10, i =>
    {
        Parallel.For(0, zDisparity.Width, k =>
        {
            zUDispData[i, k, 0] = 0;
        });
        Parallel.For(0, zDisparity.Height, k =>
        {
            zVDispData[k, i, 0] = 0;
        });
    });
}

```

```

public void Compute_Canny()
{
    zReCreateV8 = zReCreateV.Canny(75, 150);
    imageBox1.Image = zReCreateV8;
    zReCreateU8 = zReCreateU.Canny(150, 200);
    imageBox3.Image = zReCreateU8;
}

public void Compute_HT_V()
{
    lines = CvInvoke.HoughLinesP(zReCreateV8, rhoV, Math.PI /
thetaV, thresV, minV, maxV);
    lineImage = new Image<Gray, Byte>(zReCreateV8.Size);
    foreach (LineSegment2D line in lines)
    {
        lineImage.Draw(line, new Gray(255), 2);
    }
    imageBox2.Image = lineImage;
    label21.Text = lines.Length.ToString();
}

public void Compute_HT_U()
{
    lines2 = CvInvoke.HoughLinesP(zReCreateU8, rhoU, Math.PI /
thetaU, thresU, minU, maxU);
    lineImage2 = new Image<Gray, Byte>(zReCreateU8.Size);
    foreach (LineSegment2D line in lines2)
    {
        lineImage2.Draw(line, new Gray(255), 2);
    }
    imageBox4.Image = lineImage2;
    label22.Text = lines2.Length.ToString();
}

public void Compute_ROI()
{
    bool chk1 = false;
    bool chk2 = false;
    //----P1----//
    int zLineCnt1 = 0;
    foreach (LineSegment2D line in lines)
    {
        foreach (LineSegment2D line2 in lines2)
        {
            int zDispValV = line.P1.X;
            int zDispValU = line2.P1.Y;
            if (zDispValV == zDispValU)
            {
                zLineCnt1++;
            }
        }
    }
    int[] P1X = new int[zLineCnt1];
    int[] P1Y = new int[zLineCnt1];

    if (zLineCnt1 != 0)
    {
        chk1 = true;
    }
}

```

```

zLineCnt1 = 0;
foreach (LineSegment2D line in lines)
{
    foreach (LineSegment2D line2 in lines2)
    {
        int zDispValV = line.P1.X;
        int zDispValU = line2.P1.Y;
        if (zDispValV == zDispValU)
        {
            P1Y.SetValue(line.P1.Y, zLineCnt1);
            P1X.SetValue(line2.P1.X, zLineCnt1);
            zLineCnt1++;
        }
    }
}
else
{
    chk1 = false;
}

Console.WriteLine(zLineCnt1);
//----P2----//
int zLineCnt2 = 0;
for (int i = 0; i < lines.Length; i++)
{
    for (int j = 0; j < lines2.Length; j++)
    {
        int zDispValV = lines[i].P2.X;
        int zDispValU = lines2[j].P2.Y;
        if (zDispValV == zDispValU)
        {
            zLineCnt2++;
        }
    }
}

int[] P2X = new int[zLineCnt2];
int[] P2Y = new int[zLineCnt2];

if (zLineCnt2 != 0)
{
    chk2 = true;
    zLineCnt2 = 0;
    for (int i = 0; i < lines.Length; i++)
    {
        for (int j = 0; j < lines2.Length; j++)
        {
            int zDispValV = lines[i].P2.X;
            int zDispValU = lines2[j].P2.Y;
            if (zDispValV == zDispValU)
            {
                P2Y.SetValue(lines[i].P2.Y, zLineCnt2);
                P2X.SetValue(lines2[j].P2.X, zLineCnt2);
                zLineCnt2++;
            }
        }
    }
}
}

```

```

    }
    else
    {
        chk2 = false;
    }

    if (chk1 == true && chk2 == true)
    {
        int[] Xmax = new int[2] { P1X.Max(), P2X.Max() };
        int[] Xmin = new int[2] { P1X.Min(), P2X.Min() };
        int[] Ymax = new int[2] { P1Y.Max(), P2Y.Max() };
        int[] Ymin = new int[2] { P1Y.Min(), P2Y.Min() };

        zDisparity2 = zDisparity.Clone();
        Rectangle zROI = new Rectangle(Xmin.Min(), Ymin.Min(),
Xmax.Max() - Xmin.Min(), Ymax.Max() - Ymin.Min());
        zDisparity2.Draw(zROI, new Gray(255), 2);
        imageBox6.Image = zDisparity2;
    }
    else
    {
        imageBox6.Image = zDisparity;
    }
}

private void trackBar1_Scroll(object sender, EventArgs e)
{
    rhoV = trackBar1.Value;
    label16.Text = rhoV.ToString();
    Compute_HT_V();
    Compute_ROI();
}

private void trackBar2_Scroll(object sender, EventArgs e)
{
    thetaV = trackBar2.Value;
    label17.Text = thetaV.ToString();
    Compute_HT_V();
    Compute_ROI();
}

private void trackBar3_Scroll(object sender, EventArgs e)
{
    thresV = trackBar3.Value;
    label18.Text = thresV.ToString();
    Compute_HT_V();
    Compute_ROI();
}

private void trackBar4_Scroll(object sender, EventArgs e)
{
    minV = trackBar4.Value;
    label19.Text = minV.ToString();
    Compute_HT_V();
    Compute_ROI();
}

```

```

private void trackBar5_Scroll(object sender, EventArgs e)
{
    maxV = trackBar5.Value;
    label120.Text = maxV.ToString();
    Compute_HT_V();
    Compute_ROI();
}

private void trackBar6_Scroll(object sender, EventArgs e)
{
    rhoU = trackBar6.Value;
    label111.Text = rhoU.ToString();
    Compute_HT_U();
    Compute_ROI();
}

private void trackBar7_Scroll(object sender, EventArgs e)
{
    thetaU = trackBar7.Value;
    label112.Text = thetaU.ToString();
    Compute_HT_U();
    Compute_ROI();
}

private void trackBar8_Scroll(object sender, EventArgs e)
{
    thresU = trackBar8.Value;
    label113.Text = thresU.ToString();
    Compute_HT_U();
    Compute_ROI();
}

private void trackBar9_Scroll(object sender, EventArgs e)
{
    minU = trackBar9.Value;
    label114.Text = minU.ToString();
    Compute_HT_U();
    Compute_ROI();
}

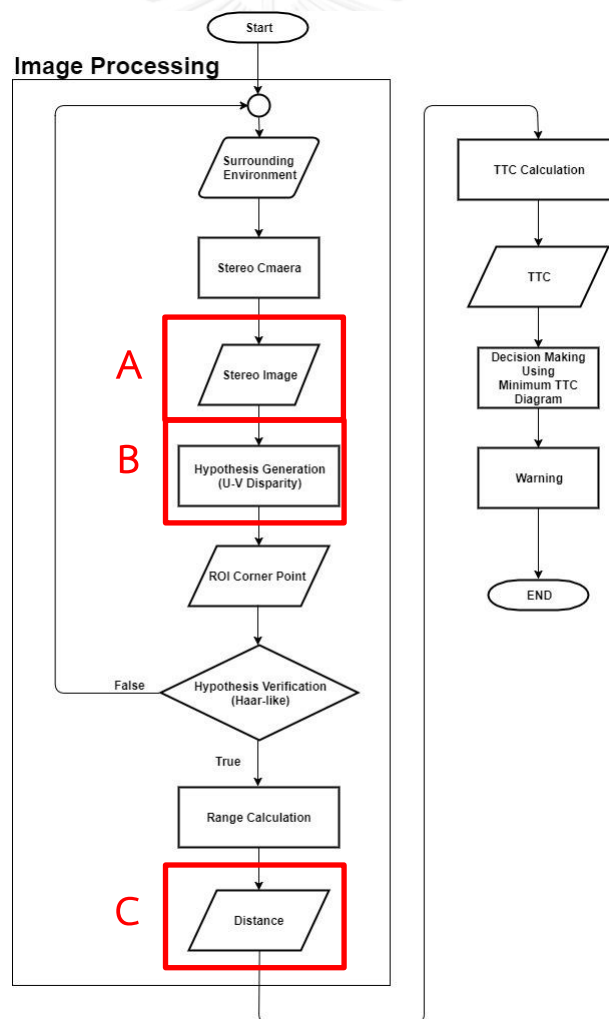
private void trackBar10_Scroll(object sender, EventArgs e)
{
    maxU = trackBar10.Value;
    label115.Text = maxU.ToString();
    Compute_HT_U();
    Compute_ROI();
}
}
}
}

```

รูปที่ 83 ภาพแสดง Code โปรแกรมที่ใช้ในการปรับแต่งค่าในการทำ Hough Transformation

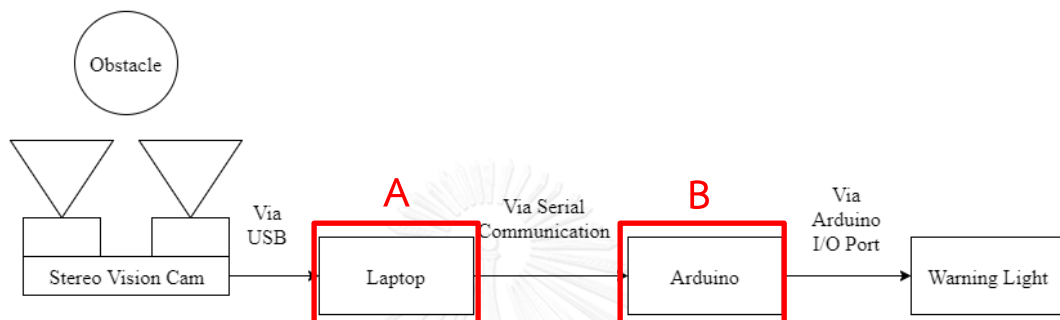
การนำระบบเตือนการชนด้านหน้าไปใช้งาน ตัวแปรที่ใช้ในกระบวนการ Image Processing ต้องถูกปรับแต่งให้เหมาะสมกับสภาพแวดล้อมที่จะถูกนำไปใช้งาน

ขั้นแรกทำการปรับแต่งค่าตัวแปรของการทำแผนภาพ Disparity (ส่วน A ในรูปที่ 84) ดังแสดงในรูปที่ 78 โดยเมื่อปรับแต่งเสร็จแล้ว ตัวแปรต่าง ๆ ที่ได้จะถูกนำไปใช้ในคำสั่ง การสร้างแผนภาพ Disparity จาก Library EmguCV หลังจากนั้นนำทำการปรับค่าตัวแปรในกระบวนการทำ Canny Edge Detection (ส่วน B ในรูปที่ 84) แสดงในรูป 80 หลังจากนั้นทำการปรับแต่งตัวแปรเพื่อการตรวจจับเส้นตรงของแผนภาพ U-V Disparity (ส่วน B ในรูปที่ 84 เช่นกัน) ดังแสดงในรูปที่ 82 ขั้นตอนสุดท้ายก่อนการนำไปใช้งานได้จริงต้องมีการ Calibration ระยะทางที่ได้จากแผนภาพ Disparity โดยค่าคู่ค่า Disparity ที่ได้จาก บรรทัดสุดท้ายในหน้าต่าง Console ที่แสดงในรูปที่ 76 เพื่อนำไปหาความสัมพันธ์ระหว่างค่า Disparity กับระยะทางเพื่อนำมาคำนวณระยะของวัตถุ (ส่วน C ในรูปที่ 84)



รูปที่ 84 แผนผังของระบบเตือนการชนท้ายด้วยการแนะนำการควบคุมเบรกร่วมกับการเลี้ยวหลบ

ระบบ และโปรแกรมที่แสดงในส่วนก่อนหน้า นี้ ของภาคผนวก ข เป็นโปรแกรมที่ใช้ในการคำนวณค่า TTC และเตือนผู้ขับขี่ตามแผนภาพ TTC ต่ำสุด (ส่วน A ในรูปที่ 85) ซึ่ง Out Put ของระบบคือ Warning State โดยจะมีอยู่ทั้งหมด 3 ค่า เป็นตัวเลข 0 คือ ไม่เตือน 1 คือ เตือนให้ผู้ขับขี่เบรก และ 2 เตือนให้ผู้ขับขี่เลี้ยวหลบ โดยค่า Warning State นี้จะถูกนำไปใช้เป็น Input ให้กับ Microcontroller Arduino Uno (ส่วน B ในรูปที่ 85) เพื่อทำการเตือนผู้ขับขี่ต่อไป โดยมี Code แสดงในรูปที่ 36



รูปที่ 85 แผนภาพจำลองระบบเตือนการชนด้านหน้า

```

#include <Adafruit_NeoPixel.h>

String zState;

Adafruit_NeoPixel zWarnLight = Adafruit_NeoPixel(8, 9);

void setup() {
  // put your setup code here, to run once:

  Serial.begin(9600);

  Serial.setTimeout(0);

  zWarnLight.begin();

  zWarnLight.setPixelColor(0, zWarnLight.Color(0,0,0));

  zWarnLight.show();

  delay(100);

  zWarnLight.setPixelColor(0, zWarnLight.Color(255,255,255));
  
```



```
zWarnLight.show();

delay(100);

zWarnLight.setPixelColor(0, zWarnLight.Color(0,0,0));

zWarnLight.show();

delay(100);

zWarnLight.setPixelColor(0, zWarnLight.Color(255,255,255));

zWarnLight.show();

delay(100);

zWarnLight.setPixelColor(0, zWarnLight.Color(0,0,0));

zWarnLight.show();

delay(100);

zWarnLight.setPixelColor(0, zWarnLight.Color(255,255,255));

zWarnLight.show();

delay(100);

zWarnLight.setPixelColor(0, zWarnLight.Color(0,0,0));

zWarnLight.show();

}

void loop() {

  // put your main code here, to run repeatedly:

  zState = Serial.readString();

  if (zState == "0"){

    // Stop the LED

    for(int i=0;i<8;i++){

      zWarnLight.setPixelColor(i, zWarnLight.Color(0,0,0));
```

```
    }  
  }  
  else if (zState == "1"){  
    // LED Red  
    for(int i=0;i<8;i++){  
      zWarnLight.setPixelColor(i, zWarnLight.Color(150,0,0));  
    }  
  }  
  else if (zState == "2"){  
    // LED Yellow  
    for(int i=0;i<8;i++){  
      zWarnLight.setPixelColor(i, zWarnLight.Color(90,60,0)); // Moderately bright  
      green color.  
    }  
  }  
  zWarnLight.show();  
}
```

รูปที่ 86 ภาพแสดง Code โปรแกรมที่ใช้ในการเตือนผู้ขับขี่ด้วย Arduino Uno

ประวัติผู้เขียนวิทยานิพนธ์

นายเสกฐวุฒิ ลาภวิสุทธิสารโรรจน์ เกิดเมื่อวันที่ 11 ตุลาคม พุทธศักราช 2533 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตรบัณฑิต เกียรตินิยมอันดับ 1 สาขา เทคโนโลยีวิศวกรรมยานยนต์ ภาควิชาเทคโนโลยีวิศวกรรมเครื่องต้นกำลัง วิทยาลัยเทคโนโลยี อุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ และเข้าศึกษาต่อในหลักสูตร วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมเครื่องกล ภาควิชาวิศวกรรมเครื่องกล คณะ วิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2557 และมีงานวิจัยที่ได้รับการเผยแพร่ ดังต่อไปนี้

1. "The Vehicle Steering and Braking Guidance using Vehicle Dynamics Approach for Forward Collision Warning System." in 9th Atrans Symposium on Transportation for a Better Life: Safe and Smart City,. 2016. Bangkok, Thailand.

2. "Study on The Effect of Driver Reaction Times to the Collision Possibility and Evasive Ability due to The Forward Collision Warning System using Driving Simulator." in The 7th TSME International Conference on Mechanical Engineering,. 2016. Chaing Mai, Thailand.