การเรียนรู้ของเครื่องแบบเสริมกำลังเพื่อควบคุมสัญญาณไฟจราจร

REINFORCEMENT LEARNING FOR CONTROLLING TRAFFIC SIGNALS

ชนายุทธ แจ่มขาว

ณัฐภัทร พวงพิพัฒน์

ทศ วรศรีวิศาล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2563

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use. Forbidden to modify the content, and cite the document when use.

ปริญญานิพนธ์ปีการศึกษา 2563

ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเรียนรู้ของเครื่องแบบเสริมกำลังเพื่อควบคุมสัญญาณไฟจราจร

REINFORCEMENT LEARNING FOR CONTROLLING TRAFFIC SIGNALS

ผู้จัดทำ

1. นายชนายุทธ แจ่มขาว รหัสนักศึกษา 60010247
2. นายณัฐภัทร พวงพิพัฒน์ รหัสนักศึกษา 60010325
3. นายทศ วรศรีวิศาล รหัสนักศึกษา 60010368

_____อาจารย์ที่ปรึกษา

(รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

การเรียนรู้ของเครื่องแบบเสริมกำลังเพื่อควบคุมสัญญาณไฟจราจร

นายชนายุทธ	แจ่มขาว	60010247
นายณัฐภัทร	พวงพิพัฒน์	60010325
นายทศ	วรศรีวิศาล	60010368
รศ.ดร.เกียรติกูล	เจียรนัยธนะกิจ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2563

บทคัดย่อ

โครงงานนี้เป็นการสร้างแบบจำลองการควบคุมไฟสัญญาณจราจรโดยใช้การเรียนรู้ของเครื่องแบบเสริมกำลัง โดยประกอบไปด้วยการศึกษากระบวนการการทำงาน ข้อดีและข้อเสีย แนวคิด เพื่อรวบรวมองค์ความรู้ต่าง ๆที่จำเป็นต่อ การควบคุมสัญญาณไฟจราจรด้วย Reinforcement Learning เพื่อปรับปรุง พัฒนา และ เพิ่มประสิทธิภาพให้กับระบบการจัดการคมนาคม ทั้งนี้ยังมีการนำเทคนิคอื่น ๆ มาประยุกต์เข้ากับ Reinforcement Learning เพื่อเพิ่มประสิทธิภาพในการทำงานให้มากยิ่งขึ้น โดยการนำเสนอแบบจำลองของ Model และสภาพแวดล้อมการจราจรผ่านทางโปรแกรม Simulation of Urban Mobility (SUMO)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

Reinforcement Learning for Controlling Traffic Signals

Mr.Chanayut	Jamkhaw	60010247
Mr.Nattapat	Puangpipat	60010325
Mr.Tot	Worasrivisal	60010368
Assoc.Prof.Dr.Kietikul	Jearanaitanakij	Advisor

Academic Year 2020

ABSTRACT

In this project will show about traffic signal control model by using reinforcement learning including working and learning progress, pros and cons, and concept which is necessary for traffic signal control using reinforcement learning to improve traffic system. In addition, other technology also have brought to apply with reinforcement learning to improve the performance of model. Simulation of Urban Mobility (SUMO) has been used as the traffic simulation environment.

King Mongkut's Institute of Technology Ladkrabang

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use. Forbidden to modify the content, and cite the document when use.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถประสบความสำเร็จลุล่วงได้ตามเป้าหมายด้วยความช่วยเหลือจาก รศ.ดร.เกียรติกูล เจียรนัยธนะกิจ อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำปรึกษา แนวคิด มุมมอง ตลอดจนแก้ไขข้อบกพร่องต่างๆ มาตลอดจนปริญญานิพนธ์ฉบับนี้เสร็จสมบูรณ์

นอกจากนี้ยังได้รับความช่วยเหลือจากคณาจารย์ รวมไปถึงนักศึกษาภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ให้คำปรึกษาต่างๆเพิ่มเติม และร่วมแบ่งปันความรู้ ประสบการณ์ รวมถึงแนวทางการแก้ปัญหาต่างๆเพื่อนำมาประกอบการทำปริญญานิพนธ์ฉบับนี้

ผู้จัดทำจึงขอกราบขอบพระคุณทุกๆท่านเป็นอย่างสูง มา ณ โอกาสนี้

ชนายุทธ แจ่มขาว

ณัฐภัทร พวงพิพัฒน์

ทศ วรศรีวิศาล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VI

สารบัญรูป VII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาของปัญหา 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ประโยชน์ที่คาดว่าจะได้รับ 2

1.4 ขอบเขตของโครงงาน 2

1.5 ข้อจำกัดของโครงงาน 3

บทที่ 2 ทฤษฎีและงานวิจัยที่เกี่ยวข้อง 4

2.1 Reinforcement Learning 4

2.2 Simulation of Urban MObility 7

2.3 Python 10

2.4 Traffic Engineering 12

2.6 เอกสารงานวิจัยที่เกี่ยวข้อง 17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

บทที่ 3 การออกแบบและการพัฒนา 19
3.1 ภาพรวมระบบ 19
3.2 Reinforcement Learning 20
3.3 SUMO (Simulation of Urban MObility) 27
3.4 Simulation API 35
3.5 CSV And Plotter 39

บทที่ 4 การออกแบบและการพัฒนา 44
4.1 การทดลองระบบไฟจราจรแบบ Fix เวลา 44
4.2 การทดลองระบบไฟจราจรด้วย Reinforcement Learning อ้างอิงงานวิจัย TSCRL 45
4.3 การทดลองระบบไฟจราจรด้วย Reinforcement Learning ของปริญญานิพนธ์ 46
4.4 กราฟสรุปผลการทดลอง 48

บทที่ 5 บทสรุปและข้อเสนอแนะ 57
5.1 บทสรุป 57
5.2 ปัญหาและอุปสรรค 58
5.3 แนวทางการพัฒนาต่อ 58

บรรณานุกรม 59
ภาคผนวก ก 60
ภาคผนวก ข 64
ภาคผนวก ค 67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญตาราง

ตาราง **หน้า**

3.1 Specification ของอุปกรณ์ที่ใช้ทดสอบ 20

3.2 State transition 22

3.3 การเก็บค่าภายใน State Space 26

3.4 ความหนาแน่นของยานพาหนะในแต่ละแผนที่ 29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

สารบัญรูป

รูป **หน้า**

2.1 Markov Decision Process Model 4

2.2 Q-Learning Algorithm 6

2.3 Epsilon-greedy Algorithm 6

2.4 Bellman Equation 7

2.5 Simulation of Urban Mobility Logo 7

2.6 โครงสร้างของภาษา XML 9

2.7 Python Logo 10

2.8 Matplotlib Logo 11

2.9 Greenshields Linear model 13

2.10 กราฟพาราโบลาแสดงความสัมพันธ์ของสมการ 3.4 และ 3.5 15

2.11 แสดงกราฟที่ได้จากสมการ 3.7 16

2.12 State and Environment 17

2.13 Action set 18

3.1 Diagram การทำงานของระบบ 19

3.2 สถานะ State ของระบบ 21

3.3 Action ในระบบ 22

3.4 การใช้โปรแกรม Netedit ในการออกแบบแผนที่การจราจร 27

3.5 การใช้โปรแกรม Netedit ในการออกแบบบริเวณ 4 แยกไฟจราจร 28

3.6 การออกแบบเส้นทางยานพาหนะบริเวณ 4 แยก 28

3.7 กำหนดสถานะไฟจราจร Character 1-3 โดยโปรแกรม Netedit 31

3.8 กำหนดสถานะไฟจราจร Character 4-6 โดยโปรแกรม Netedit 32

3.9 กำหนดสถานะไฟจราจร Character 7-9 โดยโปรแกรม Netedit 32

3.10 กำหนดสถานะไฟจราจร Character 10-12 โดยโปรแกรม Netedit 32

3.11 เส้นทางในการคานวณ Green Time 34

3.12 บริเวณที่วาง Detector เพื่อวัดค่า Flow Rate 37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

3.13 บริเวณที่วาง Detector เพื่อวัดค่า Arrival rate 38
3.14 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Speed และ Density..... 41
3.15 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Density 42
3.16 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Speed..... 42
4.1 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น มาก..... 48
4.2 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น มาก..... 48
4.3 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น กลาง 49
4.4 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น กลาง 49
4.5 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น น้อย 50
4.6 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น น้อย 50
4.7 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น มาก..... 51
4.8 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น มาก 51
4.9 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น กลาง 52
4.10 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น กลาง 52
4.11 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น น้อย..... 53
4.12 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น น้อย..... 53
4.13 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น มาก..... 54
4.14 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น มาก 54
4.15 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น กลาง 55
4.16 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น กลาง 55
4.17 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น น้อย..... 56
4.18 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น น้อย..... 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 1

บทนำ

1.1 ความเป็นมาของปัญหา

เนื่องจากในปัจจุบันการใช้ยานพาหนะบนท้องถนนมีแนวโน้มของความต้องการเพิ่มมากขึ้นอย่างมหาศาล ทั้งยานพาหนะส่วนตัว ยานพาหนะสำหรับขนส่งสาธารณะ แต่ในขณะเดียวกันระบบจัดการด้านการคมนาคม การขนส่ง การเดินทาง ไม่ได้มีประสิทธิภาพมากขึ้นตามความต้องการที่มหาศาลนั้น โดยเฉพาะอย่างยิ่งระบบการจัดการภายในเมือง ที่มีการสัญจรด้วยยานพาหนะอย่างคับคั่ง ยกตัวอย่างเช่น กรุงเทพมหานคร ในความเป็นจริงที่ว่าด้วยความต้องการในการใช้ยานพาหนะที่เพิ่มมากขึ้น แต่ในทางกลับกัน ระบบการจัดการเพื่อรองรับต่อความต้องการนั้น ๆ กลับมีข้อจำกัดมากมาย โดยสรุปปัจจัยที่ส่งผลให้การจราจรติดขัด เป็นหัวข้อคร่าวๆดังนี้ ทั้งในทางด้านของการพัฒนาและขยายพื้นที่ของพื้นถนนที่มีอยู่อย่างจำกัด ด้านการปรับปรุงซ่อมแซมถนนที่มีอยู่ตลอดเป็นระยะ รวมถึงพฤติกรรมการขับขี่ของผู้ขับขี่ที่ทำให้เกิดอุบัติเหตุเป็นผลให้การจราจรติดขัดเป็นเวลา

1.1.1 การพัฒนาหรือปรับปรุงท้องถนน

การพัฒนาปรับปรุงสภาพนั้นถือเป็นเรื่องที่ดี เพราะเมื่อทำเสร็จก็จะส่งผลดีแต่สภาพการจราจรโดยรอบ แต่ในระหว่างที่ทำนั้น บางครั้งไม่ได้วางแผนการจัดการที่ดีพอหรือด้วยปัจจัยอื่น ๆ ทำให้สภาพการจราจรโดยรอบติดขัดหนักมาก

1.1.2 พฤติกรรมผู้ขับขี่

หนึ่งในสาเหตุของการจราจรที่ติดขัดคืออุบัติเหตุ และหนึ่งในสาเหตุของอุบัติเหตุที่พบได้บ่อยที่สุดก็มาจากพฤติกรรมผู้ขับขี่ เช่นเกิดความประมาท หรือการตัดสินใจที่ผิดพลาด ส่งผลให้เกิดอุบัติเหตุตามมาในที่สุด

1.1.3 สภาพอากาศ

เราสามารถพบสภาพการจราจรที่ติดขัดได้ในวันที่ฝนตกหนัก ๆ ซึ่งเป็นปัจจัยที่เราสามารถหลีกเลี่ยงได้ยากที่สุด เนื่องจากเป็นปัจจัยที่เกิดตามธรรมชาติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.2 วัตถุประสงค์ของโครงงาน

1) จากปัญหาที่กล่าวมาในหัวข้อ 1.1 นั้น การเพิ่มประสิทธิภาพของระบบการจัดการคมนาคมในรูปแบบไฟจราจรมีความเป็นไปได้มากที่สุด ดังนั้นจุดประสงค์หลักของโครงงานนี้คือการควบคุมไฟจราจรที่มีประสิทธิภาพ ส่งผลให้การจราจรติดขัดน้อยลง
2) ไฟจราจรสามารถปรับเปลี่ยนช่วงเวลาการทำงานของตัวเองให้เข้ากับสถานการณ์การจราจรที่เปลี่ยนจากปัจจัยต่าง ๆ ไม่ว่าจะเป็นช่วงเวลาที่เร่งด่วนมีผู้ใช้ยานพาหนะบนท้องถนนเยอะ หรือช่วงเวลาที่เกิดอุบัติเหตุทำให้การจราจรติดขัดเป็นต้น
3) ศึกษาทฤษฎี Reinforcement Learning จนเกิดความเข้าใจอย่างลึกซึ้ง เพื่อนำทฤษฎีมาปรับใช้ในการควบคุมสัญญาณไฟจราจรให้มีประสิทธิภาพมากยิ่งขึ้น นอกจากนี้ยังอาจนำทฤษฎีที่ได้ ไปปรับใช้กับปัญหาอื่น ๆในภายภาคหน้าอีกด้วย
4) ศึกษาทฤษฎีต่าง ๆ ที่เกี่ยวข้อง เช่น Particle Swarm Optimization , Neural Network และอื่น ๆ เพื่อนำมาปรับปรุงโมเดลที่ถูกพัฒนาด้วย Reinforcement Learning

1.3 ประโยชน์ที่คาดว่าจะได้รับ

สามารถแก้ไขปัญหาการจราจรที่ติดขัดอันเนื่องมาจากสาเหตุในข้อ 1.1 เนื่องจากโมเดลถูกพัฒนาขึ้นมาเพื่อแก้ไขปัญหาเหล่านั้นโดยเฉพาะ ทำให้ผู้ใช้งานยานพาหนะบนถนนสามารถใช้ท้องถนนได้อย่างมีประสิทธิภาพมากกว่าการที่จอดรอรถติดนาน ๆ

นอกจากจะได้ประโยชน์ในทางตรงแล้วยังมีประโยชน์ในทางอ้อมอีกด้าน ก็คือรถเวลาการใช้ยานพาหนะบนท้องถนน ซึ่งจะส่งผลทำให้เกิดการปล่อยก๊าซคาร์บอนไดออกไซด์น้อยลง เพราะในช่วงการจราจรติดขัดนั้น ยานพาหนะต่าง ๆ ที่จอดติดอยู่เกิดการปล่อยก๊าซคาร์บอนไดออกไซด์ไปแบบเปล่าประโยชน์ ซึ่งถ้าเราสามารถลดปัญหาตรงนี้ได้ก็จะช่วยส่งผลดีต่อสิ่งแวดล้อม

1.4 ขอบเขตของโครงงาน

1) โมเดลที่พัฒนานั้นสามารถแก้ไขปัญหาการจราจร ที่มีปัจจัยมาจากยานพาหนะจำนวนมากในหลายๆระดับได้
2) โมเดลที่พัฒนาสามารถนำไปใช้กับแยกการจราจรจำนวน 1แยก 4แยก 9แยก และ 16แยกได้
3) โมเดลที่พัฒนานั้นจะถูกพัฒนาในโปรแกรมจำลองการจราจร SUMO
4) ส่วนเชื่อมต่อระหว่างโปรแกรมจำลองการจราจร SUMO และ Model Reinforcement พัฒนาด้วย python

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

1.5 ข้อจำกัดของโครงงาน

เนื่องจากโมเดลนั้นถูกพัฒนาด้วยโปรแกรมจำลองทำให้ผลที่วัดออกมาก็จะมาจากโปรแกรมจำลองเหมือนกัน ดังนั้นการนำไปใช้งานจริงอาจมีความคลาดเคลื่อนได้ในช่วงแรก แต่เมื่อใช้เวลาเรียนรู้ซักระยะหนึ่ง ตัวโมเดลจะสามารถทำงานได้อย่างมีประสิทธิภาพมากขึ้นเอง

ข้อมูลการจราจรนั้นเป็นข้อมูลที่ได้จำลองขึ้นมาโดยการสุ่มเกิดรถยนต์เข้าไปในแยกการจราจร ทำให้อาจคลาดเคลื่อนจากข้อมูลจริง ๆ แต่พยายามทำให้ใกล้เคียงกับข้อมูลจริงมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

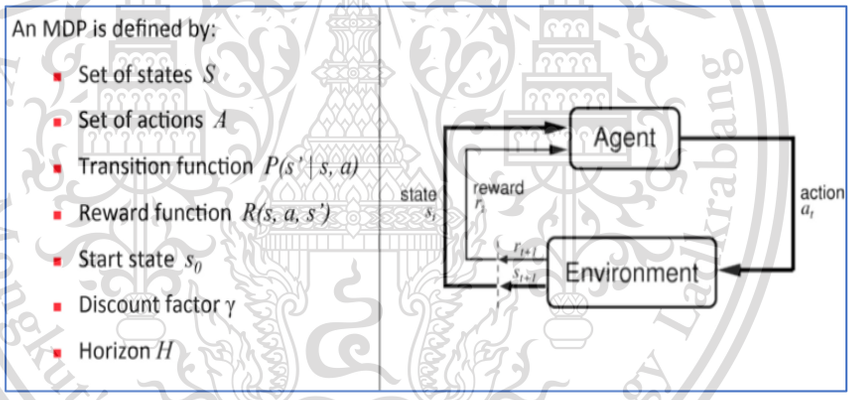
บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 Reinforcement Learning

Reinforcement Learning หรือที่เรียกว่า การเรียนรู้แบบเสริมกำลัง ซึ่งเป็นหนึ่งในวิธีในการทำ Machine Learning นอกเหนือจากวิธี Supervised Learning และ Unsupervised Learning โดย Reinforcement Learning นั้นมีรูปแบบในการเรียนรู้ด้วยการลองผิดลองถูกเพื่อให้โปรแกรมพัฒนาระบบที่ใช้ในการตัดสินใจได้ดียิ่งขึ้น ซึ่งจะเกิดจากการมีปฏิสัมพันธ์ระหว่าง ผู้เรียนรู้ (Agent) และสิ่งแวดล้อม (Environment) โดยกระบวนการทำงานของ Reinforcement Learning จะมี Model ทางคณิตศาสตร์ที่ชื่อว่า Markov Decision Process (MDP) เป็น Model อ้างอิง

2.1.1 Markov Decision Process



รูป 2.1 Markov Decision Process Model

จากรูปที่ 2.1 Markov Decision Process จะประกอบไปด้วย 4 ส่วนหลักๆได้แก่

2.1.1.1 Agent

ผู้เรียนรู้ที่ถูกสร้างขึ้นมาเพื่อใช้ในการตัดสินใจภายใต้สถานการณ์ (State) ที่แตกต่างกันออกไปโดยจะทำการเลือกการกระทำบางอย่าง (Action) เพื่อให้นำมาซึ่งรางวัล (Reward) ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.1.1.2 Action

การกระทำที่ Agent สามารถกระทำในรูปแบบสุ่ม หรือจะอ้างอิงจากสถานการณ์ (State) ต่าง ๆโดยสามารถส่งผลให้เกิดการเปลี่ยนแปลงของStateได้อาจจะเป็นในรูปแบบของการเคลื่อนย้ายวัตถุสิ่งของ เป็นต้น ซึ่งในแต่ละปัญหาก็จะมีชุดของ Action ที่แตกต่างกันออกไป

2.1.1.3 State

เป็นสถานการณ์ หรือ สถานะ ที่ทำปฏิสัมพันธ์กับ Agent โดยตรงเพื่อใช้พิจารณาในการเลือกการกระทำ Action ต่าง ๆโดยแต่ละ State จะสัมพันธ์กันในลักษณะของความน่าจะเป็น เช่น ณ State หนึ่งเลือกกระทำ Action ใด ๆ จะมีความน่าจะเป็นที่จะคงอยู่ State เดิม หรือเปลี่ยนเป็น State ที่แตกต่างกันออกไป

2.1.1.4 Reward

สิ่งที่เป็นเป้าหมายของ Agent โดย Agent ต้องการให้ค่า Reward มีค่าให้ได้มากที่สุด เปรียบเสมือนรางวัลที่ได้จากการทำ Action ใด ๆ Reward นั้นอาจจะเป็นได้หลายรูปแบบจะไม่มีข้อกำหนดตายตัว โดยอาจจะใช้สมการทางคณิตศาสตร์เข้ามาช่วยคำนวณ Reward ก็ได้ขึ้นอยู่กับโจทย์ และ ความเหมาะสม

จากความหมายที่กล่าวมาข้างต้นจะสรุปการทำงานได้ดังนี้ Reinforcement Learning นั้นอาศัยการทดลองกระทำ (Action) หนึ่งจากในสถานการณ์หรือ (state) หนึ่งที่มีสภาพแวดล้อมหนึ่ง (environment) เพื่อให้ได้มาซึ่งรางวัล (reward) และเปลี่ยนไปอยู่อีกสภาวะหนึ่ง (state) โดยมีจุดมุ่งหมายว่าถ้ากระทำจนสิ้นสุดแล้วต้องมีผลรวมของรางวัลได้มากที่สุด สิ่งที่จะทำให้ได้ผลรวมของรางวัล มากที่สุดคือการหาชุดคำสั่ง (policy) ที่ดีที่สุดสำหรับแต่ละสภาวะที่เราอยู่ ซึ่งวิธีการที่จะได้รางวัลรวมมากที่สุดคือการทดลองและประมาณค่าของรางวัลที่เราจะได้เมื่อกระทำในสภาวะใด ๆ โดยมีตารางที่ชื่อว่า Q-table คอยเก็บค่าประมาณที่ได้มา และมีเทคนิคในการหาค่าประมาณคือ Q-learning

2.1.2 Q-Learning

การประมาณค่าของรางวัลที่จะได้ด้วยการหา Reward สูงสุดที่สามารถรับได้จาก การกระทำ Action ใด ๆ ของ State ถัดไป โดย เทคนิค Q-Learning จำเป็นต้องอาศัยการทดลองซ้ำไปซ้ำมาหลาย ๆรอบเพื่อจะทำให้ค่าที่ประมาณได้นั้นมีค่าเข้าใกล้ค่าที่ดีที่สุดนั้นเอง โดย Q-Learning จะมีสิ่งที่เรียกว่า Q-Table ซึ่งปกติแล้วจะอยู่ในรูปแบบของตารางแสดงความสัมพันธ์ระหว่าง State Action และ ค่าของ Q-Value ที่ได้จากสมการ Bellman Equation ซึ่งจะกล่าวในหัวข้อถัด ๆ ไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

Q-learning: An off-policy TD control algorithm

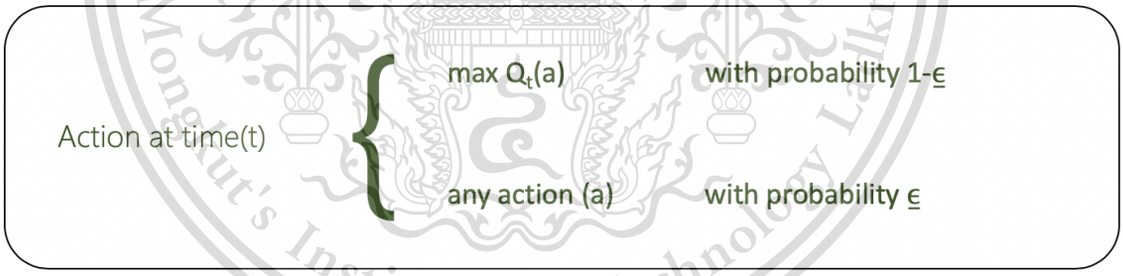
Initialize $Q(s,a), \forall s \in \mathcal{S}, a \in \mathcal{A}(s)$, arbitrarily, and $Q(\textit{terminal-state}, \cdot) = 0$
Repeat (for each episode):
  Initialize S
  Repeat (for each step of episode):
    Choose A from S using policy derived from Q (e.g., ϵ-greedy)
    Take action A, observe R, S'
    $Q(S,A) \leftarrow Q(S,A) + \alpha\big[R + \gamma \max_a Q(S',a) - Q(S,A)\big]$
    $S \leftarrow S'$
  until S is terminal

รูปที่ 2.2 Q-Learning Algorithm

จากรูปที่ 2.2 Algorithm ของ Q-Learning นั้นมีพื้นฐานมาจาก Markov Decision Process Model โดย Agent จะทำ Action กับ State เพื่อสังเกตค่าของ Reward และ Next State เพื่อใช้ในการตัดสินใจเลือก Action ถัด ๆไป ซึ่งจะมี ทฤษฎีที่เกี่ยวข้องดังนี้

2.1.2.1 Epsilon-greedy

เป็น Algorithm ที่ใช้ในการเลือก Action ของ Agent โดยจะมีการกำหนดค่า Epsilon ขึ้นมาเพื่อสร้างความน่าจะเป็นในการสุ่ม Action หรือที่เรียกอีกอย่างว่า (exploration) แทนการเลือก Action ที่ดีที่สุดตามค่าของ Q-value หรือที่เรียกว่า (exploitation)



รูปที่ 2.3 Epsilon-greedy Algorithm

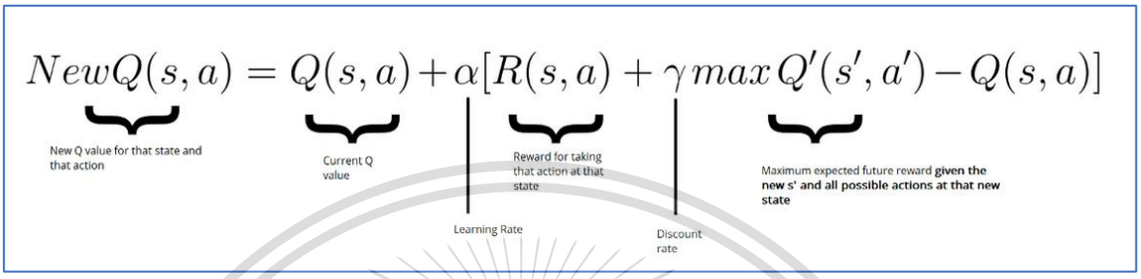
2.1.2.2 Probabilistic-greedy

เป็น Algorithm ที่มีลักษณะคล้ายกับ Epsilon-greedy โดยจะมีการกำหนดค่าค่าหนึ่งที่ใช้ในการสร้างความน่าจะเป็นระหว่างการเลือก Action แบบ Exploration และ Exploitation ซึ่งความแตกต่างของ Probabilistic-greedy และ Epsilon-greedy จะอยู่ในส่วนของการ Exploitation ซึ่งจะมีการถ่วงน้ำหนักของค่า Action ต่าง ๆแปรผันตรงกับค่าของ Q-value ที่เก็บไว้ยิ่ง Action ใด ๆ มีค่าของ Q-Value สูงก็จะมีความน่าจะเป็นที่ Agent จะเลือกกระทำ Action นั้น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.1.2.3 Bellman Equation

หรือที่เรียกอีกอย่างหนึ่งว่า Dynamic Programming Equation ซึ่งเป็นสมการที่คิดค้นโดยนักคณิตศาสตร์ประยุกต์ ชาวอเมริกาที่ชื่อว่า Richard E. Bellman โดยสมการ Bellman equation นั้นเป็นหัวใจหลักของ Q-Learning Algorithm เลยก็ว่าได้ โดยจะใช้เพื่อทำการ Update ค่าของ Q-value ใหม่ลงใน Q-Table



รูปที่ 2.4 Bellman Equation

จากรูปที่ 2.4 จะอธิบายได้ดังนี้ ทางด้านของฝั่งซ้ายจะเป็นค่าของ Q-value ใหม่ที่จะทำการ Update ไปที่ตาราง Q-Table ส่วนทางด้านของสมการฝั่งขวานั้นเริ่มจากเทอมแรกจะเป็นค่าของ Q-value เดิม เทอมถัดมาจะเป็นค่าที่ได้จากการคำนวณระหว่างค่าของ Reward ที่ได้และค่า Q-value สูงสุดของ State ถัดไป

2.2 Simulation of Urban MObility[4]

หรือที่เรียกสั้นๆว่า SUMO เป็นโปรแกรม Open source ที่ใช้ในการจำลองเส้นทางการคมนาคมที่มีประสิทธิภาพสูง SUMO ได้ถูกออกแบบมาเพื่อรองรับระบบเครือข่ายขนาดใหญ่ ได้สามารถรองรับมีความสามารถในการจำลองตั้งแต่ เส้นทางการจราจร, รถที่สัญจรบนเส้นทาง, สัญญาณไฟจราจรบริเวณแยกต่าง ๆ, การขนส่งสาธารณะ จักรยาน จักรยานยนต์ คนเดินทางเท้า รวมไปถึงความสามารถในการกำหนดนิสัยส่วนตัวของผู้ขับขี่ได้อีกด้วย

รูปที่ 2.5 Simulation of Urban Mobility Logo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2.1 Netedit

เป็นโปรแกรมที่ได้รับการติดตั้งมาพร้อมกับ โปรแกรม Sumo ซึ่งโปรแกรม Netedit มีหน้าที่เพื่อใช้ในการสร้าง แก้ไข และ จัดการ Network สำหรับใช้กับโปรแกรม SUMO โดยเฉพาะ สามารถสร้างเส้นทางการจราจรต่าง ๆ ได้อย่างมีประสิทธิภาพ อีกทั้งยังสร้าง และ กำหนดระยะเวลาของสัญญาณไฟจราจรต่าง ๆได้อย่างอิสระ กำหนดทิศทางในการเดินทางของยานพาหะนะ โดยจะมี Output ออกมาในลักษณะของ SUMO-net files ซึ่งเป็นไฟล์ที่ SUMO ไว้เรียกใช้ในการแสดงแผนที่ของการจราจร

2.2.2 OpenStreetMap

เป็นองค์กรไม่แสวงหาผลกำไร ที่ให้บริการข้อมูลด้านแผนที่ ซึ่งสามารถนำไปใช้ได้หลากหลายรูปแบบ โดยมุ่งเน้นส่งเสริมการใช้ความรู้ของคนท้องถิ่นในการปรับแก้ไขแผนที่ในแต่ละบริเวณ ซึ่งทาง SUMO สามารถ Import ไฟล์แผนที่ของ OpenStreetMap ณ บริเวณต่าง ๆที่ผู้ใช้งานกำหนดทำให้มีความสะดวกในการจำลองเส้นทางตามสถานที่จริงแต่เนื่องจาก OpenStreetMap เป็นองค์ไม่แสวงหาผลกำไร จึงทำให้ความละเอียด และ ความถูกต้องของแผนที่นั้นยังไม่เยอะมากนัก โดยผู้ใช้งานสามารถแก้ไขข้อบกพร่อง ด้วยการทำงานร่วมกับโปรแกรม Netedit ที่กล่าวมาข้างต้นเพื่อทำการแก้ไขเส้นทางจราจรต่าง ๆ ตามที่ต้องการ

2.2.3 Traffic Control Interface

เรียกสั้นๆว่า Traci เป็นการเข้าถึงการจำลองของ SUMO โดยมีลักษณะการทำในแบบ TCP Client และ Server โดยตัวโปรแกรม SUMO จะทำหน้าที่เป็น Server และผู้ใช้งานเป็น Client โดย Traci สามารถทำงานร่วมกับ Python โดยผู้ใช้งานสามารถเรียกใช้ API ของ Traci เพื่อทำการดึงค่าต่าง ๆที่ทำการจำลองอยู่เช่น เวลารอของยานพาหนะแต่ละคัน ความเร็วรถ อัตราเร่ง ขนาด ค่าของก๊าซคาร์บอนไดออกไซด์ที่ปล่อยออกมาจากรถ ไอดีของยานพาหนะ ไอดีของเส้นทางการจราจร เป็นต้น อีกทั้งผู้ใช้งานยังสามารถกำหนดค่าต่าง ๆผ่าน Traci ได้ เช่น ความเร็วรถแต่ละคัน ระยะเวลาของไฟจราจร กำหนดความเร็วรถ และอื่น ๆ จึงทำให้เกิดความสะดวกในการนำ SUMO มาใช้เป็นโปรแกรม Simulation เพื่อนำข้อมูลที่ดึงผ่าน API ของ Traci นำมารับส่งเข้าไปยังกระบวนการ Reinforcement Learning ทำการควบคุมปรับเปลี่ยนระยะเวลาของสัญญาณไฟจราจร

2.2.4 XML

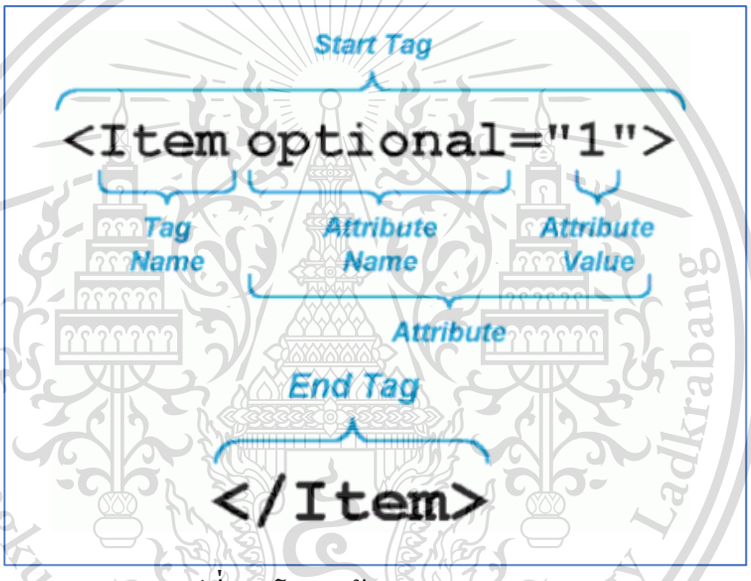
ในการใช้งาน SUMO ทั้งการเปิดไฟล์แผนที่ของการจราจร การควบคุมไฟจราจร การกำหนดเส้นทาง หรือ การกำหนดค่าของยานพาหนะ การใช้งานโปรแกรม Netedit ในการสร้างเส้นทางการจราจร จะทำงานร่วมกับไฟล์ที่เป็นภาษา XML ทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.2.4.1 XML คืออะไร

โดย XML ย่อมาจาก eXtensible Markup Language ซึ่งเป็นภาษาสำหรับใช้ในการเขียนงานเอกสาร Markup โดยเอกสารประเภท Markup จะมีการใช้ Tags เปิด-ปิด เพื่อบอกหน้าที่และประเภทของข้อมูลที่จะเขียนเพื่อให้เกิดความชัดเจนของหน้าที่ส่วนนั้น ๆ ของเอกสาร XML เป็นภาษาที่เหมาะสมกับการแลกเปลี่ยนข้อมูลผ่าน Network เนื่องจากภาษา XML นั้นมีความยืดยุ่นละไม่ขึ้นอยู่กับระบบปฏิบัติการใด ๆ นอกจากความเข้าใจง่ายตามที่กล่าวมาข้างต้นแล้ว ผู้ใช้งานยังสามารถกำหนดและตั้งค่าTagsให้เหมาะสมกับเอกสารต่าง ๆได้อย่างอิสระและยังสามารถเพิ่มเติม Tags ใด ๆได้ในภายหลังโดยจะไม่มีผลกระทบต่อเอกสารนั้น ๆ อีกด้วย

2.2.4.2 โครงสร้างของภาษา XML



รูปที่ 2.6 โครงสร้างของภาษา XML

จากรูปที่ 2.6 จะเห็นได้ว่าภาษา XML จะแบ่งเป็น 2 ส่วนหลัก ๆ คือ Start Tag และ End Tag ซึ่งจะเรียกว่า Element โดย Start Tag จะแบ่งออกเป็น 2 ส่วนย่อย ๆ คือ

1) Tag Name จะเป็นชื่อของ Element นั้น ๆมักจะตั้ง Tag Name ด้วยความหมายที่อธิบายถึงหน้าที่ของ Element นั้น ๆ
2) Attribute เป็นข้อมูลภายใน Markup ซึ่งจะแบ่งออกได้เป็น 2 ส่วนคือ
    1) Attribute Name ชื่อของ Attribute นั้น ๆ
    2) Attribute Value ค่าของ Attribute นั้น ๆ

โดยปกติแล้ว XML Attribute จะมีได้ 1 value ของแต่ละ Attribute และ 1 Element สามารถมีได้หลาย Attribute

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.3 Python

รูปที่ 2.7 Python Logo

ภาษา Python เป็น ภาษาโปรแกรมคอมพิวเตอร์ระดับสูง ซึ่ง ภาษา Python ถูกสร้างขึ้นเป็นภาษาสคริปต์ที่เข้าใจได้ง่าย ตัดความซับซ้อนของไวยกรณ์ของภาษาออกไป ภาษา Python พัฒนาแบบไม่ยึดติดแพลตฟอร์ม หรือก็คือการที่ผู้ใช้งานสามารถใช้งานได้ทั้งในระบบ Unix, Linux, Windows Python ถูกสร้างโดยมีพื้นฐานจากภาษา ซี มีระบบการประผลแบบ Interpreter ซึ่งคือการแปลชุดคำสั่งทีละบรรทัด เพื่อส่งคำสั่งข้าสู่หน่วยประมวลผลของคอมพิวเตอร์ ด้วยเหตุผลที่ Python เป็นภาษาแบบ Open Source ทำให้ทุกคนสามารถใช้ Python ในการพัฒนาโปรแกรมโดยไม่มีค่าใช้จ่าย และ ทำให้ Python มีความสามารถที่สูง Community กว้างอย่างมากในปัจจุบัน และสามารถพัฒนาโปรแกรมได้หลากหลายประเภท

2.3.1 ความเป็นมาของภาษา Python

ประวัติความเป็นมาของภาษา Python เกิดขึ้นเมื่อเดือน ธันวาคม ปี 1989 โดยโปรแกรมเมอร์ชาวดัตช์ ชื่อนาย Guido van Rossum ขณะนั้นเขาได้ทำงานอยู่ที่ สถาบันวิจัยทางคณิตศาสตร์และวิทยาการคอมพิวเตอร์แห่งชาติ Centrum Wiskunde & Informatica (CWI) นาย Guido รู้สึกว่าภาษาที่มี ณ ขณะนั้นเช่น ABC, C และ Bourne shell มีปัญหาและข้อจำกัดต่าง ๆที่เยอะเกินไป จึงเริ่มทำการพัฒนาภาษาระดับสูงขึ้นมาใหม่เพื่อใช้งานเอง ภาษา Python จึงมีพื้นฐานมาจากภาษา C และ ภาษา ABC ในเวลาต่อมาจึงได้เปิดตัว Python 1.0 ซึ่งเป็น version แรกในปี 1994 โดยชื่อของภาษานั้นไม่ได้มาจาก Python Snake เหมือนที่เข้าใจกัน แต่มาจาก Monty Python's Flying Circus รายการโทรทัศน์ที่กำลังฉาย ณ ตอนนั้น โดยนาย Guido ได้กล่าวว่าคำว่า Python นั้นเป็นชื่อที่สั้น จำได้ง่าย แตกต่างจากชื่อภาษา ณ ขณะนั้นจึงได้ตั้งชื่อ Python แก่ภาษาระดับสูงที่ตนเป็นคนพัฒนาขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

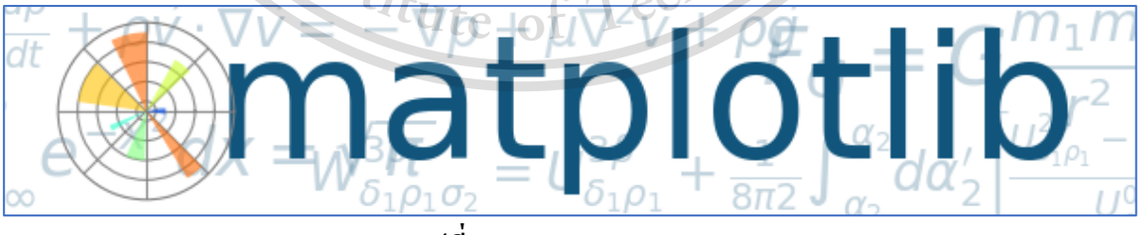
2.3.2 Library Matplotlib

เป็น Library พื้นฐานในการทำ Data Analysis มีต้นกำเนิดมาคำสั่งแสดงกราฟฟิกของ MATLAB แต่ไม่ได้เป็นส่วนนึงของ MATLAB สามารถใช้งานในลักษณะของการเขียนโปรแกรมเชิงวัตถุ หรือ Object Oriented Programming(OOP) ได้โดยส่วนใหญ่ Matplotlib จะเขียนด้วย Python เป็นหลัก แต่ผู้ใช้งานยังสามารถใช้งานร่วมกับ Library NumPy หรือ Library อื่น ๆเพื่อให้ได้ประสิทธิภาพที่ดียิ่งขึ้นการแสดงผลด้วย Array ที่มีขนาดใหญ่

Matplotlib ได้สร้างขึ้นโดยนาย John D. Hunter เมื่อปี 1968 เนื่องจากเขาต้องการทำงานระหว่าง MATLAB และ EEG ซึ่ง MATLAB นั้นมีข้อจำกัดมาก จึงได้เริ่มทำการพัฒนาด้วยภาษา Python ในเวลาต่อมา Matplotlib นั้นได้ถูกแบ่งออกเป็น 3 ส่วนคือ

1) The Pylab Interface หรือก็คือชุด Function ที่จัดทำโดย Pylab ซึ่งอนุญาตให้ผู้ใช้งานสร้างการ Plots ที่มีลักษณะเหมือนกับ MATLAB ได้
2) The Matplotlib frontend or Matplotlib API คือชุดของ Classes ที่ทำหน้าที่เป็นส่วนติดต่อของผู้ใช้งาน สามารถจัดการข้อความ ตัวเลข Plots และ อื่น ๆ เพื่อใช้ผู้ใช้งานได้สร้างการ Visualization ออกมาให้รูปแบบต่าง ๆได้ตามความต้องการ
3) The Matplotlib backends เป็นส่วนที่ทำการสร้างภาพที่ทำการแสดงออกมา เนื่องจาก Matplotlib เป็น Library ใน Python ซึ่งหมายความว่ามีความหลากหลายของผู้ใช้งาน ผู้ใช้งานอาจจะใช้งานใน Python Shell, Jupyter Notebook หรือแม้กระทั้งใช้งานใน Batch scripts เพื่อสร้างภาพ Postscript จากการจำลองตัวเลข ดังนั้นจึงจำเป็นต้องมี backends เพื่อลองรับกรณีการใช้งานที่หลากหลาย

ในปัจจุบันผู้พัฒนาหลักคือ นาย Thomas A.Caswell และมีผู้ใช้งานและนักพัฒนาที่เพิ่มมากขึ้นเรื่อย ๆ จึงทำให้ Matplotlib เป็น Library ที่ใช้แสดง Visualization ที่ได้รับความนิยมมากในปัจจุบัน



รูปที่ 2.8 Matplotlib Logo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.4 Traffic Engineering[3]

Traffic Engineering คือ สาขาหนึ่งของวิศวกรรมขนส่งซึ่งจะศึกษาเกี่ยวกับการวางแผน และ ทำการออกแบบทางเรขาคณิต และ การควบคุมการจราจรบนท้องถนน ทั้งบริเวณถนนหลัก และ ถนนย่อย ทางหลวง ระบบเครือข่ายของถนน พื้นที่ที่อยู่รอบ ๆ บริเวณถนน อีกทั้งยังศึกษาเกี่ยวกับพฤติกรรมบนท้องถนนของผู้ใช้ยานพาหนะอีกด้วย

เนื่องจากการที่ยานพาหนะแต่ละคันที่อยู่บนท้องถนนได้เคลื่อนตัวบนถนนนั้นเมื่อทำการรวมกันเป็นจำนวนมาก จึงก่อให้เกิดคำที่เรียกปรากฏการณ์แบบนี้ว่า กระแสการจราจร หรือ Traffic stream ซึ่งยานหนะที่ได้อยู่ในกระแสการจราจรเดียวกันจะมีความสัมพันธ์ต่อเนื่องกัน โดยตัวอย่างของความสัมพันธ์ต่อเนื่องในกระแสการจราจร เช่น เมื่อยานพาหนะที่อยู่ด้านหน้าทำการเบลค ก็จะส่งผลให้ยานพาหนะต้องเบลคตามกัน หรือ การลดลงของความหนาแน่นในกระแสจราจร ทำให้ยานพาหนะด้านหน้าทำการเร่งความเร็ว อาจส่งผลให้ยานพาหนะที่อยู่ด้านหลังทำการเร่งความเร็วเพิ่มขึ้น

2.4.1 ตัวแปรที่ใช้อธิบาย

การวิเคราะห์ กระแสการจราจร จำเป็นที่จะต้องมีพื้นฐาน และ ความเข้าใจในลักษณะของสภาพการจราจรที่แตกต่างกัน โดยจะมีตัวแปรสำคัญที่แสดงถึงลักษณะของสภาพการจราจร ณ ขณะนั้นดังนี้ ความเร็ว(Speed) อัตราการไหลของกระแส(Flow Rate) ความหนาแน่นของสภาพการจราจร(Density)

2.4.1.1 ปริมาณและอัตราการไหลของกระแส (Traffic volume and Flow Rate)

ความหมายของปริมาณการจราจร คือ จำนวนของยานพาหนะที่เคลื่อนที่ผ่านจุดอ้างอิงที่กำหนดบนช่องทางการจราจร โดยปกติปริมาณการจราจรจะมีหน่วยเป็น คันต่อหน่วยเวลา ส่วนความหมายของอัตราการไหลของกระแสการจราจร จะมีหน่วยเป็น คันต่อชั่วโมง

2.4.1.2 ความเร็วและเวลาในการเดินทาง (Speed and Travel time)

ความเร็ว คือ อัตราการเคลื่อนที่ในระยะทางต่อ 1 หน่วยเวลา หรือ ส่วนกลับของเวลาที่ใช้ในการเคลื่อนที่ไปในระยะทางที่กำหนด คูณด้วยระยะทางนั้น เขียนสมการได้ดังนี้

$$s = \frac{d}{t} \tag{3.1}$$

โดย s = ความเร็วหน่วยเป็น ไมล์ต่อชั่วโมง กิโลเมตรต่อชั่วโมง หรือ ฟุตต่อวินาที

d = ระยะทางที่เดินทางได้หน่วยเป็น ไมล์ กิโลเมตร ฟุต

t = เวลาที่ใช้ในการเดินทางหน่วยเป็น ชั่วดมง หรือ วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.4.1.3 ความหนาแน่นในกระแสการจราจร (Density)

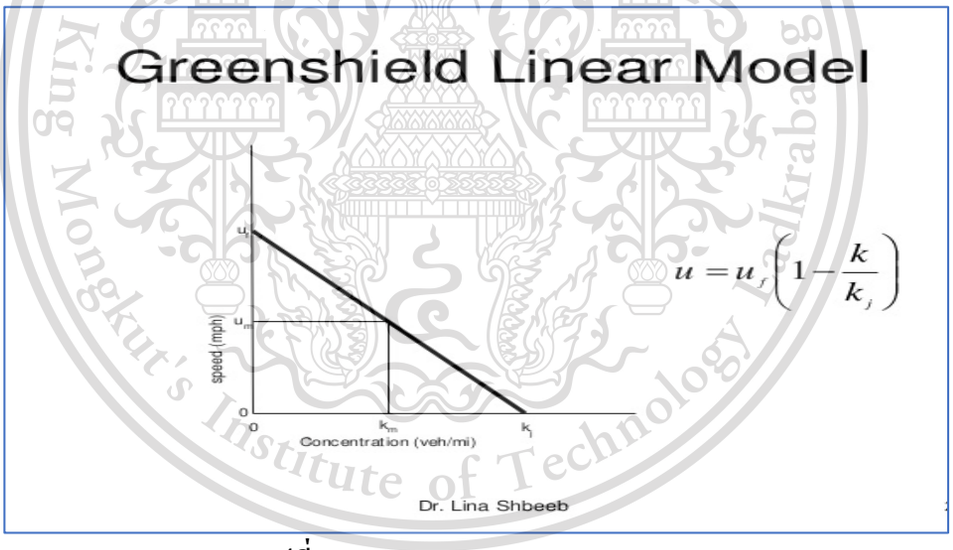
ความหนาแน่น คือ จำนวนของยานพาหนะที่อยู่บนพื้นผิวจราจรในช่วงของความยาวที่กำหนด โดยมีหน่วยเป็น คันต่อไมล์ (vpm) หรือ คันต่อไมล์ต่อช่องจราจร (vpmpl) โดยความหนาแน่นนั้นเป็นค่าที่วัดโดยตรงได้ยาก โดยสามารถวัดทางอ้อมได้โดยการคำนวนจากค่าความเร็วเฉลี่ยและอัตราการไหล

2.4.2 ความสัมพันธ์พื้นฐานของตัวแปร

จากการสำรวจข้อมูลของกระแสจราจร เพื่อทำการสร้าง Greenshields Model ซึ่งสรุปได้โดยจะมีความสัมพันธ์ได้อยู่ 3 รูปแบบคือ

1) กราฟแสดงความสัมพันธ์ระหว่างความเร็วและความหนาแน่น
2) กราฟแสดงความสัมพันธ์ระหว่างความเร็วและอัตราการไหล
3) กราฟแสดงความสัมพันธ์ระหว่างอัตราการไหลและความหนาแน่น

โดยในเบื้องต้นได้กล่าวไว้ว่าให้พิจารณาจากกราฟแสดงความสัมพันธ์ระหว่างความเร็วและความหนาแน่นซึ่งเป็นกราฟเส้นตรง



รูปที่ 2.9 Greenshields Linear model

จากรูปที่ 2.9 จะมีจุดที่เป็นความสำคัญหลักๆอยู่ 2 จุดคือ

1) จุดตัดแกน Y จะเรียกว่า “Free flow Speed” ซึ่งเป็นจุดที่มีค่าความเร็วเท่ากับ Uf (S) เมื่อมีค่าความหนาแน่นเป็นศูนย์
2) จุดตัดแกน X จะเรียกว่า “Jam density” ซึ่งเป็นจุดที่มีค่าความหนาแน่นเป็น Kj (D)ซึ่งจะเป็นจุดที่ทำให้ไม่มีการเคลื่อนที่กล่าวคือค่าความเร็วเป็นศูนย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

จากจุดตัดแกน Y หรือ “Free flow Speed” และ จุดตัดแกน X หรือ “Jam density” จะสามารถสร้างความสัมพันธ์ออกมาเป็นสมการได้ดังนี้

$$S = Uf(1 - \frac{D}{Kj}) \tag{3.2}$$

ต่อจากสมการที่ 3.2 จะพิจารณาสมการระหว่างความสัมพันธ์นี้

$$V = S * D \tag{3.3}$$

โดย V = อัตราการไหล

D = ความหนาแน่น

S = อัตราเฉลี่ยของความเร็วในระบบ

จากความสัมพันธ์ของสมการที่ 3.2 และ 3.3 จะสามารถแทนค่าของความเร็วและแทนค่าของความหนาแน่นได้ เพื่อทำให้เกิดกราฟแสดงความสัมพันธ์ระหว่างความเร็วและอัตราการไหลและกราฟแสดงความสัมพันธ์ระหว่างอัตราการไหลและความหนาแน่นตามลำดับ

โดยแทนค่า $S = \frac{v}{D}$ ลงไปในสมการที่ 3.2 จะได้ดังนี้

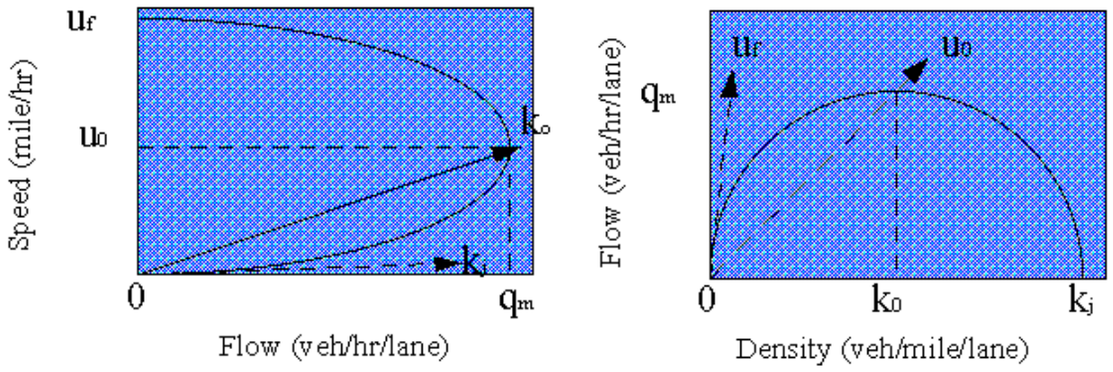
$$V = UfD - \frac{UfD^2}{Kj} \tag{3.4}$$

โดยแทนค่า $D = \frac{v}{S}$ ลงไปในสมการที่ 3.2 จะได้ดังนี้

$$V = KfS - \frac{kfS^2}{Uj} \tag{3.5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

จากสมการที่ 3.4 และ 3.5 จะนำมาแสดงเป็นกราฟพาราโบลาเพื่อแสดงความสัมพันธ์



รูปที่ 2.10 กราฟพาราโบลาแสดงความสัมพันธ์ของสมการ 3.4 และ 3.5

โดยค่าการรองรับความจุ (Capacity) เป็นค่า peak ของกราฟแสดงความสัมพันธ์ระหว่างความเร็วและอัตราการไหลหรือกราฟแสดงความสัมพันธ์ระหว่างอัตราการไหลและความหนาแน่น ซึ่งเราสามารถคำนวณหาค่าการรองรับความจุได้ในเชิงคณิตศาสตร์ โดยค่าการรองรับความจุจะเกิดขึ้นเมื่อความชันของกราฟแสดงความสัมพันธ์เป็นศูนย์

พิจารณาจากกราฟแสดงความสัมพันธ์ระหว่างอัตราการไหลและความหนาแน่น

$$V = UfD - \frac{UfD^2}{Kj}$$

$$\frac{dV}{dD} = 0 = Uf - \frac{2UfD}{Kj}$$

$$D = \frac{Kj}{2} \qquad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

พิจารณาจากกราฟแสดงความสัมพันธ์ระหว่างความเร็วและอัตราการไหล

$$V = KfS - \frac{KfS^2}{Uj}$$

$$\frac{dV}{dS} = 0 \ = Kf - \frac{2KfS}{Uj}$$

$$S = \frac{Uj}{2} \tag{3.7}$$

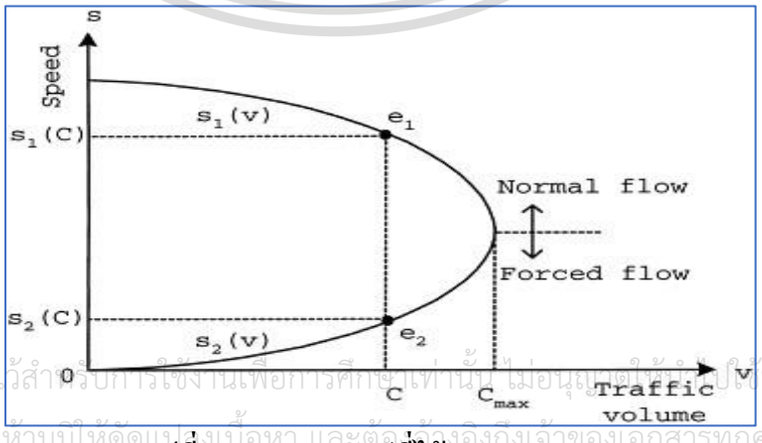
จากสมการที่ 3.6 และ 3.7 จะสรุปได้ว่าค่าการรองรับความจุจะเกิดขึ้นเมื่อค่าความเร็วมีค่าเป็นครึ่งนึงของค่า Free flow speed และเมื่อค่าความหนาแน่นเป็นครึ่งนึงของค่า Jam density

ดังนั้นจะได้สมการของค่าการรองรับความจุ

$$C = \frac{Uf}{2} * \frac{Kj}{2} \tag{3.8}$$

ค่าการรองรับความจุนั้นสามารถตีความหมายได้สองความหมายในค่าอัตราการไหล โดยอ้างอิงจากกราฟแสดงความสัมพันธ์ระหว่างความเร็วและอัตราการไหล คือ

1) การจราจรจะมีสภาพคล่องตัวที่อัตราเฉลี่ยของความเร็วในระบบจราจรสูงกว่าครึ่งนึงของค่า Free flow speed
2) การจราจรจะมีสภาพติดขัดที่อัตราเฉลี่ยของความเร็วในระบบจราจรต่ำกว่าครึ่งนึงของค่า Free flow speed



รูปที่ 2.11 แสดงกราฟที่ได้จากสมการ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2.6 เอกสารงานวิจัยที่เกี่ยวข้อง

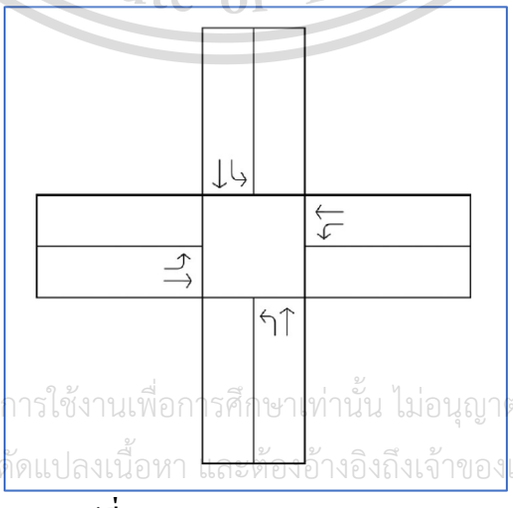
2.6.1 Traffic signal control for smart cities using Reinforcement learning [1] (TSCRL)

ปริญญานิพนธ์เล่มจะมีการอ้างอิงถึงงานวิจัย Traffic signal control for smart cities using Reinforcement learning [1] (TSCRL) ซึ่งทำการวิจัยโดย Hyunjin Joo, Yujin Lim จาก Sookmyung Women University, Korea และ Syed Hassan Ahmed จาก Georgia Southern University, USA โดยจะเป็นเอกสารงานวิจัยหลักที่จะใช้ในการวัดและประเมินผลการทดลองในโครงงานครั้งนี้

งานวิจัยนี้พูดถึงการได้พิจารณาของปัญหาในการเพิ่มประสิทธิภาพของสัญญาณไฟจราจรโดยมีวัตถุประสงค์ในการลดความล่าช้าของการจราจร และ ให้ความสำคัญในการรจัดการยานพาหนะบริเวณทางแยกอย่างเท่าเทียมกันเพื่อลดความล่าช้าที่เกิดขึ้นบริเวณแยกจราจร โดยจะไม่พิจารณาเวลารอของยานพาหนะบริเวณแยกอื่น ๆ โดยค่าที่สามารถแสดงได้ถึงไฟจราจรที่มีความเท่าเทียมกันในแต่ละฝั่งนั้นคือ ส่วนเบี่ยงเบนมาตรฐานของความยาวคิว ซึ่ง ความยาวของคิวคือจำนวนรถที่รออยู่ด้านใดด้านหนึ่งของถนน โดยค่าส่วนเบี่ยงเบนมาตราฐานยิ่งมีค่าน้อยยิ่งแสดงถึงความเท่าเทียมกัน

2.6.1.1 การ Implement Q-Learning

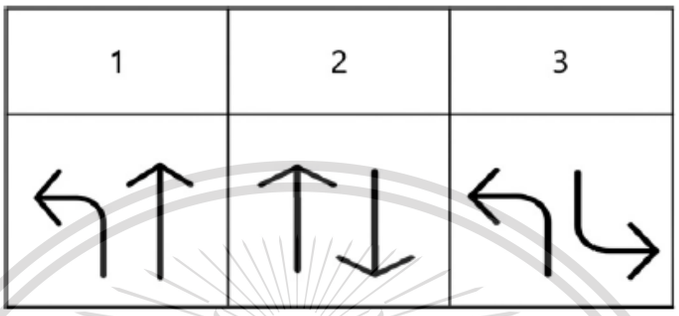
1) State คือ การกำหนด Environment เป็นบริเวณ 4 แยกไฟจราจรโดยถนนจะสามารถเดินทางได้ 2 ทิศทางคือ เลนซ้ายจะเป็นการเลี้ยงซ้ายและเลนขวาคือการตรงไป ดังนั้นจะกล่าวแทนได้ว่า Environment ที่แยกไฟจราจร n แยกจะมีสถานะทั้งหมด 2 * n สถานะโดยกำหนดให้ทิศทางที่มียานพาหนะมากที่สุดจะเป็น สถานะปัจจุบัน



รูปที่ 2.12 State and Environment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only. Not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

2) Action คือ การกระทำที่สามารถนำรถออกจากระบบได้มากที่สุดโดยจะถูกเลือกจาก Action set ของ State ปัจจุบัน โดยในงานวิจัยนี้จะแทน Action ด้วยการปล่อยรถออกเป็นทั้งหมด 3 รูปแบบตามภาพต่อไปนี้



รูปที่ 2.13 Action set

3) Reward คือ รางวัลที่ลดความล่าช้าบริเวณแยกไฟจราจรโดยในงานวิจัยนี้จะกำหนดค่าของ Reward ด้วยค่า 2 ค่า คือ ส่วนเบี่ยงเบนมาตรฐานของความยาวคิว และ throughput ซึ่งจะนำมารูปแบบของสมการ Exponential โดยจะมีค่าอยู่ระหว่าง 0 ถึง 1 เพื่อนำไปใช้ในการคำนวณ Reward ด้วย Sigmoid Function เพื่อให้มีค่าอยู่ระหว่าง 0 ถึง 1

$$\mathrm{f(t)} = \alpha \cdot \left(d_{ql}\right) + (1-\alpha) \cdot \left(\tau^{tp}\right) \qquad (3.9)$$

$$r_t = \log_{\delta}(f(t)) \qquad (3.10)$$

โดยที่ d_{ql} = ส่วนเบี่ยงเบนมาตรฐานของความยาวแถว

τ^{tp} = Throughput ที่เข้าสมการ Exponential

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

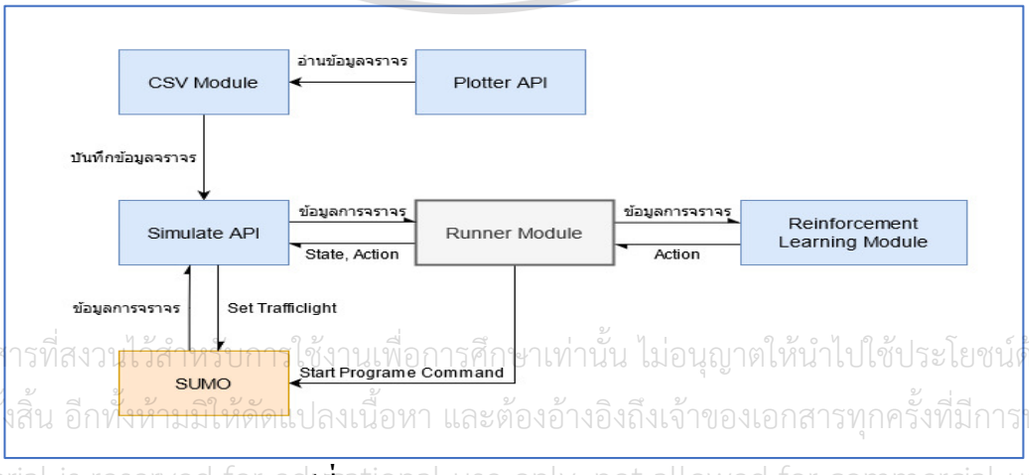
บทที่ 3

การออกแบบและการพัฒนา

3.1 ภาพรวมระบบ

ในการพัฒนาโมเดล Reinforcement Learning เพื่อใช้สำหรับการควบคุมสัญญาณไฟจราจรนั้น ระบบจะถูกแบ่งออกเป็นส่วนๆ เพื่อแบ่งแยกหน้าที่ในการทำงาน และ นำแต่ละส่วนมาประกอบการทำงานเข้าด้วยกัน โดยการออกแบบ ระบบจะถูกแบ่งออกเป็น 4 ส่วนหลัก ๆได้แก่ Runner, Reinforcement Learning, Simulation API, CSV Module, Plotter โดยที่ส่วนของ Runner จะเป็นตัวกลางที่เชื่อมโยงและเรียกใช้แต่ละส่วนเข้าด้วยกัน

การทำงานของระบบจะเริ่มจากการ Run ไฟล์ Runner จะทำการสั่งให้เปิดตัวโปรแกรม SUMO ที่มีการ Config ENV ต่างๆ เช่น แผนที่ เส้นทางการวิ่งของยานพาหนะ รวมไปถึงรูปแบบการแสดงผล GUI (graphical user interface) ของตัวโปรแกรม SUMO ผ่านทาง XML File ตัวของไฟล์ Runner นั้นจะมีการเชื่อมต่อกับส่วนของ Simulation API เพื่อใช้ในการติดต่อสื่อสารกับตัว Program SUMO ผ่าน Traci โดยการควบคุมสัญญาณไฟจราจร และ เก็บค่าผลลัพธ์จากการ Simulation ที่ส่งมาจากโปรแกรม SUMO และนำค่าเหล่านั้นส่งไปยัง CSV Module และ Reinforcement Learning ในส่วนของ CSV Module โดย CSV Module นี้นั้นจะเป็นที่ใช้ในการจัดการข้อมูลกับ csv file ทำการอ่าน และ เขียน ข้อมูลเพื่อบันทึกเก็บไว้และนำไปใช้งานต่างๆ และ ส่วนของ Plotter นั้นจะทำการอ่านค่าผลลัพธ์จาก SUMO ผ่านไฟล์ csv และ แสดงผลออกมาในลักษณะของกราฟ ในขณะที่ข้อมูลการจราจรที่ส่งมายัง Reinforcement Learning นั้นก็จะนำมาคิดค่าของ Reward เพื่อนำไปคำนวณค่าของ Q-value ด้วย Algorithm ที่แตกต่างกันในแต่ละวิธี เพื่อใช้ในการเลือก Action ต่างๆให้เหมาะสมกับ State ณ ขณะนั้น



รูปที่ 3.1 Diagram การทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โดยระบบนั้นได้ทำการออกแบบทำงานในลักษณะของการวน Loop จนกระทั้งข้อมูลของยานพาหนะภายในไฟล์ XML ในระบบนั้นหมดลง ระหว่างการทำงานของระบบนั้น ในส่วนทางด้านของ Reinforcement Learning ก็จะทำการเรียนรู้ และ ปรับค่าของ Q-value ภายใน Q-table จนกระทั่งจบการทำงาน ในช่วงแรกนั้นค่า Q-value จะยังมีค่าที่ไม่เสถียร ทำให้การตัดสินใจเลือก Action ของ Model นั้นประสิทธิภาพไม่สูงมากนัก และเมื่อ Model Reinforcement Learning ได้ทำการเรียนรู้เป็นระยะเวลาที่เพิ่มขึ้น ค่า Q-value ใน Q-table ก็จะลู่เข้าใกล้ค่าใดค่าหนึ่ง และ เสถียรในที่สุด ทักษะในการตัดสินใจ Action ของ Model Reinforcement Learning นั้นก็จะมีประสิทธิภาพสูงตามระยะเวลาการเรียนรู้ที่ผ่านมา

โดยสภาพแวดล้อมที่ใช้ในการทำงานและทดสอบระบบในโครงงานในครั้งนี้คือ

ตารางที่ 3.1 Specification **ของอุปกรณ์ที่ใช้ทดสอบ**

Specification	**Action**
CPU	Intel Core i7-7700HQ
GPU	NVIDIA GeForce GTX 1050 (4GB GDDR5)
Memory Size	12 GB DDR4
Hard Disk Drive	1 TB 7200 RPM
Solid State Drive	128 GB SSD

3.2 Reinforcement Learning

การออกแบบในส่วนของ Reinforcement Learning นั้น จะแบ่งออกเป็น Reinforcement Learning ที่อ้างอิงจากงานวิจัย TSCRL เพื่อใช้ในการเปรียบเทียบ และ Reinforcement Learning ที่ทางผู้จัดทำได้ทำการออกแบบใหม่ โดยโครงสร้างหลักจะมีลักษณะที่คล้ายกัน โดยมีส่วนที่แตกต่างกันคือ วิธีการคำนวณ Reward การตั้งค่าระยะเวลาของไฟเขียว และ การจัดการ State ถัดไป ซึ่งจะทำการอธิบายรายละเอียดในแต่ละส่วนในหัวข้อถัดไป

การทำงานของ Reinforcement Learning นั้นจะอ้างอิงตาม Markov Decision Process Model ที่กล่าวไปในบทที่ 2 ซึ่งจะประกอบไปด้วย State, Action, Reward และ Agent โดยจะมี State Space หรือ Q-table ทำการเก็บค่าความสัมพันธ์ระหว่าง State, Action และ Q-value เอาไว้เพื่อทำการบันทึกผลของกระบวนการเรียนรู้ที่ผ่านมาเพื่อนำไปใช้งานต่อไป โดยการทำงานของ Reinforcement Learning จะทำงานบริเวณระบบไฟจราจร ที่ซึ่งภายในแผนที่ที่ใช้ในการ Simulation เดียวกันจะมีการใช้ Model Reinforcement Learning และ State Space เดียวกัน โดยที่แต่ละแยกจะมีการเก็บ State และ Action เป็นของตัวเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.2.1 State

ในการออกแบบ State ของระบบ จะทำการแทนที่ State ด้วยชื่อของเลนที่มีทิศทางเข้าสู่ระบบ โดยที่ถนนหนึ่งเส้นได้มีการออกแบบให้มีลักษณะ 2 เลน โดยเลนทางด้านซ้ายของถนนจะมีทิศทางให้ยานพาหนะตรงไปและเลี้ยวซ้าย ส่วนเลนทางด้านขวาจะมีทิศทางบังคับให้ยานพาหนะเลี้ยวขวา โดยในส่วนของการทำงาน Reinforcement Learning นั้น ภายในแผนที่ที่ใช้ในการ Simulation จะมีการใช้ State Space เดียวกัน โดยที่แต่ละแยกไฟจราจร จะมีการเก็บ State เป็นของตัวเอง ดังนั้นจึงทำให้ค่าของ State ของแต่แยกไฟจราจรในระบบนั้นมีจำนวนที่เท่ากันทุกแยก โดยที่จำนวน State ที่เป็นไปได้ทั้งหมดนั้นคือ 8 State ดังรูป

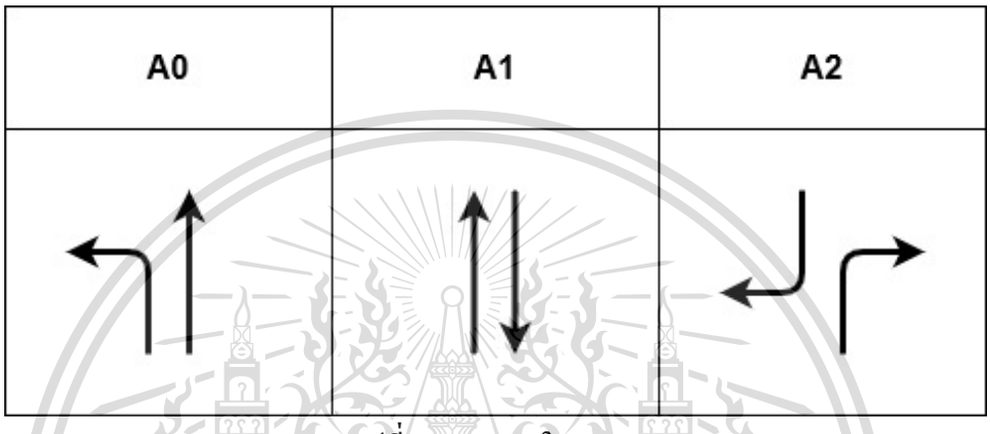
รูปที่ 3.2 สถานะ State ของระบบ

โดยกระบวนการ การเลือก Next State นั้น ได้มีการออกแบบให้มีการคัดเลือก Next State จาก Set ของ State ที่ไม่ได้ทำการถูกเลือกให้เปิดไฟเขียว ณ ขณะนั้น โดยเมื่อจบ Cycle การทำงานแล้ว จะทำการเลือกจาก State ที่มีจำนวนความยาวของแถวยานพาหนะที่หยุดรอภายใน เลนแต่ละเลน ที่มากที่สุดเพื่อนำมาเป็น State ถัดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.2.2 Action

Action เป็นการกระทำที่เกี่ยวเนื่องต่อ State โดยตรงโดยทำการส่งผลในการคัดเลือก State ถัดไป โดยการออกแบบ State เป็นการออกแบบโดยให้ แต่ละ State คือเลนของถนน ตามที่กล่าวมาในข้อหัวที่แล้ว จึงทำการออกแบบให้ Action คือทิศทางที่จะทำการปล่อยไฟเขียวภายในระบบ โดย Action จะสอดคล้องและเกี่ยวเนื่องกันกับ State ปัจจุบัน โดยจำนวนของ Action จะมีทั้งหมด 3 Action ดังรูปที่ 3.3



รูปที่ 3.3 Action ในระบบ

โดยการกระทำของ Action ต่อ State ทั้ง 8 State นั้นจะต้องสอดคล้องกับทิศทางของ State ปัจจุบัน ดังนั้น State แต่ละState จะมี Action ที่ตรงกับเงื่อนไขเป็นจำนวน 2 Action จาก 3 Action เท่านั้น จึงจำนวนที่จะต้องมีส่วนในการตรวจสอบของ Action ที่ได้มาจากการเลือก Action ของกระบวนการ E-Greedy หรือ P-Greedy เสียก่อนจะนำมาใช้งานจริงเพื่อป้องกันการทำงานผิดพลาดของ Algorithm โดยความเกี่ยวเนื่องกันระหว่าง State และ Action จะสามารถอธิบายได้ตามตาราง State transition ดังนี้

ตารางที่ 3.2 State transition

Current State	Action	Next State
S0	A0	{S2,S3,S4,S5,S6,S7}
	A2	{S1,S2,S3,S5,S6,S7}
S1	A0	{S2,S3,S4,S5,S6,S7}
	A1	{S0,S2,S3,S4,S6,S7}
S2	A0	{S0,S1,S4,S5,S6,S7}
	A2	{S0,S1,S3,S4,S5,S7}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ตารางที่ 3.2 State transition(**ต่อ**)

Current State	Action	Next State
S3	A0	{S0,S1,S4,S5,S6,S7}
	A1	{S0,S1,S2,S4,S5,S6}
S4	A0	{S0,S1,S2,S3,S6,S7}
	A2	{S1,S2,S3,S5,S6,S7}
S5	A0	{S0,S1,S2,S3,S6,S7}
	A1	{S0,S2,S3,S4,S6,S7}
S6	A0	{S0,S1,S2,S3,S4,S5}
	A2	{S0,S1,S3,S4,S5,S7}
S7	A0	{S0,S1,S2,S3,S4,S5}
	A1	{S0,S1,S2,S4,S5,S6}

3.2.3 Reward

ในการออกแบบ Reward ของระบบเนื่องจาก Reward เป็นรางวัลที่เกิดจากการกระทำ Action ต่อ State ใด ๆ ดังนั้นรางวัลจำเป็นต้องได้จากผลลัพธ์ของการเปลี่ยนแปลงของสัญญาณไฟจราจร และ เนื่องจากการทดลองมีลักษณะเปรียบเทียบระหว่าง Model Reinforcement Learning ของงานวิจัย TSCRL ที่ใช้ในการอ้างอิง และ Model Reinforcement Learning ที่ได้ทำการออกแบบขึ้นมาเอง ดังนั้นค่าผลลัพธ์ที่ได้จากการ Simulate และ สมการที่นำใช้ในการคำนวณ Reward จึงมีความแตกต่างกัน

3.2.3.1 Reward ที่คำนวณโดย Hyunjin Joo, Yujin Lim และ Syed Hassan

การคำนวณ Reward ของ Model Reinforcement Learning นี้จะทำการอ้างอิงวิธีการคำนวณมาจากงานวิจัย TSCRL โดยตรง โดยการคำนวน Reward จะมีค่า Parameters จำนวน 2 ตัวที่เกี่ยวข้องกับการคำนวณซึ่งก็คือ ค่าส่วนเบี่ยงเบนมาตรฐานของความยาวแถว และ ค่า throughput โดยค่าส่วนเบี่ยงเบนมาตรฐานนั้นเป็นการบ่งบอกถึงความแตกต่างกันของความยาวแถวของยานพาหนะที่ทำการรอในแต่ละเลนของระบบ ส่วนของ throughput เป็นค่าที่วัดประสิทธิภาพของยานพาหนะที่เคลื่อนที่ผ่านบริเวณ 4แยกไฟจราจรจะได้สมการการคำนวณ Reward ในลักษณะดังต่อไปนี้

$$f(t) = \alpha \cdot (d_{ql}) + (1 - \alpha) \cdot (\tau^{tp}) \quad (3.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

$$r_t = \log_{\delta}(f(t)) \tag{3.2}$$

จากสมการที่ 3.1 ในส่วนของสมการส่วนหลังนั้นเนื่องจาก throughput จะเป็นวัดการเคลื่อนที่ผ่านระบบจึงทำให้มีค่าเป็นจำนวนมากจึงทำการเข้า Exponential function โดยที่ค่าของ Tau อยู่ระหว่าง 0 ถึง 1 ดังนั้นเมื่อ throughput มีค่าที่มากค่าที่ผ่าน Exponential function ก็จะมีค่ายิ่งน้อยลง ส่วนของค่า แอลฟา นั้นจะทำหน้าที่เป็น weighting factor โดยเปลี่ยนแปลงขึ้นอยู่กับจำนวนของยานพาหนะที่มาถึงในระบบ โดยค่าของ แอลฟา จะมีค่าอยู่ระหว่าง 0 ถึง 1 และอยู่ในลักษณธการคำนวณแบบ sigmoid function ยิ่งจำนวนยานพาหนะมีค่ามาก ค่าของแอลฟา ก็จะเข้าใกล้ 1 กล่างโดยสรุปของสมการที่ 3.1 แล้วคือ ยิ่งค่าของ ส่วนเบี่ยงเบนมาตรฐานของความยาวแถว และ ค่าในส่วนของ throughput ที่ผ่าน Exponential function มีค่าที่น้อยก็จะส่งผลให้ Reward มีค่าที่มากนั้นเอง

3.2.3.1 Reward ของปริญญานิพนธ์

Reward ที่ได้ทำการออกแบบนั้นจะมีค่าของ Parameters เป็นจำนวน 2 ตัวที่ส่งผลต่อการคำนวณ Reward โดยเป็นค่าของผลลัพธ์ Flow Rate และ Waiting Time เฉลี่ยของยานพาหนะในระบบ ที่ได้จากการ Simulation ในแต่ละ Cycle โดยหลักการคิดคำนวนนั้นจะอ้างอิงจากความหมายของ Parameters ทั้ง 2 นั้นก็คือ

1) Flow Rate ยิ่งค่าของ Flow Rate มีค่าที่สูงแสดงว่ารถสามารถเคลื่อนที่ออกจากระบบได้ดี แต่การที่ Flow Rate เพิ่มมากขึ้นไปจนถึงจุดอิ่มตัว จำทำให้เกิดเหตุการ์ณ Congestion ดังนั้นค่าของ Flow Rate จึงไม่ควรมีค่าเกิดกว่าจุด Peak ของกราฟ
2) Waiting Time ยิ่งค่าเวลารอเฉลี่ยของยานพาหนะในระบบมีค่าน้อยจะยิ่งดี แสดงว่ายานพาหนะในระบบนั้นมีการเคลื่อนตัวออกจากระบบได้ดีใช้เวลารอที่น้อย แต่ในทางกลับกันทางค่าเฉลี่ยของ Waiting Time ยิ่งมีค่ามากยิ่งไม่ดีเนื่องจากยานพาหนะในระบบรอนาน

จากหลักการคิดที่ได้กล่าวมาให้ 2 ข้อย่อยข้างต้น การสร้างสมการที่ใช้ในการคำนวน Reward ได้ในลักษณะดังนี้

$$Reward = \frac{Flow\ Rate}{1+Waiting\ Time} \tag{3.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โดยค่าของ Flow Rate และ Waiting Time นั้นไม่ได้มีค่าใน Scale เดียวกันจึงจำเป็นที่ต้องเริ่มจากการ Scaling Value ของค่า Parameters ทั้ง Flow Rate และ Waiting time ให้มีค่าอยู่ระหว่าง 0 ถึง 1 ก่อน โดยผ่านการใช้ Sigmoid Function ดังนี้

$$Scaling\ FlowRate = \frac{1}{1+e^{A2(FR-A3)}} \tag{3.4}$$

$$A3 = \frac{Saturate\ FlowRate}{2} \tag{3.5}$$

$$Scaling\ Waiting\ Time = \frac{1}{1+e^{A2(WT-A3)}} \tag{3.6}$$

$$A3 = \frac{Maximum\ Waiting\ Time}{2} \tag{3.7}$$

โดยที่ FR = Flow Rate (veh/min)

WT = Average Waiting Time (sec)

ค่าของตัวแปร A2 นั้นจะเป็นค่า Slope ที่เหมาะสมของสมการ Sigmoid โดยจะมีขั้นตอนการคำนวนตามสมการที่อ้างอิงจากงานวิจัย “Calculation of threshold and saturation points of sigmoidal baroreflex function curves” [2] ดังต่อไปนี้

$$Saturate\ 5\% = A3 + \frac{2.944}{A2} \tag{3.8}$$

$$A2 = \frac{2.944}{0.95 - A3} \tag{3.9}$$

จะเห็นได้ว่าในการคำนวณ Reward นั้นจำเป็นที่จะต้องหาค่าของ A3 จากสมการที่ 3.5 และ 3.7 ให้ได้ก่อนจึงจะสามารถหาค่าอื่นๆ ได้ โดยวิธีการหาค่าของ Maximum Waiting Time และ Saturate Flow Rate นั้นจะกล่าวในบทของการทดลองต่อไป โดย Saturate Flow Rate นั้นจะสามารถหาได้จากสมการดังนี้

$$Saturate\ FlowRate = Density * Average\ Speed \tag{3.10}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only. Commercial use, Forbidden to modify the content, and cite the document when use.

3.2.4 State Space

เป็นตารางซึ่งมีหน้าที่เก็บค่า ความสัมพันธ์ระหว่าง State Action และ Q-Value โดยเนื่องจากภายใน Algorithm มีการใช้งาน ค่า Q-max และ Q-Sum ของ Q-Value ด้วยจึงออกแบบให้เก็บทั้ง 2 ค่านี้ภายใน State Space ด้วยเพื่อความสะดวกในการใช้งาน โดย Q-Value จะถูกเก็บอยู่ในลักษณะของ Array ที่มีสมาชิกตามจำนวนของ Action โดยมีทั้งหมด 3 Action ซึ่งแต่ละ Index ก็จะแทนด้วย Action ที่กระทำนั้น ๆ จึงได้ State Space ในลักษณะดังนี้

ตารางที่ 3.3 การเก็บค่าภายใน State Space

State	Q-Value	Q-Sum	Q-Max
S0	[0.0,0.0,0.0]	0.0	0.0
S1	[0.0,0.0,0.0]	0.0	0.0
S2	[0.0,0.0,0.0]	0.0	0.0
S3	[0.0,0.0,0.0]	0.0	0.0
S4	[0.0,0.0,0.0]	0.0	0.0
S5	[0.0,0.0,0.0]	0.0	0.0
S6	[0.0,0.0,0.0]	0.0	0.0
S7	[0.0,0.0,0.0]	0.0	0.0

โดยตาราง State Space นี้เป็นตารางที่ใช้เก็บค่าของ Q-value เพื่อเป็นการอ้างอิงถึงผลของกระบวนการ การเรียนรู้ของตัว Model Reinforcement Learning ดังนั้นเมื่อทำการเรียนรู้แต่ละครั้งเสร็จสิ้นแล้ว จึงจำเป็นจะต้องมีการบันทึกข้อมูลของตาราง State Space นี้เพื่อเป็นการเก็บและนำเอาไปใช้ในครั้งต่อไปในการทำงาน ดังนั้นจึงมีการออกแบบให้มีการบันทึก ตาราง State Space นี้เป็นไฟล์สกุล CSV ไปยังแหล่งความจำภายนอก และ จะทำการอ่านไฟล์ CSV ขณะเริ่มต้นกระบวนการการทำงานในครั้งถัดไปเพื่อนำข้อมูลของกระบวนการเรียนรู้มาใช้งานได้อย่างต่อเนื่องโดยที่ไม่จำเป็นจะต้องทำการเรียนรู้ใหม่ทุกครั้งที่เริ่มทำงาน

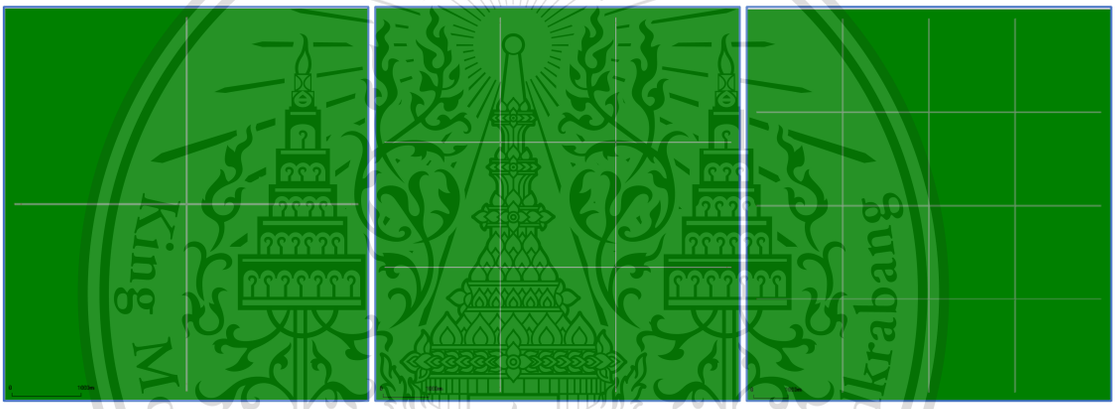
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.3 SUMO (Simulation of Urban MObility)[4]

การออกแบบของการทำงานภายใน SUMO นั้นเปรียบเสมือนการออกแบบ Environment ให้แก่ระบบ ซึ่งโปรแกรม SUMO จะเป็นโปรแกรมที่ใช้ในการจำลองสภาพการจราจรที่กำหนดค่าของสถานะไฟจราจรที่มาจาก Reinforcement Learning ในการออกแบบการทำงานของโปรแกรม SUMO นั้นจะสามารถแบ่งออกเป็น การออกแบบแผนที่ที่ใช้ในการจำลอง การออกแบบสถานะไฟจราจร การออกแบบเส้นทางสัญจรของยานพาหนะ โดยจะมีรายละเอียดดังต่อไปนี้

3.3.1 แผนที่การจราจร

การออกแบบแผนที่ การจราจรเพื่อใช้ในการจำลองสภาพการจราจรของระบบ โดยทำการใช้งานทางด้านโปรแกรม Netedit ในการสร้างแผนที่ขึ้นมา โดยจากกระบวนออกแบบการทดลองนั้นทำให้มีการออกแบบแผนที่ทั้งสิ้น 3 แผนที่ โดยจะมีลักษณะที่แตกต่างกันดังต่อไปนี้



ก) ข) ค)

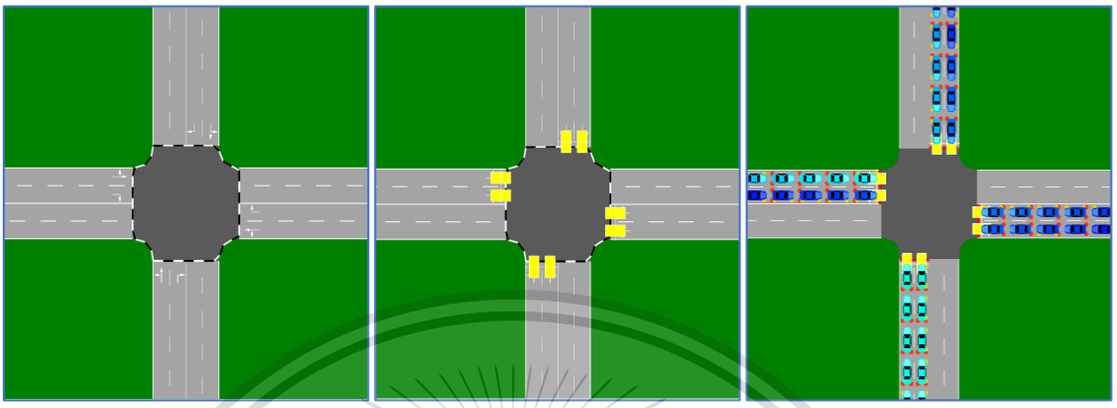
รูปที่ 3.4 การใช้โปรแกรม Netedit ในการออกแบบแผนที่การจราจร

ก) รูปแผนที่การจำลองแบบ 1 x 1
ข) รูปแผนที่การจำลองแบบ 2 x 2
ค) รูปแผนที่การจำลองแบบ 3 x 3

จากรูปที่ 3.4 จะเห็นได้ว่าเป็นการออกแบบให้ครอบคลุมกับสภาพ ของ environment ของเส้นทางต่างๆ ทั้งบริเวณพื้นที่ขนาดเล็ก กลาง จนไปถึงพื้นที่ระบบการจราจรขนาดใหญ่ โดยแผนที่ทั้ง 3 นั้นจะมีลักษณธที่เหมือนกันคือ ถนน 1 เส้นทั้งขาเข้า และ ขาออกจากระบบมีลักษณะ 2 เลน โดยบริเวณรอบนอกของระบบนั้น ที่ปลายของถนนจะมีบริเวณจุดกลับยานพาหนะ โดยที่ยานพาหนะฝั่งขาออกที่กำลังจะออกจากระบบ หากวิ่งทางด้านขวา ก็จะสามารถกลับรถและกลับเข้าสู่ระบบได้อีกครั้ง หรือ เลือกที่จะตรงไปเพื่อทำการออกจากระบบก็สามารถทำได้เช่นเดียวกัน การออกแบบแผนที่นั้นได้ทำการกำหนดให้ความยาวของถนน 1 เส้นมีความยาว 2500 เมตร หรือ 2.5 km นั้นเอง โดยภายในแผนที่จำลองนั้น จำเป็นต้องมีเครื่องมือที่ช่วยในการตรวจจับยานพาหนะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is copyrighted and is for educational use only. Commercial use is prohibited. Forbidden to modify the content, and cite the document when use.

จึงออกแบบให้มีการวางเครื่องมือ Detector ซึ่งเป็นเครื่องมือในการตรวจจับยานพาหนะของโปรแกรม SUMO โดยมีการกำหนดให้วางไว้ที่บริเวณทางเข้า พื้นที่ 4 แยกไฟจราจร และ บริเวณจุดเริ่มต้นของถนนที่ยานพาหนะเข้ามาเพื่อใช้ในการวัดค่าและนำไปหาค่า flowrate และ arrival rate



ก) ข) ค)

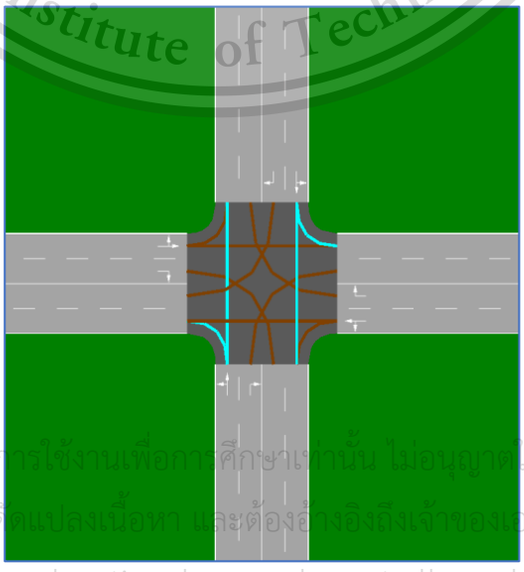
รูปที่ 3.5 การใช้โปรแกรม Netedit ในการออกแบบบริเวณ 4 แยกไฟจราจร

ก) ลักษณะของเลนและการแบ่งเลนของแผนที่

ข) บริเวณจุดวางตัว Detector การจราจร

ค) ลักษณะการ Simulation ของโปรแกรม SUMO

ในการออกแบบแผนที่นั้นก็จะมีส่วนที่ต้องออกแบบการกำหนดเส้นทางการจราจรในบริเวณ 4 แยกไฟจราจรด้วยซึ่งออกแบบให้เลนทางด้านฝั่งซ้ายนั้นยานพาหนะสามารถตรงไปและเลี้ยวซ้ายไป ส่วนเลนทางด้านฝั่งขวาจะเป็นการบังคับให้ยานพาหนะเลี้ยวขวา ซึ่งจะกำหนดทิศทางการไปตามสัญญาณไฟจราจร



รูปที่ 3.6 การออกแบบเส้นทางยานพาหนะบริเวณ 4 แยก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.3.2 การออกแบบเส้นทางของยานพาหนะ

การกำหนดเส้นทางของยานพาหนะในแต่ละคันของโปรแกรม SUMO นั้นจะใช้วิธีการเขียน File XML ขึ้นมาเพื่อทำการกำหนดเส้นทาง โดยสามารถกำหนดความสามารถลักษณะเฉพาะตัวของรถในแต่ละคันได้จากการเขียน File XML เช่นเดียวกัน ในการกำหนดเส้นทางของยานพาหนะในแต่ละคันนั้น จะทำการออกแบบโดยการสุ่มเส้นทางของยานพาหนะ โดยทำการ ใช้ command run python file ที่ชื่อว่า “randomTrips.py” ซึ่งเป็น Python File ที่ทาง SUMO มีมาให้กับการดาวโหลดโปรแกรม SUMO ซึ่งรายละเอียดคำสั่งการสุ่มเส้นทางของยานพาหนะจะมีลักษณะดังนี้

โปรแกรม 3.1 ตัวอย่างคำสั่งการ Run randomTrips.py

```
python randomTrips.py
-n 4-way.net.xml -r Period_1.rou.xml -e 1000 -l --
validate --fringe-factor 10 --period 1 --trip-
attributes="departLane=\"best\" departSpeed=\"random\"
departPos=\"random\" type=\"vehDist\"" --weights-prefix
4-way --additional-file vTypeDistributions.add.xml
```

จากโปรแกรมที่ 3.1 จะสามารถอธิบายการทำงานได้โดยทำการอ้างอิง ไฟล์แผนที่ที่ชื่อว่า “4-way.net.xml” โดยทำการสร้างไฟล์เส้นทางของยานพาหนะที่ชื่อว่า “Period_1.rou.xml” โดยมีระยะเวลาทั้งสิ้น 1000 หน่วยเวลา และมีความคาบของยานพาหนะที่เข้ามาในระบบ 1 คันต่อ 1 หน่วยเวลา นอกจากนั้นก็จะเป็นส่วนในการกำหนดลักษณะเฉพาะตัวของยานพาหนะ

จากการออกแบบการทดลองนั้นมีออกแบบการกำหนดความหนาแน่นของยานพาหนะที่จะเข้ามาในระบบนั้นอยู่ 3 ลักษณะ คือความหนาแน่นมาก ความหนาแน่นกลาง และความหนาแน่นน้อย เพื่อทำการอ้างอิงการใช้งานและวัดประสิทธิภาพของตัว Model Reinforcement Learning ในการรองรับสภาพการจราจรที่หลากหลาย จึงมีการกำหนดการสร้าง File Rou ด้วย “randomTrips.py” ตามที่กล่าวไปข้างต้น ไว้ทั้งหมด 3 file ในแต่ละ Map ที่แตกต่างกัน โดยจะทำการเปลี่ยนแปลงข้อมูลในส่วนของ period ในการรันคำสั่งเพื่อเป็นการกำหนดความหนาแน่นของยานพาหนะที่จะเข้ามาในระบบ โดยความหนาแน่นของยานพาหนะจะสามารถสรุปเป็นตารางได้ดังนี้

ตารางที่ 3.4 ความหนาแน่นของยานพาหนะในแต่ละแผนที่

แผนที่	ความหนาแน่น	Period	คัน/ชั่วโมง
4-Way	มาก	0.8	4500
	กลาง	1.0	3600
	น้อย	1.2	3000

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ตารางที่ 3.4 ความหนาแน่นของยานพาหนะในแต่ละแผนที่(ต่อ)

แผนที่	ความหนาแน่น	Period	คัน/ชั่วโมง
16 -Way	มาก	0.6	6000
	กลาง	0.8	4500
	น้อย	1.0	3600
36-Way	มาก	0.6	6000
	กลาง	0.8	4500
	น้อย	1.0	3600

การสร้าง File Rou โดยการกำหนดความหนาแน่นของยานพาหนะที่เข้ามาในระบบตามตารางที่ 3.4 ด้วย “randomTrips.py” จะใช้คำสั่งต่อไปนี้

โปรแกรม 3.2 คำสั่งการ Run randomTrips.py แผนที่ 4-way ความหนาแน่นปานกลาง

```
python randomTrips.py -n ./4-way/4-way.net.xml
-r ./4-way/route/Period_2.rou.xml -e 10800 -l
--validate --period 1.0
--trip-attributes="departLane=\"random\"
departSpeed=\"random\" departPos=\"random\"
type=\"vehDist\"" --weights-prefix ./4-way/4-way
--additional-file vTypeDistributions.add.xml
```

โปรแกรม 3.3 คำสั่งการ Run randomTrips.py แผนที่ 16-way ความหนาแน่นปานกลาง

```
python randomTrips.py -n ./16-way/16-way.net.xml
-r ./16-way/route/Period_2.rou.xml -e 10800 -l
--validate --period 0.8
--trip-attributes="departLane=\"random\"
departSpeed=\"random\" departPos=\"random\"
type=\"vehDist\"" --weights-prefix ./16-way/16-way
--additional-file vTypeDistributions.add.xml
--min-distance 2500
```

โปรแกรม 3.4 คำสั่งการ Run randomTrips.py แผนที่ 36-way ความหนาแน่นปานกลาง

```
python randomTrips.py -n ./36-way/36-way.net.xml
-r ./36-way/route/Period_2.rou.xml -e 10800 -l
--validate --period 0.8
--trip-attributes="departLane=\"random\"
departSpeed=\"random\" departPos=\"random\"
type=\"vehDist\"" --weights-prefix ./36-way/36-way
--additional-file vTypeDistributions.add.xml
--min-distance 7500
```

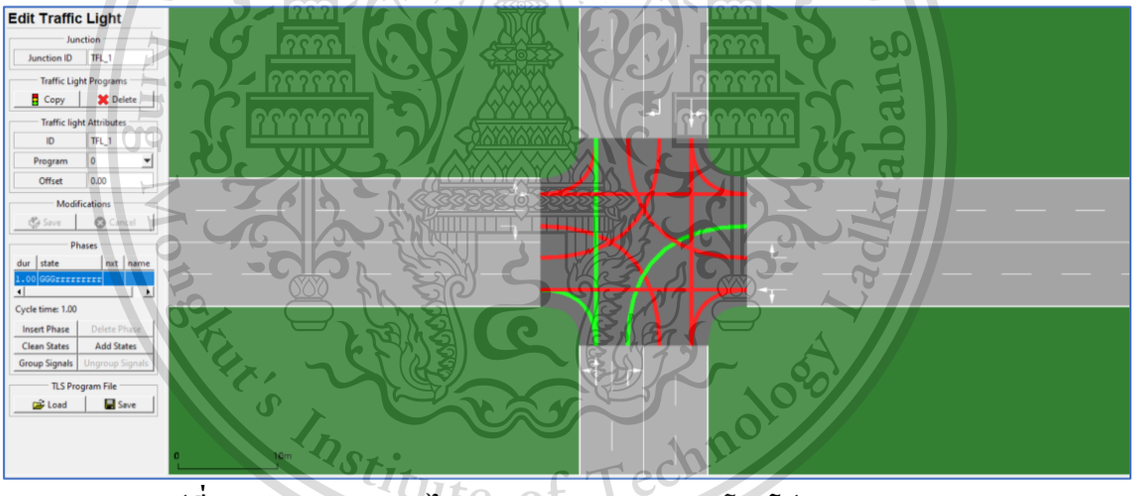
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is restricted for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.3.3 การออกแบบสถานะไฟจราจร

วิธีที่ใช้ในการกำหนดสัญญาณไฟจราจรภายโปรแกรม SUMO นั้นจะใช้วิธีการกำหนดสัญญาณด้วยตัวอักษรแทน สถานะดังต่อไปนี้

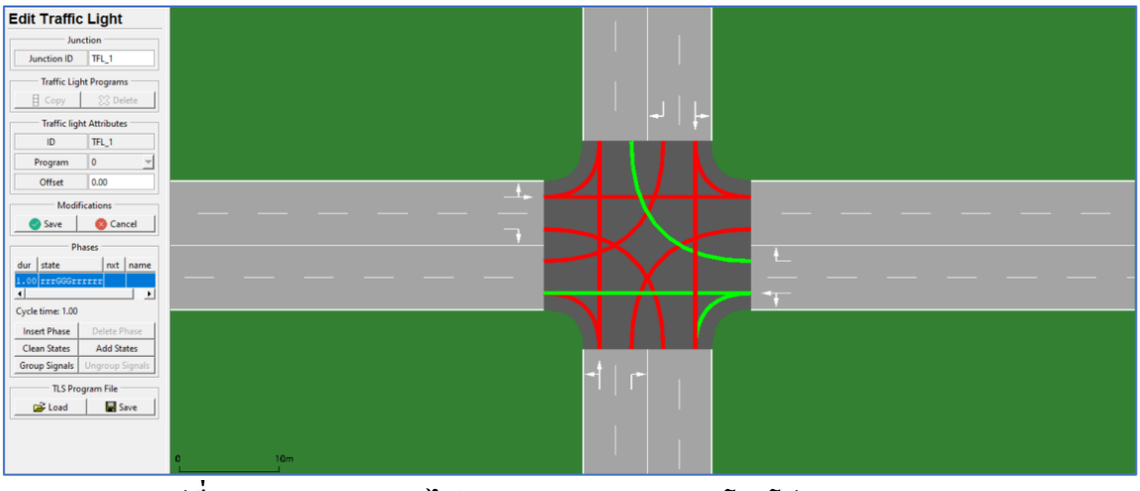
1) G หมายถึง สถานะไฟเขียว
2) g หมายถึง สถานะไฟเขียวโดยต้องหยุดรอยานพาหนะทางหลักไปก่อน
3) y หมายถึง สถานะไฟเหลือง
4) r หมายถึง สถานะไฟแดง

การใช้ตัวอักษรแทนสถานะ จะเป็นการกำหนดค่าให้กับเส้นทางทุกเส้นทางที่เชื่อมต่อกันในบริเวณ 4 แยกไฟจราจร โดยเส้นทางที่ต้องทำการกำหนดนั้นได้แบ่งออกเป็นด้านละ 3 เส้นทาง โดยมีทั้งหมด 4 ด้าน รวมเส้นทางที่ต้องกำหนดในบริเวณ 4 แยก คือ 12 เส้นทาง โดยแต่ละเส้นทางแทนตัวอักษรสถานะ 1 ตัวอักษร ดังนั้นการกำหนดสถานะไฟจราจรของระบบ ใช้ตัวอักษรทั้งหมด 12 ตัวอักษร โดยการกำหนดสัญญาณไฟจราจรนั้นสามารถกำหนดได้โดยผ่านโปรแกรม Netedit และตั้งค่าผ่านคำสั่งใน Library Traci

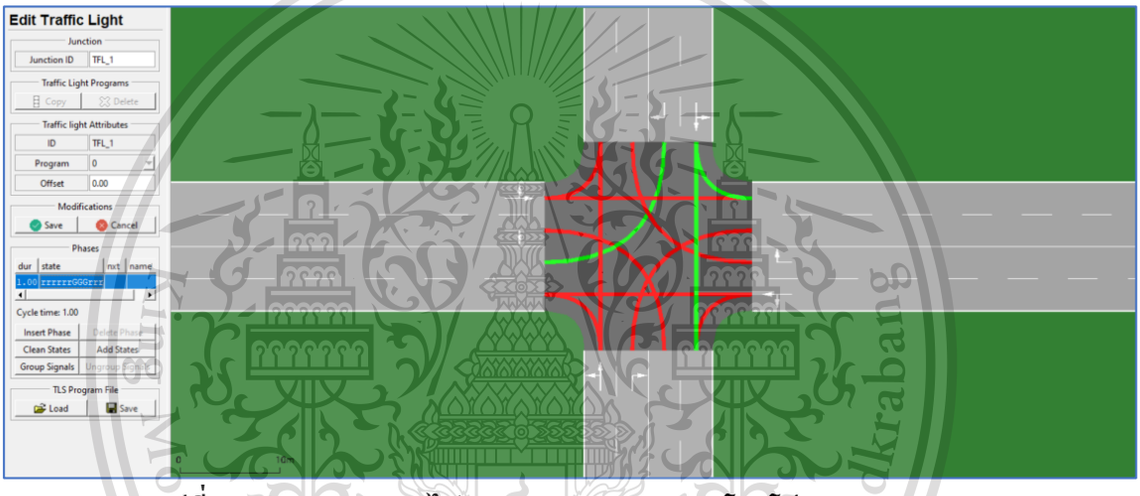


รูปที่ 3.7 กำหนดสถานะไฟจราจร Character 1-3 โดยโปรแกรม Netedit

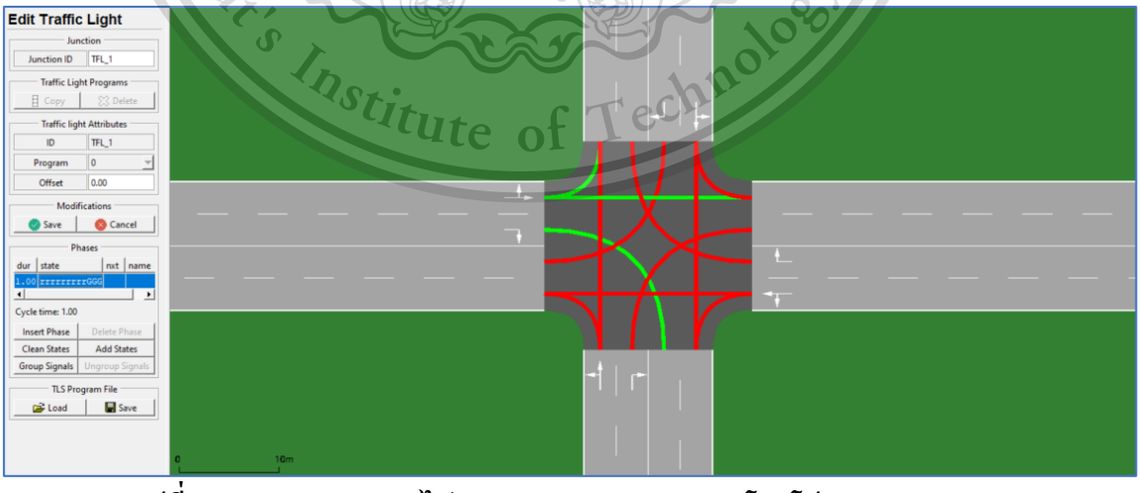
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 3.8 กำหนดสถานะไฟจราจร Character 4-6 โดยโปรแกรม Netedit



รูปที่ 3.9 กำหนดสถานะไฟจราจร Character 7-9 โดยโปรแกรม Netedit



รูปที่ 3.10 กำหนดสถานะไฟจราจร Character 10-12 โดยโปรแกรม Netedit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรม 3.5 ตัวอย่างการกำหนดสัญญาณไฟจราจรผ่าน Traci

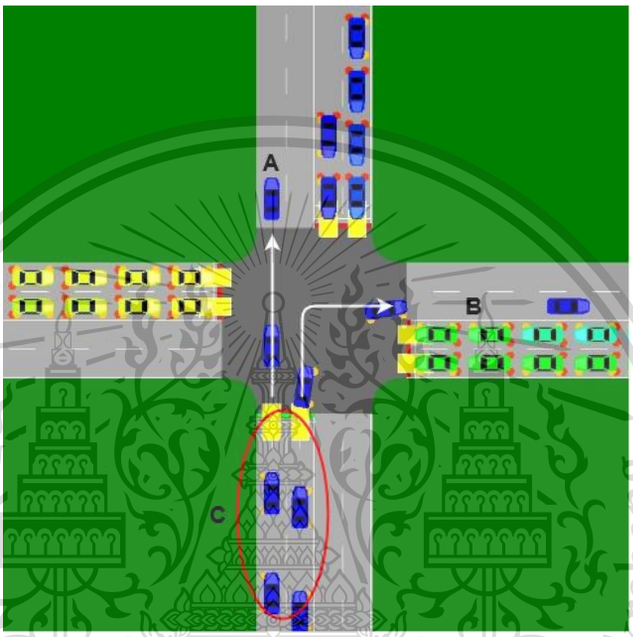
```
import traci
   def set_logic(phase):
       tlPhase = []
       tlPhase.append(traci.trafficlight.Phase(3,
phase[1], 3, 3))
       tlPhase.append(traci.trafficlight.Phase(phase[2],
phase[0], phase[2], phase[2]))
       logic = traci.trafficlight.Logic("0", 0, 0,
tlPhase)
       traci.trafficlight.setProgramLogic(self.id,
logic)

   def set_TrafficLight():
       g_phase = "rrrrrrrrrGGG"
       y_phase = "rrrrrrrrryyy"
       phase = [g_phase, y_phase, 60]
       tls.set_logic(phase)
```

จากโปรแกรมที่ 3.2 สามารถอธิบายได้โดย เริ่มจากเป็นการเพิ่ม Phase ของสัญญาณไฟจราจรเข้าไป Array โดยใช้คำสั่ง traci.trafficlight.Phase ซึ่งค่าของ Parameter ตัวแรกของ คำสั่งนี้คือ ค่า Duration ของสถานะไฟจราจร Parameter ตัวที่ 2 เป็นค่าของ State ของไฟจราจร โดยลักษณะเป็นค่าของตัวอักษร 12 ตัว ตามที่กล่าวไปข้างต้น ในส่วนของ Parameter ที่ 3 และ 4 นั้นเป็นค่าของ Max และ Min Duration เพื่อกำหนดขอบเขตของระยะเวลาในกรณีที่ทำการสุ่มระยะเวลาของสัญญาณไฟจราจร ซึ่งในการทำงานของสัญญาณไฟจราจรของวิทยานิพนธ์เล่มนี้ไม่มีงานทำงานในส่วนของการสุ่มค่าระยะเวลาจึงทำแทนค่าด้วย Duration ของสัญญาณไฟจราจร แทนที่การกำหนดขอบเขตของสัญญาณไฟจราจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

การกำหนดสถานะของไฟจราจรบริเวณ 4 แยกไฟจราจรนั้น ในงานวิจัย TSCRL ที่ใช้ในการอ้างอิงจะใช้วิธีในการกำหนดสถานะของไฟจราจรเป็นแบบ Fix Time โดยมีการกำหนด Parameters ของระยะเวลาไว้ที่ 60 วินาทีต่อรอบ ในขณะที่ Model Reinforcement Learning ที่ได้ทำการออกแบบเองนั้นจะใช้วิธีถ่วงน้ำหนักตามความหนาแน่นของบริเวณ 4 แยกไฟจราจรโดยใช้ความหนาแน่นของถนนที่ยานพาหนะกำลังไป และ ความหนาแน่นของถนนของ ณ ปัจจุบัน โดยทำการคำนวนเพื่อถ่วงน้ำหนัก Green time ที่เหมาะสม



รูปที่ 3.11 เส้นทางในการคำนวณ Green Time

จากรูปที่ 3.7 สามารถอธิบายได้ว่า ณ State ปัจจุบันอยู่ที่จุด C โดยเลนที่ยานพาหนะจากจุด C จะเคลื่อนที่ไปคือจุด A และ จุด B การคำนวน Green Time จะคำนวณจาก

1) ความหนาแน่นของเส้นทางที่จะไป ซึ่งก็คือ จุด A และ จุด B
2) ความหนาแน่นของเส้นทางปัจจุบัน ซึ่งก็คือ จุด C

สามารถสร้างสมการเพื่อใช้ในการคำนวณได้ดังนี้

$$\frac{\text{ความหนาแน่นจุด } A + \text{ความหนาแน่นจุด } B}{2} \tag{3.11}$$

โดยความสัมพันธ์ของ Green time และ ความหนาแน่นของยานพาหนะในแต่ละเส้นทางจะเป็นดังนี้

$$Green\ Time \propto \frac{\text{สมการที่ (3.11)}}{\text{ความหนาแน่นจุด } C} \tag{3.12}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เมื่อนำความสัมพันธ์ระหว่าง Green Time และ ความหนาแน่นของยานพาหนะในเลน จากสมการความสัมพันธ์ที่ 3.12 นำมาถ่วงน้ำหนักกับค่าระยะเวลา Green Time ที่กำหนดให้ค่าพื้นฐานเป็นระยะเวลา 60 วินาที แล้วจะได้สมการคำนวณ Green Time ดังต่อไปนี้

$$Green\ Time = \frac{\text{สมการที่ } (3.11)}{\text{ความหนาแน่นจุด } C} \times 60 \qquad (3.13)$$

นอกจากการถ่วงน้ำหนัก Green Time แล้วจะมีการออกแบบโดยมีการวัดค่า Waiting Time ที่มากที่สุดในแต่ละ State โดย State ใดที่มี Waiting Time เกินระยะเวลา 300 วินาที หรือ ประมาณ 5 นาที จะทำการกำหนดให้เลน หรือ State นั้น ๆ เป็น State ต่อไปเพื่อที่จะให้ยานพาหนะภายในเลน ได้รับสิทธิในการเปิดสัญญาณไฟเขียวทันที

3.4 Simulation API

เป็นส่วนที่เกี่ยวกับการใช้งาน Library Traci เพื่อเป็น Module ตัวกลางในการติดต่อสื่อสารระหว่างส่วนต่างๆ ทั้งส่วน Reinforcement Learning และส่วนแสดง Visualization กับตัวโปรแกรมจำลอง SUMO Simulation โดยในส่วนของ Simulation API จะมีการใช้ Traci ในการกำหนดสัญญาณไฟจราจรภายในระบบของโปรแกรม SUMO อีกส่วนหลักสำคัญของ Module Simulation API นั้นก็คือ การติดต่อสื่อสารกับโปรแกรม SUMO เพื่อทำการรับค่าผลลัพธ์ที่ได้จากการจำลอง เช่นค่าของ ความหนาแน่น ระยะเวลารอของยานพาหนะ ปริมาณของยานพาหนะที่เคลื่อนที่ผ่านบริเวณ 4 แยกจราจร เป็นต้น เพื่อทำการส่งค่าผลลัพธ์เหล่านี้ไปให้ส่วนของ Module Reinforcement Learning เพื่อนำไปใช้ในการคำนวณ Reward และ คำนวณ Green Time อีกทั้งยังนำข้อมูลเหล่านั้นไปใช้ในส่วนของการแสดง Visualization เพื่อทำการ Output ผลลัพธ์ออกมาแสดงเป็นลักษณะของกราฟ เพื่อความเข้าใจง่ายในการสื่อสาร

โปรแกรม 3.6 ตัวอย่างการเรียกใช้งาน Traci API

```
import traci

lane = 'genE3'
vehicle_ID = 'Car1'
TrafficLightPhases = []

#getWaitingtime
waiting_time = traci.lane.getWaitingTime(lane)

#getSpeed
speed = traci.vehicle.getSpeed(vehicle_ID)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

จากโปรแกรมที่ 3.1 ซึ่งเป็นโปรแกรมแสดงตัวอย่างวิธีการเรียกใช้งาน Library Traci ของโปรแกรม SUMO Simulation เพื่อนำมาติดต่อสื่อสารกับส่วนต่างๆของระบบตามที่กล่าวไปข้างต้น โดยตัวอย่างการใช้งาน จะเป็นการเรียกใช้งาน Library Traci เพื่อทำการดึงค่า ระยะเวลารอของยานพาหนะ หรือ Waiting Time โดยคำสั่งนี้จำเป็นต้องมีการใส่ค่าของ Parameters เป็นค่าของ Lane ที่จะทำการหา Waiting Time ในบรรทัดถัดมาจะเป็นการแสดงตัวอย่างในการเรียกใช้งานเพื่อทำการดึงข้อมูลความเร็วของยานพาหนะภายในระบบ โดยมีการใส่ค่าของ Parameters เป็นเลข ID ของยานพาหนะที่ต้องการในดึงข้อมูลนำออกมาใช้งาน

3.4.1 Waiting Time

Waiting Time หรือ ระยะเวลารอของยานพาหนะในระบบ โดยออกแบบให้การนับเวลารอของยานพาหนะนั้น จะเริ่มนับตั้งแต่วินาทีที่ยานพาหนะแต่ละคันนั้นได้เข้ามาสู่ในระบบ และ จะนับเวลาจนถึงยานพาหนะคันนั้นๆออกจากระบบไป โดยจะนับเวลา Waiting Time ขณะที่ความเร็วของยานพาหนะเป็น 0 หรือ ขณะที่ยานพาหนะหยุดเคลื่อนที่ โดยการเก็บ Waiting Time จะเริ่มจากการหา ID ทั้งหมดของยานพาหนะในแต่ละเลน และนำเลข ID เหล่านั้นไปหาค่าของ Accumulated Waiting Time ด้วย Library Traci ซึ่งเป็นการเก็บระยะเวลารอของยานพาหนะโดยเฉลี่ยแยกกันในแต่ละเลนของบริเวณ 4 แยกไฟจราจร เพื่อที่จะทำการส่งออกไปยังส่วนของ Module Reinforcement Learning และนำไปทำการคำนวณ Reward และ แสดงผลด้วย Visualization ต่อไป

3.4.2 Density

ข้อมูลที่ระบุถึงปริมาณความหนาแน่นของยานพาหนะบนท้องถนน โดยเทียบกับความยาวของถนนแต่ละเส้นในระบบ ซึ่งทำการออกแบบวิธีเก็บข้อมูลของ Density โดยเริ่มจากทำการเก็บเลข ID ของยานพาหนะในส่วนของขาเข้าทุกเลน ณ บริเวณ 4 แยกไฟจราจรเสียก่อน แล้วจึงนำเลข ID เหล่านั้นไปหาจำนวนของยานพาหนะขาเข้าทุกคันในระบบนั้นๆ ด้วย Library Traci เมื่อได้จำนวนของยานพาหนะทุกคันมาแล้วจึงนำจำนวนของยานพาหนะนำไปหารด้วย 4 ซึ่งจะกลายเป็นค่าเฉลี่ยความหนาแน่นในแต่ละเลนและนำไปหารด้วยความยาวของถนนอีกที และสุดท้ายทำการแปลงหน่วยให้เป็นต่อกิโลเมตร โดยจะได้สมการดังต่อนี้ไป

$$Density = \frac{\left(\frac{Amount}{4}\right) \times 1000}{lane\ length} \tag{3.14}$$

เมื่อได้ค่าของ Density จากสมการที่ 3.14 แล้วก็จะทำการบันทึกค่าและทำการออกไปยังส่วนของการทำ Visualization เพื่อจะนำค่า Density ที่ได้ไปทำการแสดงผลวัดประสิทธิภาพด้วยกราฟต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ ดัดแปลง เผยแพร่ หรือแสวงหาประโยชน์ทางการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

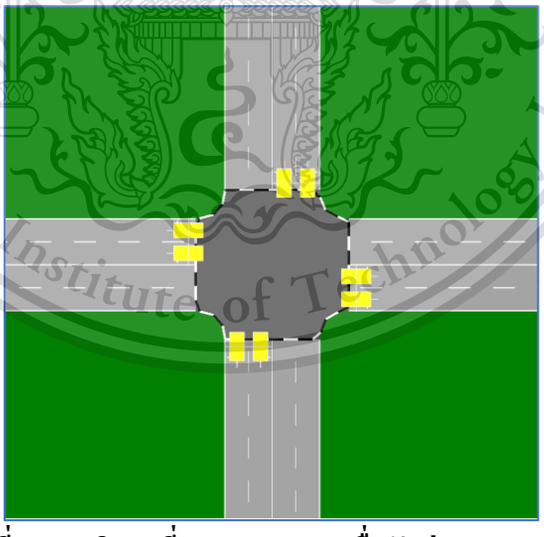
3.4.3 Speed

ข้อมูลความเร็วของยานพาหนะนั้น จะทำการออกแบบโดยการเก็บเป็นค่าเฉลี่ยของยานพาหนะทุกคันในระบบ ซึ่งจะใช้วิธีเดียวกันกับการเก็บค่าของ Density คือการเริ่มจากการเก็บค่าของ ID ของยานพาหนะในส่วนของขาเข้าทุกเลนในระบบ และ นำเลข ID มาหาค่า Average Speed โดยใช้ Traci ในการดึงข้อมูลออกมาเก็บไว้ใน Array โดยหน่วยที่ได้จากโปรแกรม SUMO นั้นจะมีหน่วยเป็น เมตร ต่อ วินาที (m/s) จึงจำเป็นจะต้องทำการแปลงหน่วยให้เป็น กิโลเมตร ต่อ ชั่วโมง ซึ่งจะแสดงการหาความเร็วเฉลี่ยได้ดังสมการต่อไปนี้

$$Avg\ Speed = \left(\frac{avg_speed}{4}\right) \times 3.6 \tag{3.15}$$

3.4.4 Flow Rate

ข้อมูลของปริมาณของยานพาหนะที่สามารถเคลื่อนที่ผ่านจุด ๆ หนึ่งได้ใน 1 หน่วยเวลาโดยให้ Flow Rate เป็นการวัดประสิทธิภาพการเคลื่อนที่ของยานพาหนะที่เคลื่อนตัวออกจากระบบหรือ ผ่านบริเวณ 4 แยกไฟจราจรแล้ว โดยใช้เครื่องมือ Detector ในการวัดค่าของปริมาณ ซึ่งทำการวางตัววัด Detector เอาไว้ที่บริเวณด้านต้นของเลนในส่วนของที่ติดกับถนนกลาง 4 แยกไฟจราจรตามนิยามของ Flow Rate ตามภาพต่อไปนี้



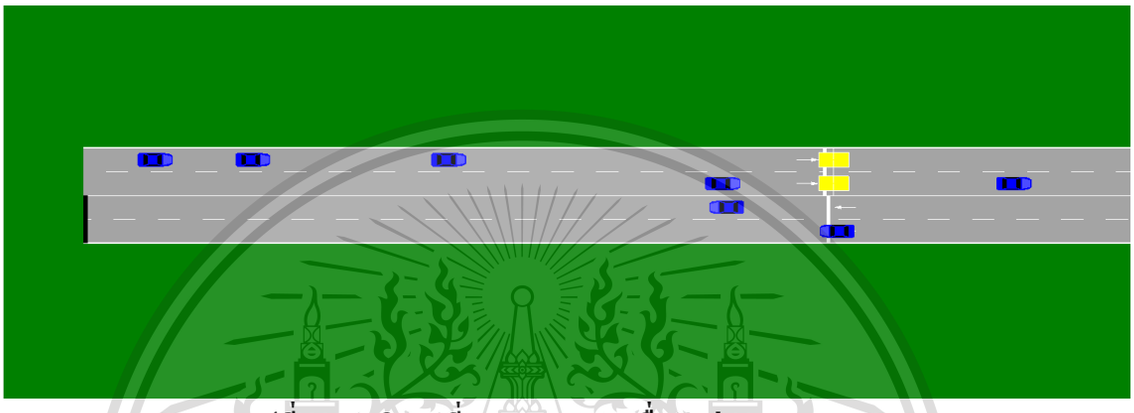
รูปที่ 3.12 บริเวณที่วาง Detector เพื่อวัดค่า Flow Rate

โดยเครื่องมือ Detector นั้นจะทำการตรวจจับยานพาหนะที่เคลื่อนที่ผ่านตัว Detector เพื่อป้องกันปัญหาในกรณีที่ยานพาหนะมีการหยุดเคลื่อนที่อยู่บริเวณ Detector พอดี จึงมีการเช็คในส่วนของเลข ID ของยานพาหนะคันก่อนหน้า หากมีหมายเลข ID ที่แตกต่างกันจึงจะทำการนับยานพาหนะที่ผ่านตามเงื่อนไข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารฉบับนี้ทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.4.5 Arrival rate

จะเป็นการใช้นิยามเดียวกันกับ Flow Rate โดยวัดปริมาณของยานพาหนะที่สามารถเคลื่อนที่ผ่านจุด ๆ หนึ่งได้ใน 1 หน่วยเวลา โดยข้อแตกต่างระหว่าง Flow Rate และ Arrival Rate คือ Arrival Rate นั้นจะทำการวัดปริมาณของยานพาหนะที่เข้ามาสู่ระบบ โดยจะใช้วิธีการวาง Detector เหมือนกันกับ Flow Rate แต่จะทำการวาง Detector เอาไว้ที่ปลายของถนนของทั้ง 4 ด้านฝั่งขาเข้าบริเวณที่ยานพาหนะเข้ามาในระบบตามรูปตัวอย่างต่อไปนี้



รูปที่ 3.13 บริเวณที่วาง Detector เพื่อวัดค่า Arrival rate

ในการเก็บค่าของ Flow Rate และ Arrival Rate ผ่านเครื่องมือ Detector จะได้ค่าผลลัพธ์ ที่ออกมามีหน่วยเป็น คันต่อวินาที ดังนั้นจึงจำเป็นจะต้องนำผลลัพธ์ที่ได้ เอามาทำการเปลี่ยนหน่วยให้เป็นคัน/ชั่วโมง โดยทำการคูณด้วย 3600 และ หารด้วย เวลา ตามสมการดังต่อไปนี้

$$\text{Flow Rate} = \left(\frac{flow_rate}{time}\right) \times 3600 \tag{3.16}$$

$$\text{Arrival Rate} = \left(\frac{arrival_rate}{time}\right) \times 3600 \tag{3.17}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.5 CSV And Plotter

ในส่วนของ Module ที่เป็นขั้นตอนการแสดงการวัดผล โดยเป็นส่วนที่นำข้อมูลผลลัพธ์ที่ได้จากโปรแกรม SUMO Simulation และทำการบันทึกค่าลงไฟล์ CSV และ ให้ Module Plotter ทำการอ่านไฟล์ CSV ที่บันทึกและแสดงผล Visualization ในลักษณะของกราฟทั้งแบบแผนภูมิเส้น และ แบบแผนภูมิแท่ง ด้วย Matplotlib โดย Output ที่ออกมาจะมี 2 ส่วนเพื่อใช้ประโยชน์ตามหน้าที่ ที่แตกต่างกันออกไป ดังต่อไปนี้

3.5.1 กราฟมอนิเตอร์ค่าเฉลี่ยของผลลัพธ์ของระบบแบบ Real Time

โดยขณะที่ระบบกำลังทำงาน เพื่อเรียนรู้ข้อมูลต่างๆ ตาม Algorithm ของ Model Reinforcement Learning อยู่นั้น จึงต้องมีการออกแบบ การ Output ข้อมูลระหว่างการทำงาน โดยเป็นลักษณะของการ Monitoring Data จะเป็นแบบ Real Time ซึ่งการทำการจะเริ่มต้นด้วย เมื่อมีข้อมูลที่ส่งออกมาจากโปรแกรม SUMO ด้วย Traci ที่ทำการเขียนไว้ จะทำการบันทึกข้อมูลเหล่านั้นไว้เป็นไฟล์ CSV และ Run File Python ใน Terminal แยกเปิดหน้าต่างของการ Plot กราฟขึ้นมา โดยจะทำการอ่านข้อมูลของไฟล์ CSV ที่ทำการบันทึกโดยทันที เพื่อใช้ในการดูแนวโน้มของผลลัพธ์ที่ได้ออกมา นำมาเปรียบเทียบกับสภาพการจราจรที่แสดงผลออกมาผ่านทาง GUI ขอ
โปรแกรม SUMO Simulation โดยการ Monitoring ชุดข้อมูลนั้นจะแบ่งเป็นหัวข้อดังนี้

3.5.1.1 กราฟผลลัพธ์ที่ได้จากการ Simulation

ข้อมูลผลลัพธ์ที่จะนำออกมาวัดผล ซึ่งได้จากการ Simulation ของโปรแกรม SUMO จะเป็นกราฟข้อมูลโดยที่ตัวแปรในแต่ละตัวจะแทนด้วยแนวแกน Y และ นำมาเปรียบเทียบด้วยแกนข้อมูลของเวลา โดยทำการ Monitoring แบบ Real Time เพื่อที่จะดูแนวโน้มของข้อมูล และ การเปลี่ยนแปลงของ Environment ของระบบ ณ ขณะนั้น ๆ เพื่อความเข้าใจง่ายต่อการสังเกต โดยข้อมูลของกราฟผลลัพธ์ที่แสดงออกมานั้นจะมีข้อมูลดังนี้

1) กราฟเส้นวัดผลค่าเฉลี่ยของ Flow Rate
2) กราฟเส้นวัดผลค่าเฉลี่ยของ Speed
3) กราฟเส้นวัดผลค่าเฉลี่ยของ Density
4) กราฟเส้นวัดผลค่าเฉลี่ยของ Waiting Time
5) กราฟเส้นวัดผลค่าเฉลี่ยของ Arrival Rate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.5.1.2 กราฟค่าเฉลี่ยของ Q-Value

การทำงานของ Reinforcement Learning นั้นจะสามารถนำค่าของ Q-Value นำมาทำการวัดค่าได้ โดยจะบ่งบอกถึงการทำงานและการเรียนรู้ที่ผ่านมาของ Algorithm ในแต่ละครั้งที่ทำการเรียนรู้โดยจะบันทึกเก็บไว้ในตาราง Q-Table หรือ State Space โดยนำค่าเฉลี่ยของค่า Q-Value นำมา Plot Graph โดยกราฟที่แสดงออกมานั้นจะสามารถอธิบายการทำงานโดยการเรียนรู้ในช่วงเริ่มแรกนั้นลักษณะกราฟค่าของ Q-Value จะมีการเหวี่ยงของค่าเฉลี่ยที่ค่อนข้างสูง เนื่องจากค่าใน State Space นั้นยังไม่คงที่และยังไม่มีการปรับค่า Q-Value ที่เพียงพอทำให้ผลลัพธ์ที่ได้จากการตัดสินใจในการเลือก Action ของ Model นั้นยังมีประสิทธิภาพที่ไม่สูงมากนัก และ เมื่อทำการ Training Reinforcement ในระยะเวลาที่มากขึ้น จะทำให้ค่าของ Q-Value ใน State Space มีค่าที่เข้าสู่จุดสมดุล ณ ค่าๆหนึ่ง ซึ่งหมายความว่าค่า Q-Value นั้นมีการเรียนรู้ที่มากเพียงพอและคงที่ทำให้ความสามารถในการตัดสินใจเลือก Action มีความน่าเชื่อถือและประสิทธิภาพที่ใช้งานได้ โดยหาก Environment ของระบบนั้นมีการเปลี่ยนแปลงในระหว่างการเรียนรู้ เช่นมีปริมาณของยานพาหนะที่เข้ามามากกว่าปกติ ค่าของ Q-value ในตาราง State Space นั้นก็จะเริ่มมีการเปลี่ยนแปลงอีกครั้ง และจะต้องใช้เวลาในการเริ่ม Training สำหรับ Environment ที่เปลี่ยนแปลงไปจากเดิมเพิ่มทำการปรับเปลี่ยนการตัดสินใจให้เหมาะสมกับ Environment นั้นๆ

3.5.1.3 กราฟแสดงความสัมพันธ์ของตัวแปรตาม ทฤษฎี Traffic Engineering[3]

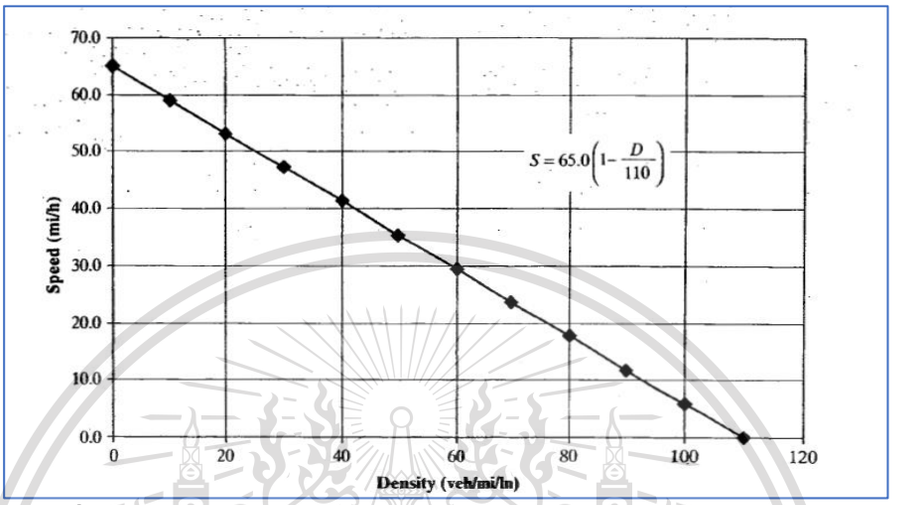
จากข้อมูลในบทที่ 2 เกี่ยวกับทฤษฎีของ Traffic Engineering จะมีการกล่าวถึงการแสดงความสัมพันธ์กันระหว่างตัวแปรในการวัดผลทาง Traffic Engineering ทั้ง 3 ตัวแปร ซึ่งก็คือ Speed, Flow Rate และ Density โดยสามารถอธิบายความสัมพันธ์ได้ผ่านจากการ Plot กราฟ โดยจะสามารถแสดงออกมาผ่านทางกราฟได้ทั้งหมด 3 กราฟ ดังนั้นเพื่อทำการวัดผลข้อมูลที่ได้จากการ Simulation ด้วยโปรแกรม SUMO ว่าตัวแปรทั้ง 3 ชนิดนี้มีลักษณะความสัมพันธ์ที่เป็นไปตาม ทฤษฎีของ Traffic Engineering หรือไม่ จึงมีการออกแบบให้แสดงผลของข้อมูลโดยยึดตามกราฟความสัมพันธ์ โดยกราฟที่ทำการแสดงออกมามีดังต่อไปนี้

3.5.1.3.1 กราฟแสดงความสัมพันธ์ระหว่าง Speed และ Density

กราฟแสดงความสัมพันธ์กราฟแรกจะเป็นกราฟที่แสดงความสัมพันธ์ระหว่างค่าของ Speed และ Density โดยลักษณะของกราฟที่เป็นไปตามทฤษฎี จะเป็นกราฟประเภท Linear โดยกราฟนี้จะบ่งบอกถึงความสามารถในการรองรับยานพาหนะที่เข้ามาในระบบ โดยกำหนดให้ค่าในแนวแกน Y เป็นค่าของ Speed และค่าในแนวแกน X แทนด้วยค่าของ Density โดยการตีความกราฟนี้จะเริ่มจากจุดที่ Density หรือ ความหนาแน่นของยานพาหนะที่เข้ามาในระบบนั้นมีค่าที่น้อย ซึ่งจะส่งผลให้ยานพาหนะนั้นสามารถเคลื่อนที่ด้วยความเร็วที่มากได้ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

เมื่อความหนาแน่นของยานพาหนะในระบบเริ่มเพิ่มมากขึ้นเรื่อย ๆ ก็จะส่งผลให้ยานพาหนะนั้นเคลื่อนที่ด้วยความเร็วที่ลดลงจนในที่สุดเมื่อค่าความเร็วของยานพาหนะมีค่าเข้าใกล้ 0 จะแสดงได้ว่าค่า Density ณ ขณะที่ความเร็วของยานพาหนะเข้าใกล้ 0 จะเป็นค่า Density ที่มากที่สุดที่ระบบของเราจะสามารถรองรับได้

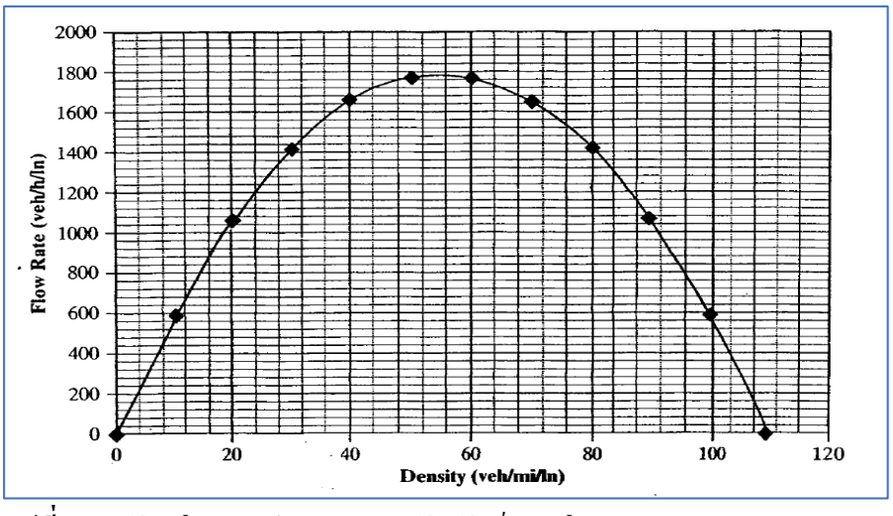


รูปที่ 3.14 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Speed และ Density

3.5.1.3.2 กราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Density

กราฟแสดงความสัมพันธ์ต่อมาที่จะทำการเปรียบเทียบคือ กราฟแสดงความสัมพันธ์ระหว่างค่า Flow Rate ของยานพาหนะในระบบกับค่าของ Density โดยลักษณะของกราฟที่เป็นไปตามทฤษฎี กราฟจะออกมาในรูปแบบของ พาราโบลา ซึ่งจะมีการกำหนดให้ค่าของ Flow Rate จะเป็นค่าในแนวแกน Y และ ค่าของ Density จะแทนค่าในแนวแกน X โดยกราฟความสัมพันธ์นี้จะสามารถตีความหมายได้จากจุด Peak ของกราฟพาราโบลา ที่บอกถึงจุดที่ปริมาณของ Density ที่จะส่งผลให้ค่าของ Flow Rate มีค่าสูงที่สุด หรือ ในอีกความหมายนึงก็คือ จุดที่ความหนาแน่นที่จะทำให้ยานพาหนะสามารถเคลื่อนที่ได้อย่างมีประสิทธิภาพสูงที่สุด เมื่อทำการเพิ่ม Density ที่มากเกินกว่าจุด Peak จะส่งผลให้มีความหนาแน่นที่มากเกินไป ยานพาหนะจะเริ่มคับคั่งและ Flow Rate ของระบบก็จะลดลง

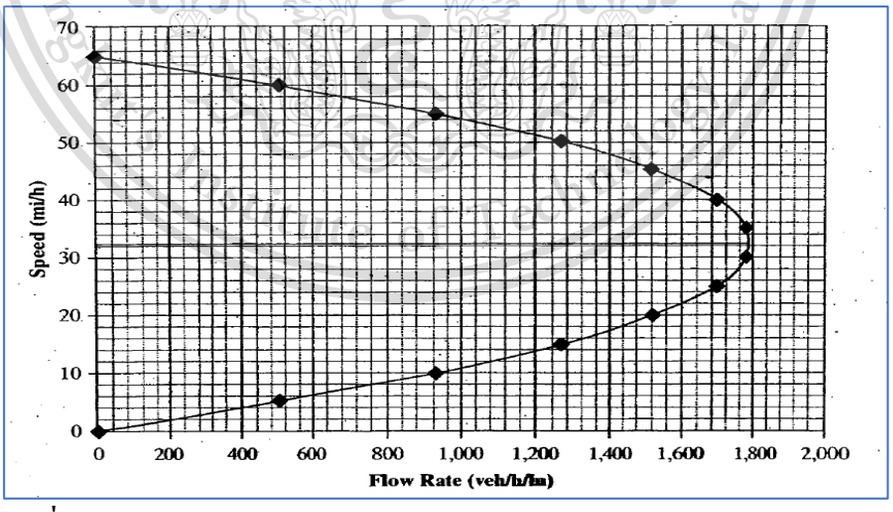
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.



รูปที่ 3.15 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Density

3.5.1.3.3 กราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Speed

กราฟแสดงความสัมพันธ์สุดท้าย คือกราฟแสดงความสัมพันธ์ระหว่างค่าของ Flow Rate และ Speed ของยานพาหนะในระบบ โดยกำหนดให้ค่าในแนวแกน Y แทนด้วยค่าของ Speed และ ค่าในแนวแกน X แทนด้วยค่าของ Flow Rate โดยถ้าหากกราฟนั้นมีความสัมพันธ์ที่เป็นไปตามทฤษฎีของ Traffic Engineering กราฟจะมีลักษณะเป็นรูปแบบพาราโบลาโดยจุด Peak ของกราฟความสัมพันธ์นี้ได้บ่งบอกถึงจุดที่ความเร็วที่สูงที่สุดของยานพาหนะ ที่จะส่งผลให้ Flow Rateในระบบมีค่ามากที่สุด



รูปที่ 3.16 ตัวอย่างกราฟแสดงความสัมพันธ์ระหว่าง Flow Rate และ Speed

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

3.5.2 กราฟเปรียบเทียบผลลัพธ์ในการทดลองแต่ละวิธี

ในขั้นตอนการออกแบบ กระบวนการที่จะใช้ในการทดลองนั้น จะมีการจำแนกการทดลองด้วยวิธีการที่ใช้ในการควบคุมสัญญาณไฟจราจรของระบบ โดยวิธีที่ใช้ในการควบคุมระบบมีทั้งหมด 3 วิธีดังต่อไปนี้

1) การทดของแบบ Fix ระยะเวลาไฟจราจร
2) การทดลองอ้างอิง Model Reinforcement Learning ตามงานวิจัย “Traffic signal control for smart cities using Reinforcement learning.”
3) การทดลอง Model Reinforcement Learning ที่ออกแบบขึ้นเอง

โดยในแต่ละวิธีที่ใช้ในการควบคุมระบบจะถูกทดสอบด้วยชุดข้อมูลความหนาแน่นของยานพาหนะที่แตกต่างกัน 3 ระดับคือ

1) ชุดข้อมูลความหนาแน่นของยานพาหนะ มาก
2) ชุดข้อมูลความหนาแน่นของยานพาหนะ ปานกลาง
3) ชุดข้อมูลความหนาแน่นของยานพาหนะ น้อย

เมื่อผ่านกระบวนการในการทดลองทั้ง 3 วิธีตามที่ออกแบบเสร็จสิ้นแล้ว จึงจะต้องมีส่วนในการแสดงผลลัพธ์ที่ใช้ในการเปรียบเทียบและข้อแตกต่างของการทดลองในแต่ละวิธี และ แต่ละชุดข้อมูล เพื่อเป็นการวัดผลสัมฤทธิ์ของงานวิจัยนี้ ซึ่งจะมีหัวข้อในการวัดผลสัมฤทธิ์ดังต่อไปนี้

1) กราฟเส้นเปรียบเทียบข้อมูล Waiting Time
2) กราฟเส้นเปรียบเทียบข้อมูล Queue Length
3) กราฟเส้นเปรียบเทียบข้อมูล Green Time
4) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Waiting Time
5) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Queue Length
6) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Green Time
7) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Flow Rate
8) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Density
9) กราฟแท่งเปรียบเทียบข้อมูลค่าเฉลี่ย Speed

ในส่วนของการทำงานของโปรแกรม Code และ ผลลัพธ์ของการทดลองของวิทยานิพนธ์การเรียนรู้ของเครื่องแบบเสริมกำลังเพื่อควบคุมสัญญาณไฟจราจร
สามารถศึกษาเพิ่มเติมได้ที่ : https://github.com/Hin-Nattapat/Reinforcement_Traffic_Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองระบบไฟจราจรแบบ Fix เวลา

การทดลองแรกเป็นการทดลองเพื่อวัดประสิทธิภาพของการควบคุมสัญญาณไฟจราจรแบบพื้นฐาน เพื่อทำการอ้างอิงถึงวิธีการควบคุมสัญญาณไฟจราจรตามท้องถนนในประเทศไทย ณ ปัจจุบัน โดยจะทำการเปรียบเทียบกับ Model Reinforcement Learning อีกทั้ง 2 วิธีเพื่อใช้ในการวัดผลการควบคุมว่า การควบคุมไฟจราจรแบบปกติที่ใช้งานกันในปัจจุบันกับวิธีควบคุมแบบการใช้ Reinforcement Learning แบบไหนที่ได้ประสิทธิภาพที่ดีกว่ากัน

4.1.1 รายละเอียดการทดลอง

ทำการทดลองโดย เริ่มต้นด้วยการกำหนดค่าของสัญญาณไฟจราจรของระบบ ทุกแยกโดยกำหนดให้ระยะเวลาของสัญญาณไฟเขียวอยู่ที่ 60 วินาทีทุกแยก โดยจะทำการ Simulation ตามไฟล์ Rou หรือ ไฟล์กำหนดเส้นทางของยานพาหนะ ไปจนสิ้นสุดการทำงานโดย ในการกำหนดไฟล์ Rou นั้นก็จะมีขนาดของความหนาแน่นที่แตกต่างกันทั้งหมด 3 ความหนาแน่น คือ มาก กลาง และ น้อย ในแต่ละ แผนที่ที่ใช้ในการ Simulation ทำการสังเกตผลลัพธ์ และ ทำการบันทึกผลการทดลอง

4.1.2 วิธีการทดลอง

1) ตั้งค่าการทำงานภายในไฟล์ Runner โดยทำการเลือกแผนที่แบบ 4-way เพื่อทำการทดลองกับแผนที่ขนาดเล็ก
2) ทำการเลือก Function ในการทำงาน ด้วยคำสั่งการ Simulation การทดลองแบบ Fix เวลา
3) ทำการใส่ค่าของ Parameters ในการกำหนดระยะเวลาของสัญญาณไฟจราจรโดยใส่ค่า Parameters เป็น 60 วินาที
4) กำหนด Path ของไฟล์เส้นทางยานพาหนะ ในส่วนของการ Config Environment ของระบบ โดยทำการเริ่มด้วยไฟล์ความหนาแน่นน้อย
5) ทำการ Run ไฟล์ Runner.py เพื่อเริ่มต้นการ Simulation
6) ทำการ Run ไฟล์ Plotter.py เพื่อทำการเปิดหน้าต่างของโปรแกรมแสดงผล
7) ติดตามผลการทดลองจาก หน้าต่าง Plotter และ ทำการบันทึกผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

8) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 3 ถึง การทดลองที่ 6 โดยทำการเปลี่ยน Path ความหนาแน่นเป็น ความหนาแน่นกลาง และ ความหนาแน่นมาก และทำการบันทึกผลการทดลอง
9) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 1 ถึง 8 โดยทำการเลือกแผนที่ ที่ใช้ในการทำงานแบบ 16-way และ 36-way เพื่อทำการ Simulation ในแผนที่ขนาดกลาง และ แผนที่ขนาดใหญ่

4.2 การทดลองระบบไฟจราจรด้วย Reinforcement Learning อ้างอิงงานวิจัย TSCRL

การทดลองต่อมาเป็นการทดลองอ้างอิงโดยใช้วิธีการควบคุมสัญญาณไฟจราจรด้วย Reinforcement Learning ซึ่งมีการออกแบบตามในทฤษฎีภายในงานวิจัย “Traffic signal control for smart cities using Reinforcement learning.” ซึ่งเป็นแหล่งข้อมูลหลักในการทำวิจัยเล่มนี้ โดยจะนำมาเป็นการทดลองอ้างอิงในส่วนของการควบคุมแบบใช้ Reinforcement Learning โดยนำมาเปรียบเทียบประสิทธิภาพในการทำงานกับ Model Reinforcement Learning ที่ออกแบบขึ้นมาเอง

4.2.1 รายละเอียดการทดลอง

เริ่มการทดลองโดยทำการกำหนด Function ที่ใช้ในการทำงานของ Reinforcement Learning โดยเป็น Algorithm ที่อ้างอิงจากข้อมูลภายในงานวิจัย TSCRL ทำการกำหนดค่าระยะเวลาที่ใช้ควบคุมสัญญาณไฟเขียวของระบบไว้ที่ 60 วินาทีในแต่ละแยก โดยจะทำการ Simulation โดยมีข้อมูลยานพาหนะตามไฟล์ Rou ที่กำหนดโดยไฟล์ Rou นี้จะเป็นไฟล์เดียวกันกับการทดลองที่ 4.1 เพื่อทำการกำหนดตัวแปรควบคุมในการทดลอง ทำการ Simulation ไปจนสิ้นสุดการทำงานโดย ในการกำหนดไฟล์ Rou นั้นก็จะมีขนาดของความหนาแน่นที่แตกต่างกันทั้งหมด 3 ความหนาแน่น คือ มาก กลาง และ น้อย โดยในแต่ละแผนที่ ที่ใช้ในการ Simulation ทำการสังเกตผลลัพธ์ และ ทำการบันทึกผลการทดลอง

4.2.2 วิธีการทดลอง

1) ตั้งค่าการทำงานภายในไฟล์ Runner โดยทำการเลือกแผนที่แบบ 4-way เพื่อทำการทดลองกับแผนที่ขนาดเล็ก
2) ทำการเลือก Function ในการทำงาน การทดลองแบบ Reinforcement Learning
3) กำหนดค่าในไฟล Runner ให้เลือก Agent เป็น TSCRL Model
4) กำหนด Path ของไฟล์เส้นทางยานพาหนะ ในส่วนของการ Config Environment ของระบบ โดยทำการเริ่มด้วยไฟล์ความหนาแน่นน้อย
5) ทำการ Run ไฟล์ Runner.py เพื่อเริ่มต้นการ Simulation
6) ทำการ Run ไฟล์ Plotter.py เพื่อทำการเปิดหน้าต่างของโปรแกรมแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

7) ติดตามผลการทดลองจาก หน้าต่าง Plotter และ ทำการบันทึกผลการทดลอง
8) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 3 ถึง การทดลองที่ 6 โดยทำการเปลี่ยน Path ความหนาแน่นเป็น ความหนาแน่นกลาง และ ความหนาแน่นมาก และทำการบันทึกผลการทดลอง
9) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 1 ถึง 8 โดยทำการเลือกแผนที่ ที่ใช้ในการทำงานแบบ 16-way และ 36-way เพื่อทำการ Simulation ในแผนที่ขนาดกลาง และ แผนที่ขนาดใหญ่

4.3 การทดลองระบบไฟจราจรด้วย Reinforcement Learning ของปริญญานิพนธ์

การทดลองสุดท้ายเป็นการทดลองการทำงาน และ การวัดงานวิจัยเล่มนี้ โดยเป็นการทดลองการควบคุมระบบสัญญาณไฟจราจรด้วย Model Reinforcement Learning ที่ทำการออกแบบขึ้นมาเอง โดยใช้ทฤษฎีในงานวิจัย TSCRL เป็น Model อ้างอิง และ ทำการพัฒนาในส่วนต่างๆเพิ่มเติมเพื่อเพิ่มประสิทธิภาพในการทำงานที่ดียิ่งขึ้น โดยจะทำการวัดผลเปรียบเทียบกับการทดลองที่ 4.1 และ 4.2

4.3.1 รายละเอียดในการทดลอง

การทดลองจะเริ่มจากการเปลี่ยน Function ที่ใช้ในการทำงานของ Reinforcement Learning โดยจะทำการเปลี่ยนจาก Algorithm ที่อ้างอิงจากงานวิจัย TSCRL เป็น Algorithm ที่ออกแบบขึ้นมาเองในงานวิจัยครั้งนี้ ซึ่งจะไม่ต้องทำการกำหนดระยะเวลาที่ใช้ควบคุมสัญญาณไฟเขียวเหมือนการทดลองอื่น ๆ เพราะภายใน Algorithm ที่ทำการออกแบบนั้นมีการถ่วงน้ำหนักของสัญญาณไฟเขียวด้วยความหนาแน่นอยู่แล้ว ซึ่งวิธีในการทดลองจะเหมือนกับในการทดลองที่ 4.1 และ 4.2 โดยทำการ Simulation โดยใช้ข้อมูลยานพาหนะไฟล์ Rou ทั้ง 3 ความหนาแน่นซึ่งก็คือ ความหนาแน่น น้อย กลาง และ มาก ที่เป็นไฟล์เดียวกันกับ 2 การทดลองแรกเพื่อที่จะกำหนดตัวแปรควบคุมในการทดลองเมื่อทำการ Simulation เสร็จสิ้นแล้วก็จะทำการเปลี่ยนแผนที่ที่ใช้ในการ Simulation ทำการสังเกตผลลัพธ์ผ่านกราฟแบบ Real Time และ ทำการบันทึกผลการทดลอง

4.3.2 วิธีการทดลอง

1) ตั้งค่าการทำงานภายในไฟล์ Runner โดยทำการเลือกแผนที่แบบ 4-way เพื่อทำการทดลองกับแผนที่ขนาดเล็ก
2) ทำการเลือก Function ในการทำงาน การทดลองแบบ Reinforcement Learning
3) กำหนดค่าในไฟล Runner ให้เลือก Agent เป็น Model ที่ออกแบบ
4) กำหนด Path ของไฟล์เส้นทางยานพาหนะ ในส่วนของการ Config Environment ของระบบ โดยทำการเริ่มด้วยไฟล์ความหนาแน่นน้อย
5) ทำการ Run ไฟล์ Runner.py เพื่อเริ่มต้นการ Simulation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

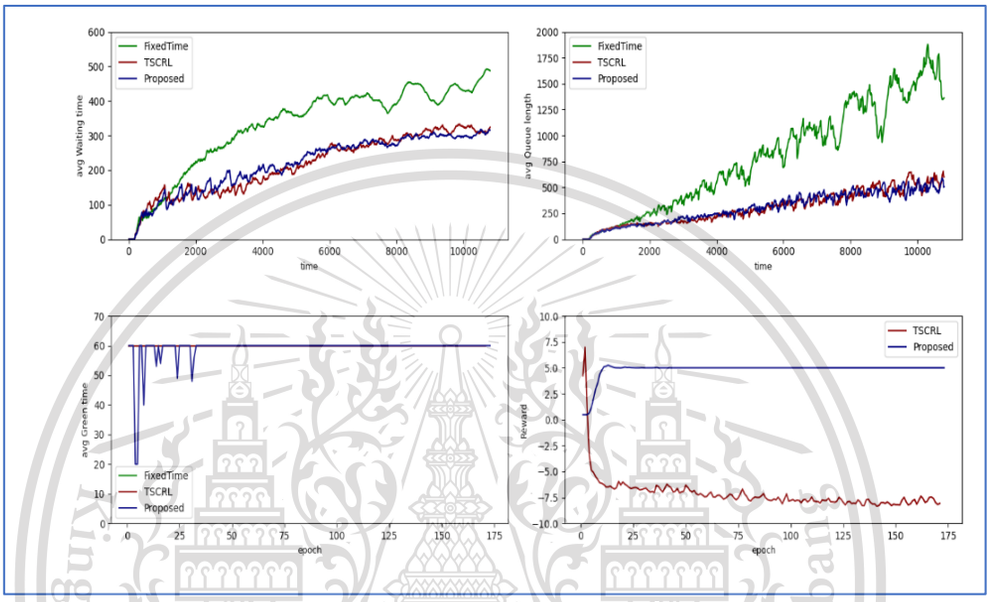
6) ทำการ Run ไฟล์ Plotter.py เพื่อทำการเปิดหน้าต่างของโปรแกรมแสดงผล
7) ติดตามผลการทดลองจาก หน้าต่าง Plotter และ ทำการบันทึกผลการทดลอง
8) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 3 ถึง การทดลองที่ 6 โดยทำการเปลี่ยน Path ความหนาแน่นเป็น ความหนาแน่นกลาง และ ความหนาแน่นมาก และทำการบันทึกผลการทดลอง
9) ทำการทดลองซ้ำ ตามขั้นตอนการทดลองที่ 1 ถึง 8 โดยทำการเลือกแผนที่ ที่ใช้ในการทำงานแบบ 16-way และ 36-way เพื่อทำการ Simulation ในแผนที่ขนาดกลาง และ แผนที่ขนาดใหญ่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

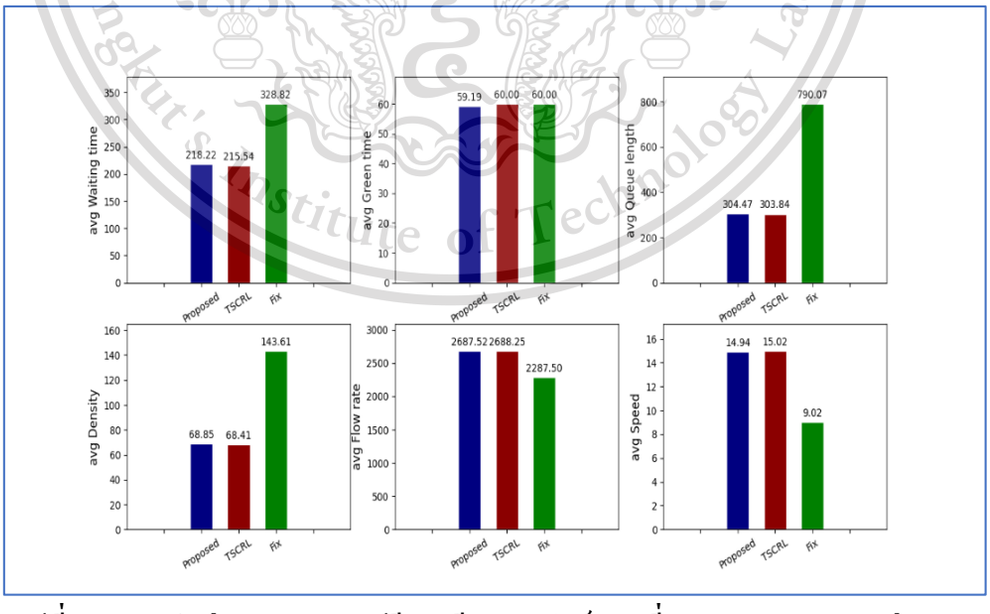
4.4 กราฟสรุปผลการทดลอง

เมื่อทำการทดลองที่ 4.1 ถึง 4.3 ครบทุกการทดลองเสร็จสิ้นแล้วจำเป็นต้องทำการสรุปความแตกต่างกันของผลลัพธ์ที่ได้ โดยทำการนำค่าที่บันทึกเป็นไฟล์ CSV ในแต่ละการทดลอง นำมาอ่านค่าจะทำการ Plot กราฟสรุปผลการทดลอง โดยผลลัพธ์ที่ได้จากการ Plot กราฟจะมีดังนี้

4.4.1 กราฟสรุปผลการทดลองสำหรับแผนที่ 4-way ยานพาหนะความหนาแน่น มาก



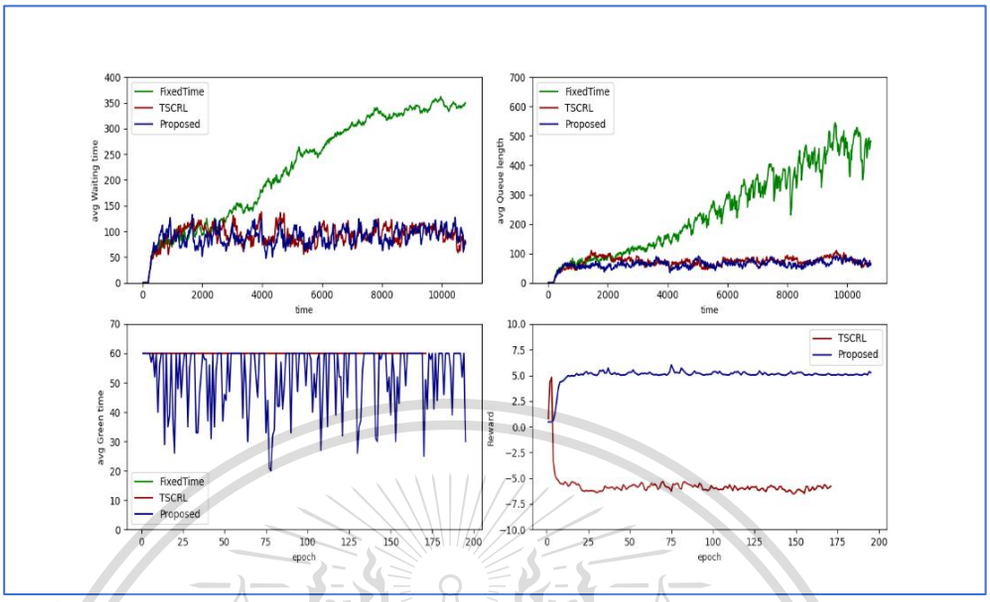
รูปที่ 4.1 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น มาก



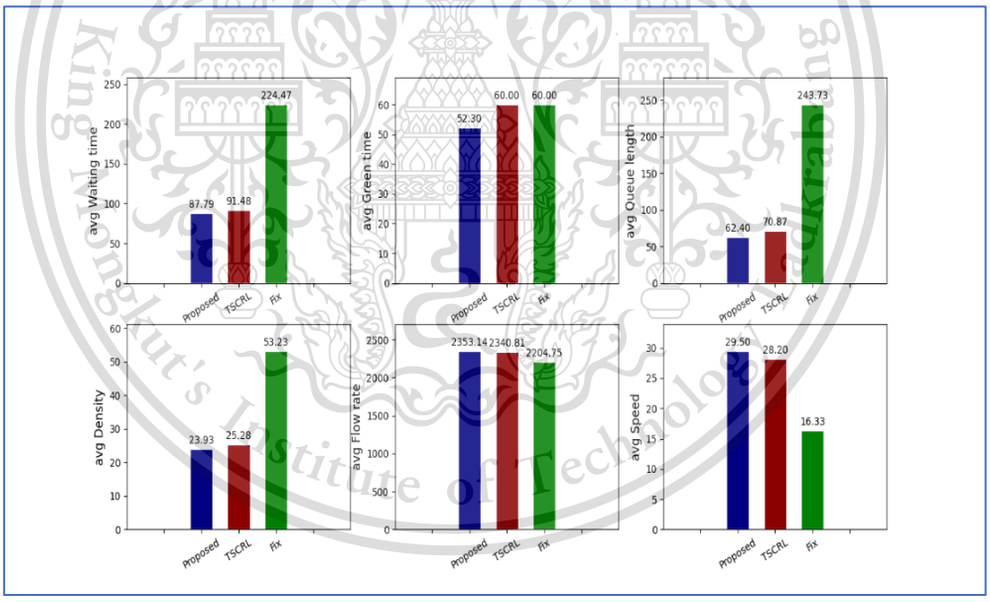
รูปที่ 4.2 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.2 กราฟสรุปผลการทดลองสำหรับแผนที่ 4-way ยานพาหนะความหนาแน่น กลาง



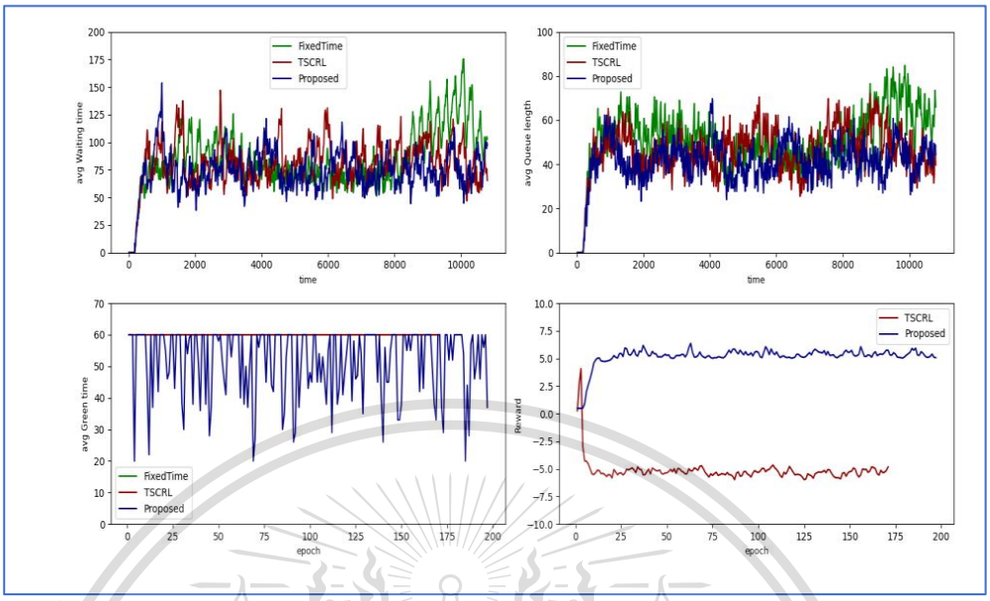
รูปที่ 4.3 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น กลาง



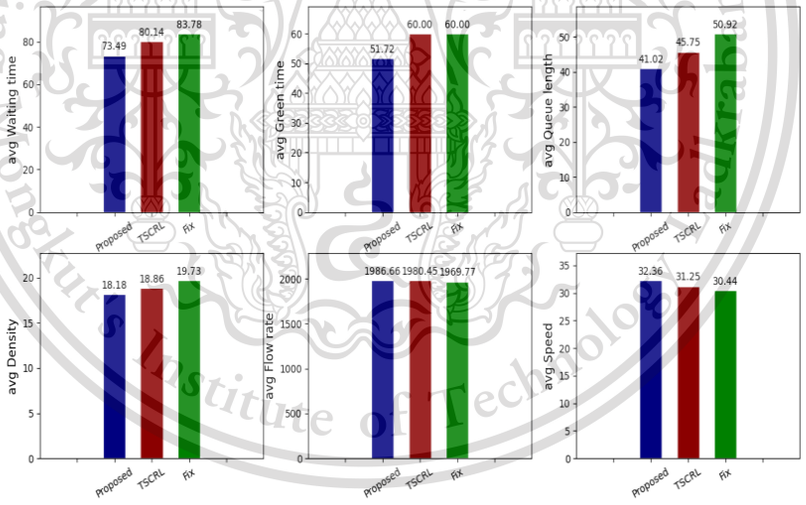
รูปที่ 4.4 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.3 กราฟสรุปผลการทดลองสำหรับแผนที่ 4-way ยานพาหนะความหนาแน่น น้อย



รูปที่ 4.5 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น น้อย

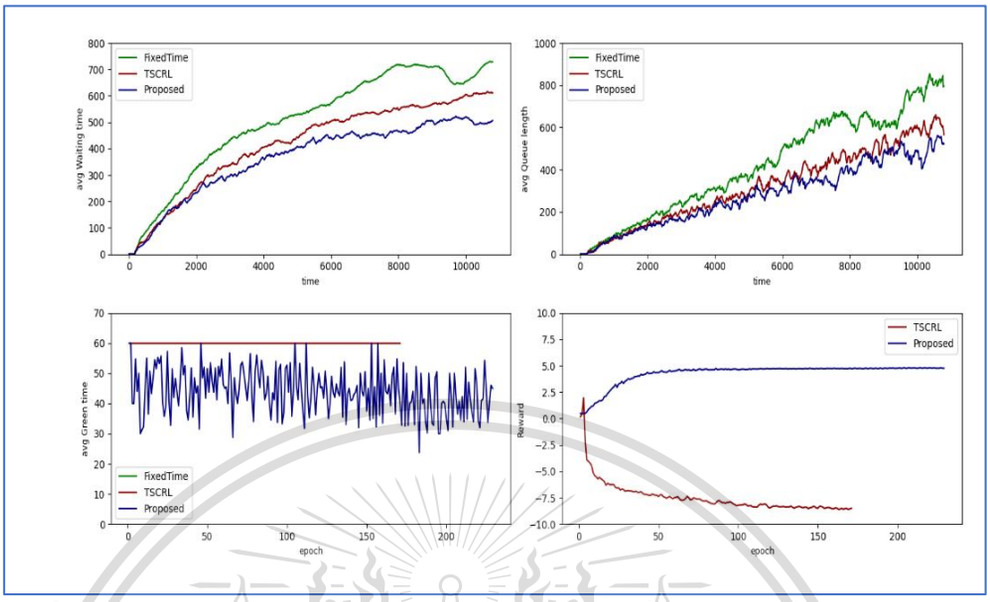


รูปที่ 4.6 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 4 way ความหนาแน่น น้อย

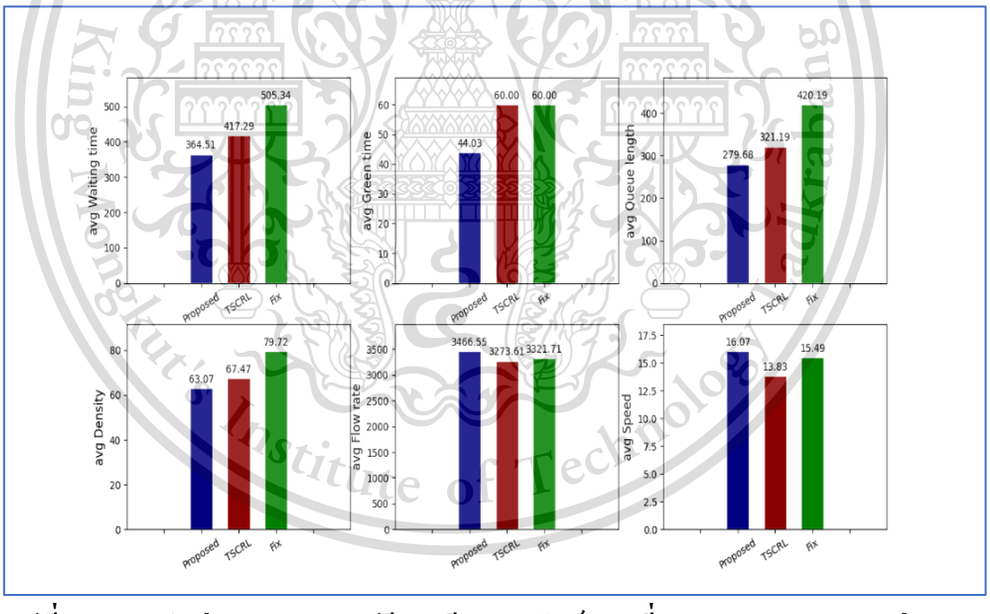
สรุปผลการทดลองของแผนที่ 4 way ในระดับความหนาแน่นของยานพาหนะมาก และ กลาง นั้น การทดลองของวิทยานิพนธ์ และ วิธี TSCRL นั้นผลลัพธ์ค่อนข้างใกล้เคียงกันมากทั้งด้านของ ค่าเฉลี่ย Waiting Time, Flow Rate, Density, Queue Length และ Speed โดยในส่วนของระดับความหนาแน่นน้อยผลลัพธ์ของวิทยานิพนธ์จะค่อนข้างดีกว่าผลลัพธ์ของวิธี TSCRL อย่างเห็นได้ชัด โดยผลลัพธ์ของวิธีแบบ Fix เวลาจะให้ผลลัพธ์ที่ต่ำที่สุดในการทดลองทั้ง 3 วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.4 กราฟสรุปผลการทดลองสำหรับแผนที่ 16-way ยานพาหนะความหนาแน่น มาก



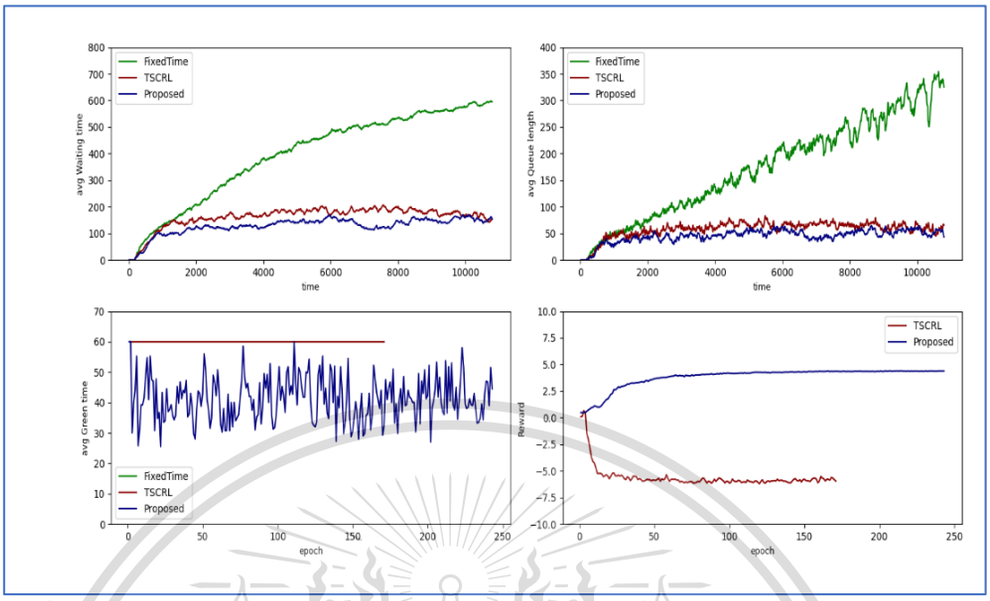
รูปที่ 4.7 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น มาก



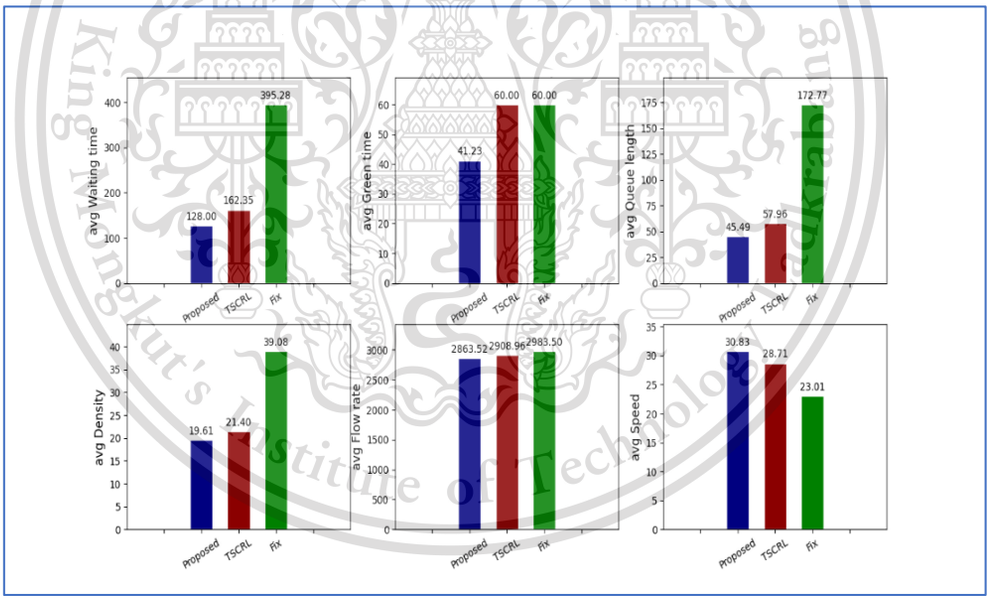
รูปที่ 4.8 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.5 กราฟสรุปผลการทดลองสำหรับแผนที่ 16-way ยานพาหนะความหนาแน่น กลาง



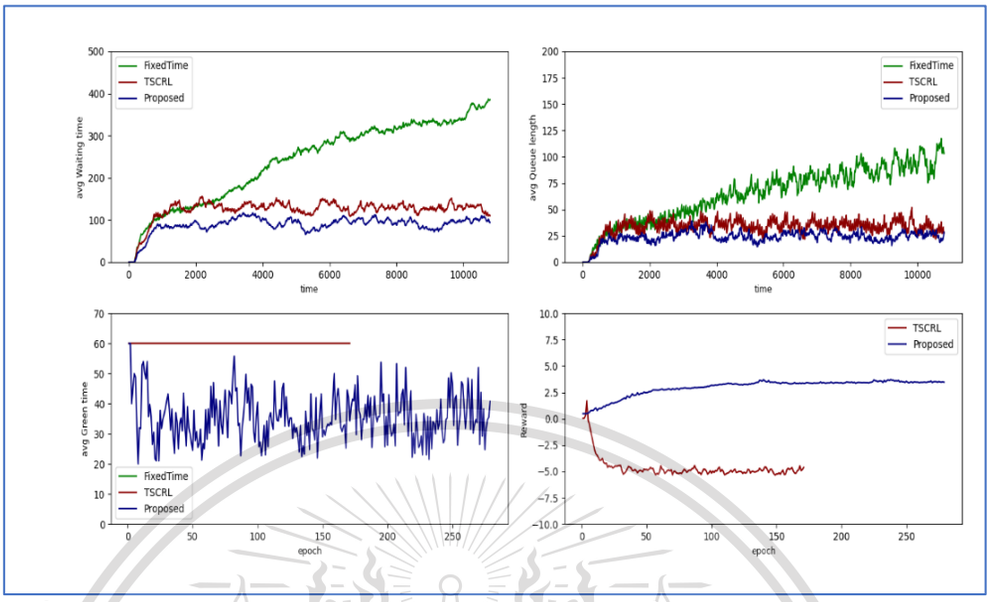
รูปที่ 4.9 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น กลาง



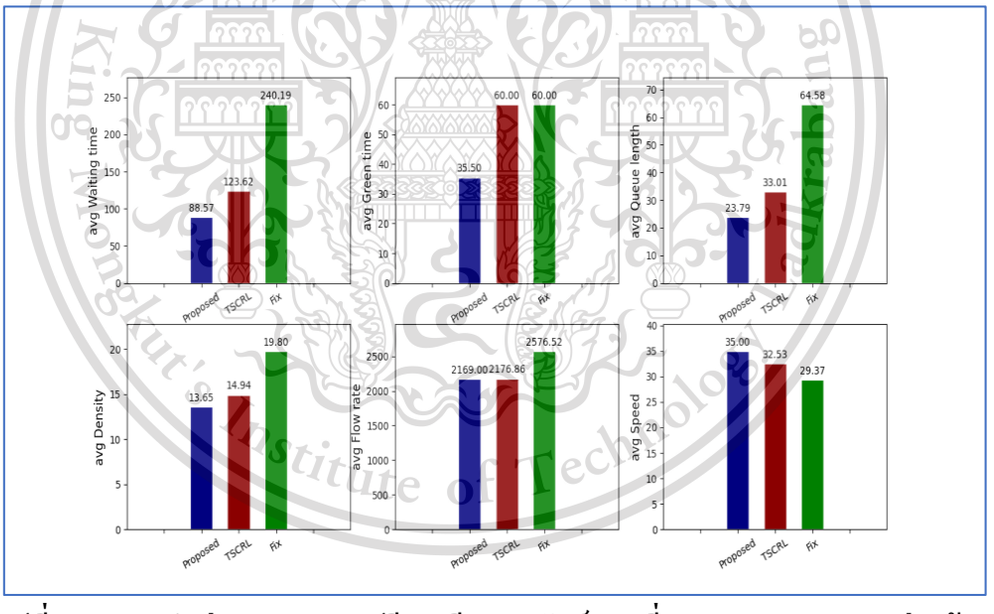
รูปที่ 4.10 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.6 กราฟสรุปผลการทดลองสำหรับแผนที่ 16-way ยานพาหนะความหนาแน่น น้อย



รูปที่ 4.11 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น น้อย

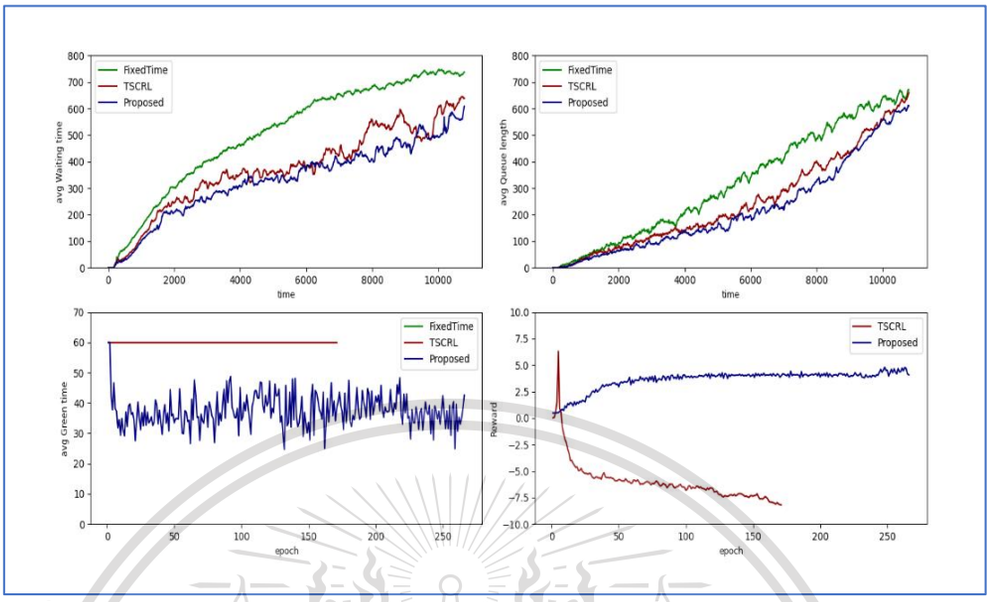


รูปที่ 4.12 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 16 way ความหนาแน่น น้อย

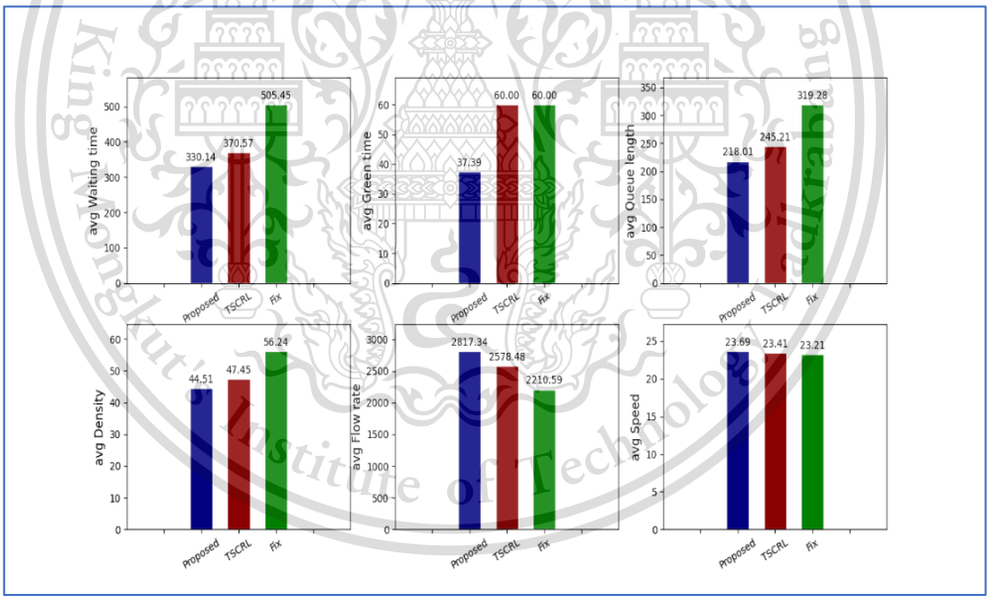
สรุปผลการทดลองของแผนที่ 16 way ในระดับความหนาแน่นของยานพาหนะทั้ง 3 ระดับนั้น ผลของการทดลองไปในทิศทางเดียวกันโดย การทดลองของวิทยานิพนธ์ จะให้ผลลัพธ์ที่ดีที่สุดของทั้ง 3 การทดลอง โดยผลลัพธ์นั้นจะไล่เลี่ยกับในส่วนของวิธี TSCRL ที่ให้ผลลัพธ์รองลงมา และ ในส่วนของวิธีการ Fix ระยะเวลาไฟจราจรจะให้ผลลัพธ์ที่แย่ที่สุดอย่างเห็นได้ชัด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.7 กราฟสรุปผลการทดลองสำหรับแผนที่ 36-way ยานพาหนะความหนาแน่น มาก



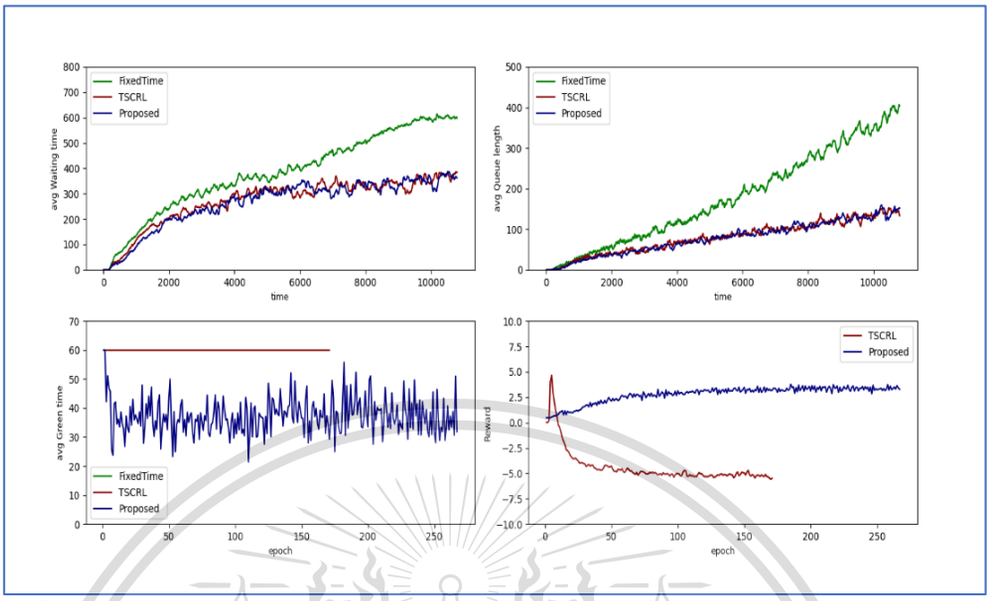
รูปที่ 4.13 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น มาก



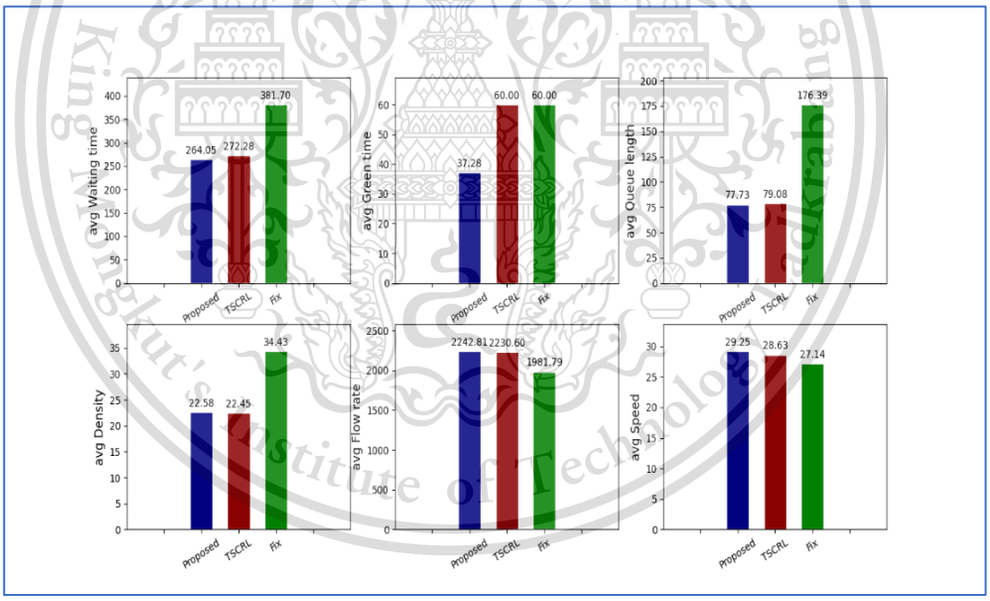
รูปที่ 4.14 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.8 กราฟสรุปผลการทดลองสำหรับแผนที่ 36-way ยานพาหนะความหนาแน่น กลาง



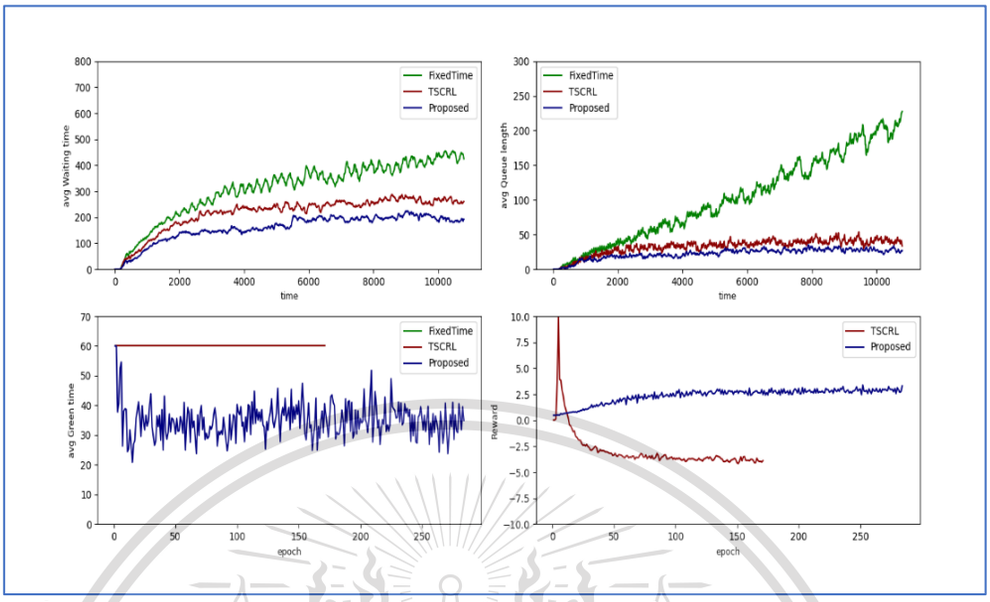
รูปที่ 4.15 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น กลาง



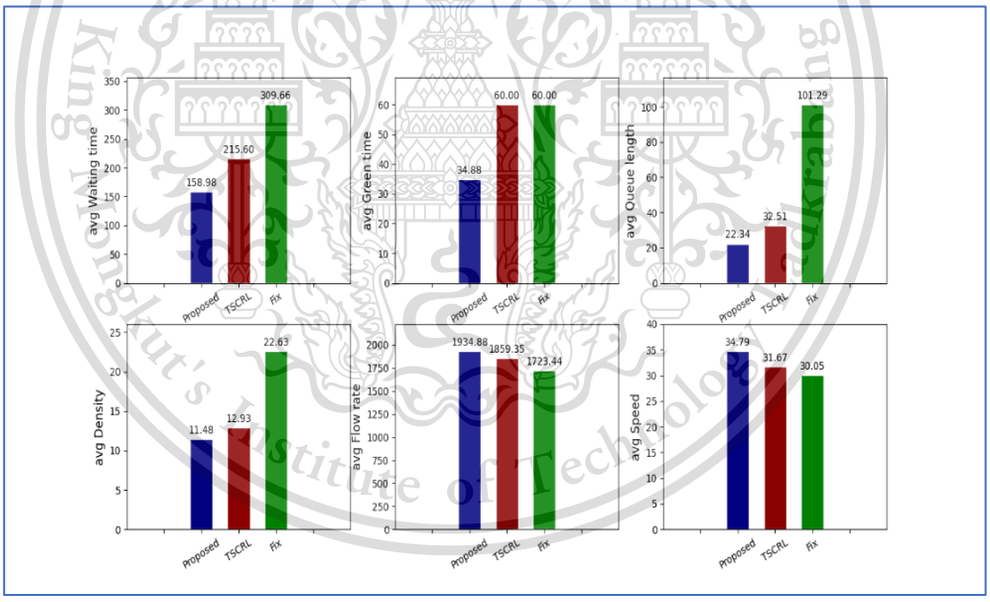
รูปที่ 4.16 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น กลาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

4.4.9 กราฟสรุปผลการทดลองสำหรับแผนที่ 36-way ยานพาหนะความหนาแน่น น้อย



รูปที่ 4.17 กราฟเส้นแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น น้อย



รูปที่ 4.18 กราฟแท่งแสดงความเปรียบเทียบผลลัพธ์แผนที่ 36 way ความหนาแน่น น้อย

สรุปผลการทดลองของแผนที่ 36 way ในระดับความหนาแน่นของยานพาหนะกลาง การทดลองของวิทยานิพนธ์ และ วิธี TSCRL นั้นจะให้ผลลัพธ์ค่อนข้างใกล้เคียงกันมากทั้งด้านของค่าเฉลี่ย Waiting Time, Flow Rate, Density, Queue Length และ Speed โดยในส่วนของ ระดับความหนาแน่นมาก และ น้อย ผลลัพธ์ของวิทยานิพนธ์จะค่อนข้างดีกว่าผลลัพธ์ของวิธี TSCRL อย่างเห็นได้ชัด และเช่นเดียวกับแผนที่อื่นๆผลลัพธ์ของวิธีแบบ Fix เวลาจะให้ผลลัพธ์ที่ต่ำที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บทที่ 5

บทสรุปและข้อเสนอแนะ

5.1 บทสรุป

ปริญญานิพนธ์เล่มนี้เป็นการออกแบบการทำงานของระบบที่ใช้ในการควบคุมสัญญาณไฟจราจร โดยจะมีการเปรียบเทียบวิธีที่ใช้งาน ทั้งสิ้น 3 วิธี คือ

1) การควบคุมแบบ Fix เวลา
2) การควบคุมโดย Reinforcement Learning โดยใช้การอ้างอิงตามงานวิจัย TSCRL
3) การควบคุมโดย Reinforcement Learning ที่ทำการพัฒนา

โดยระยะเวลาของสัญญาณไฟเขียว ได้ออกแบบโดยการทำการตั้งค่าพื้นฐานไว้ที่ 60 วินาที เพื่อใช้ในการทำงานของวิธี Fix เวลา และ วิธีอ้างอิงจาก TSCRL ในส่วนของวิธีที่ทำการออกแบบขึ้นมาเองจะใช้ระยะเวลา 60 วินาทีเป็นจุดอ้างอิงแล้วทำการถ่วงน้ำหนักด้วยการคำนวณความหนาแน่น เพื่อหาระยะเวลาของสัญญาณไฟเขียว ในส่วนของการออกแบบ Environment ของความหนาแน่นยานพาหนะในระบบให้มีความแตกต่างกันออกไปโดย 3 ระดับ คือ

1) ความหนาแน่นยานพาหนะในระบบ มาก
2) ความหนาแน่นยานพาหนะในระบบ กลาง
3) ความหนาแน่นยานพาหนะในระบบ น้อย

การออกแบบ Environment ของแผนที่ ที่ใช้ในการ Simulation โดยเป็นเชิงเปรียบเทียบระหว่างแผนที่ และ สภาพการจราจร ด้วยระบบการจราจรขนาดเล็ก ขนาดกลาง และ ระบบการจราจรขนาดใหญ่ โดยแผนที่ที่ทำการออกแบบมีทั้งหมด 3 แผนที่ดังนี้

1) แผนที่บริเวณ 4 แยกไฟจราจรขนาด 1*1
2) แผนที่บริเวณ 4 แยกไฟจราจรขนาด 2*2
3) แผนที่บริเวณ 4 แยกไฟจราจรขนาด 3*3

เมื่อทำการทดลองตามขั้นตอนที่ได้ทำการออกแบบ ก็จะทำการวัดผลด้วยการทำ Visualization ที่ออกมาในลักษณะของกราฟ เพื่อทำการเปรียบเทียบผล และ วัดประสิทธิภาพการทำงานทั้ง 3 วิธี โดยเมื่อสรุปผลลัพธ์ที่ได้จากการทดลองแล้วจะมีลักษณะดังนี้ ผลการทดลองของแผนที่ 4 way ระดับความหนาแน่นมาก และ กลาง แผนที่ 36 way ระดับความหนาแน่นปานกลาง วิธีของวิทยานิพนธ์ และ TSCRL จะให้ผลลัพธ์ที่ค่อนข้างใกล้เคียงกัน ในส่วนของการทดลองอื่นๆ วิธีของวิทยานิพนธ์ จะให้ผลลัพธ์ดีที่สุด วิธี TSCRL จะให้ผลลัพธ์ที่รองลงมา ในส่วนของวิธีแบบ Fix เวลาจะให้ผลลัพธ์ต่ำที่สุดในการทดลองทั้ง 3 วิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

5.2 ปัญหาและอุปสรรค

จากการลงความเห็นเรื่องแนวทางการทำงานของ Model และการทดลอง ที่ออกแบบในรายวิชาโปรเจค 1 ได้ข้อสรุปโดย ทำการแก้ไขการทดลอง และ Model Reinforcement Learning ใหม่ทั้งหมด จึงทำให้มีส่วนที่ต้องแก้ไขภายในรายวิชาโปรเจค 2 เป็นจำนวนมาก จึงทำให้การทำวิจัยในครั้งนี้ค่อนข้างลำบากด้วยปริมาณเนื้อหางานที่จะต้องทำการแก้ไขค่อนข้างมาก และ ระยะเวลาที่มีอย่างจำกัด โดยการวางแผนในการออกแบบการทำงานในส่วนของ Code ไม่ดีเท่าที่ควรจึงทำให้ต้องทำการแก้ไขในบางส่วนอยู่ตลอดเพราะไม่สามารถใช้งานได้กับแผนที่การจำลองที่ต่างออกไป โดยปัญหาการทำงานของ Algorithm ในการเรียนรู้ของ Reinforcement Learning นั้นยังมีความล่าช้าอยู่ในกรณีที่สภาพการจราจร หรือ Environment ของระบบมีการเปลี่ยนแปลงที่กะทันหันจนเกินไป ทำให้ประสิทธิภาพที่ออกมาไม่ดีเท่าที่ควร ส่วนต่อมาของปัญหา คือการสื่อสารระหว่างสมาชิกในการทำงานมีการเกิดปัญหาในการสื่อสารบ้างเป็นบางครั้ง ทำให้งานในส่วนที่เกิดปัญหาจะต้องทำการแก้ไขเพิ่มเติม

5.3 แนวทางการพัฒนาต่อ

แนวทางในการพัฒนางานวิจัยต่อไป จะเริ่มจากการนำ Model Reinforcement Learning ที่ได้นำมาพัฒนาต่อโดยจะไปในทิศทางที่นำมาประยุกต์ในการใช้งานจริงตามท้องถนน โดยเริ่มจากการนำข้อมูลสภาพการจราจรจริงที่มีการเก็บสถิติ หรือ นำข้อมูลสภาพการจราจรจาก Google API นำมาใช้งานเพื่อดึงความหนาแน่นของการจราจร เพื่อนำมาใช้อ้างอิงในการทำ Simulation โดยทำการปรับ Model Reinforcement Learning แก้ไขค่า Parameters ต่างๆ ให้เหมาะสมกับชุดข้อมูลตามสภาพการจราจรจริง โดยทำการแก้ไข Algorithm ในการคำนวณภายใน Model เพื่อลดเวลาที่ใช้ในการเรียนรู้ของ Reinforcement Learning ให้เร็วยิ่งขึ้น โดยหลังจากทำการปรับ Model ให้เข้ากับชุดข้อมูลการจราจรจริง ก็จะเพิ่มเติมในส่วนของแผนที่ที่ใช้ในการจำลอง โดยการใช้ Open Street Map เข้ามาใช้ในการดึงแผนที่การจราจรในประเทศ เพื่อทำการสร้างแผนที่เสมือนจริงเพื่อทำการทดสอบภาพรวมของระบบ โดยในส่วนสุดท้าย ของแนวทางในการพัฒนาสูงสุดของงานวิจัยที่คาดหวัง คือการใช้ฐานของมูลของสภาพการจราจรที่ได้จากกล้อง CCTV บริเวณ 4 แยกไฟจราจรเพื่อทำการวัดค่าของความหนาแน่นของ ยานพาหนะ ที่เข้ามาในระบบแบบ Real Time

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

บรรณานุกรม

[1] Hyunjin Joo, Yujin Lim and Syed Hassan Ahmed. 2020. “Traffic signal control for smart cities using Reinforcement learning.” Ph.D.Thesis of Sookmyung Women University. and Georgia Southern University.

[2] Lachlan M. McDowall, and Roger A. L. Dampney. 2006. “Calculation of threshold and saturation points of sigmoidal baroreflex function curves.” School of Medical Sciences (Physiology) and Bosch Institute for Biomedical Research, University of Sydney, Australia

[3] Roger, P. Elena, S. and William, R. 2011. **Traffic Engineering.** USA : PEARSON.

[4] German Aerospace Center (DLR). 2020. **Simulation of Urban MObility.** [Online].Available : https://www.eclipse.org/sumo/

[5] Chatchawan Niyomthum. 2560. **Reinforcement Learning (RL) 101 : Q-Learning.** [Online].Available : https://medium.com/@thebear19/reinforcement-learning-rl-101-q-learning-e84c2e92d448

[6] Mr.P L. 2561. **เริ่มเรียน Machine Learning 0-100 zero to Mr.incredible (Introduction).** [Online]. Available : https://medium.com/mmp-li/เริ่มเรียน-machine-learning-0-100-introduction-1c58e516bfcd

[7] MC.AI. 2561. **EP.1-Reinforcement Learning เบื้องต้น.** [Online].Available : https://medium.com/asquarelab/ep-1-reinforcement-learning-เบื้องต้น-acfa9d42394c

[8] สำนักงานนโยบายและแผนการขนส่งและจราจร. 2559. **“การจัดระบบจราจร”,** กรุงเทพ : สำนักงานนโยบายและแผนการขนส่งและจราจร.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

ภาคผนวก ก

ส่วนประกอบของ Reinforcement Learning Module

โปรแกรมที่ ก.1 Function เลือก Next State

```
def getNextState(self, qLength, moveState, waitingTime):
    for i in range(len(waitingTime)):
        if waitingTime[i] > 300:
           return i
    length = qLength
    maxLength = max(length)
    nextState = length.index(maxLength)
    while nextState in moveState:
        length[nextState] = -1
        maxLength = max(length)
        nextState = length.index(maxLength)
    print(nextState)
    return nextState
```

โปรแกรมที่ ก.2 Function การคำนวณ Reward ของงานวิจัย

```
def getSrlReward (self, data):
    FR = data['flowRate']
    WT = data['waitingTime']
    flowrateExpo = self.find_flowrate_expo(FR)
    waitingtimeExpo = self.find_waitingtime_expo(WT)
    FlowRateScale = 1/(1+math.exp(flowrateExpo))
    WaitingTimeScale = 1/(1+math.exp(waitingtimeExpo))
    reward = FlowRateScale/(1+WaitingTimeScale)
    return reward
```

โปรแกรมที่ ก.3 Function การคำนวณ Reward ที่เขียนโดย Hyunjin Joo, Yujin Lim and Syed Hassan Ahmed.

```
def getPrlReward (self, data):
    expo = -0.003930312*(data['arrivalRate'] - 750)
    alpha = 1/(1+math.exp(expo))
    func = (alpha * data['qStd']) + ((1-alpha) *
(math.pow(self.TAU, data['flowRate'])))
    reward = math.log(func, self.DELTA)
    return reward
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ก.4 Function หา Action ด้วย eGreedy

```
def eGreedy(self, currentState):
    action = 0
    stateData = self.stateSpace[currentState]
    if (self.EXPLORE_RATE > random.uniform(0, 1)) or
       (stateData["sumQ"] == 0.0):
       #explore
       print('explore' ,end=' | ')
       action = self.randomAction(currentState)
    else:
        #exploit
        print('exploit' ,end=' | ')
        for count in range(self.MAX_ACTIONS):
            if self.actionVerify(currentState, count):
                if (stateData["qValue"][count] ==
                    stateData["maxQ"]):
                    action = count
    return action
```

โปรแกรมที่ ก.5 Function หา Action ด้วย pGreedy

```
def pGreedy(self, currentState):
    action = 0
    stateData = self.stateSpace[currentState]
    if (self.EXPLORE_RATE > random.uniform(0, 1)) or
       (stateData["sumQ"] == 0.0):
        #explore
        print('explore' ,end=' | ')
        action = self.randomAction(currentState)
    else:
        #exploit
        print('exploit' ,end=' | ')
        action = self.randomAction(currentState)
        for count in range(self.MAX_ACTIONS):
            if self.actionVerify(currentState, action):
               prob = (stateData["qValue"][action]) /
               stateData["sumQ"]
               if prob > random.uniform(0, 1):
                  return action
            action += 1
            if action == self.MAX_ACTIONS:
               action = 0
        if count == self.MAX_ACTIONS:
            action = self.randomAction(currentState)
    return action
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ก.6 Function Take Action

```
def takeAction(self, action, currentState):
    green_state = [currentState] #which state can go?
    phase = [[''],[''],0]
    if action == 0:
       green_state.append(currentState + (-2 *
       (currentState % 2)) + 1)
    else:
       green_state.append((currentState + 4) % 8)
    g_phase = ''
    y_phase = ''
    for i in [7,6,5,4,3,2,1,0]:
        if i in green_state:
            g_phase += 'G' * (1 + (i % 2))
            y_phase += 'y' * (1 + (i % 2))
        else:
            g_phase += 'r' * (1 + (i % 2))
            y_phase += 'r' * (1 + (i % 2))
    phase[0] = g_phase
    phase[1] = y_phase
    return phase, green_state
```

โปรแกรมที่ ก.7 Function Find Green Time

```
def getGreenTime(self, moveLane, nextLane):
    moveDens = []
    nextDens = []

    for i in range(len(moveDens)):
        moveDens.append(
           traci.lane.getLastStepVehicleNumber
           (moveLane[i])*0.4)
        nextDens.append(
            traci.lane.getLastStepVehicleNumber
            (nextLane[i])*0.4)
        if sum(moveDens)==0 or sum(nextDens)== 0:
            self.greenTime.append(60)
            return 60
        greenTime = (sum(moveDens)/len(moveDens)) /
            (sum(nextDens)/len(nextDens)) * 60
        self.greenTime.append(greenTime)
        return greenTime
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ก.8 Function Update State Space

```
def update(self, currentState, nextState, action, data):
    reward = self.getSrlReward(data)
    self.reward.append(reward)
    self.setMaxQ()
    currentData = self.stateSpace[currentState]
    nextData = self.stateSpace[nextState]

    currentData["qValue"][action] +=
         round((self.LEARNING_RATE * (reward +
         (self.DISCOUNT_RATE * nextData["maxQ"])) -
         currentData["qValue"][action]), 5)

    self.stateSpace[currentState] = currentData
    self.setSumQ()
    self.count += 1

    if self.count == self.numJunc:
         self.EPOCH += 1
         avgGreen = round(sum(self.greenTime) /
                    len(self.greenTime), 2)
         avgRew = round(sum(self.reward) /
                  len(self.reward), 2)
         self.reward = []
         self.greenTime = []
         self.count = 0
         return [self.EPOCH - 1, avgGreen, avgRew]

    return None
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ภาคผนวก ข

ส่วนประกอบของ Simulation API Module

โปรแกรมที่ ข.1 Function SetLogic TrafficLight

```
def setLogic(self, phase):
    tlPhase = []
    tlPhase.append(traci.trafficlight.Phase
       (3, phase[1], 3, 3))

    tlPhase.append(traci.trafficlight.Phase
       (phase[2], phase[0], phase[2], phase[2]))
    logic = traci.trafficlight.Logic("0", 0, 0, tlPhase)
    traci.trafficlight.setProgramLogic(self.id, logic)
    self.cycle = phase[2] + 3
```

โปรแกรมที่ ข.2 Function ที่ใช้ในการหา Average Speed

```
def get_avg_speed(self, edgeId):
    avg_spd = 0
    for e_id in edgeId:
        if traci.edge.getLastStepVehicleNumber(e_id)>0:
            avg_spd += traci.edge
                    .getLastStepMeanSpeed(e_id)
    return round(((avg_spd / 4) * 3.6), 2)
```

โปรแกรมที่ ข.3 Function ที่ใช้ในการหา Density

```
def get_density(self, edgeID):
    density = 0
    for e_id in edgeID:
        density +=
            traci.edge.getLastStepVehicleNumber(e_id) *
            1000 / traci.lane.getLength(e_id + '_0')
    return round((density / 4), 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ข.4 Function ที่ใช้ในการหา WaitingTime

```
def getWaitingTime(self, edgeID):
    wTime = []
    for eId in edgeID:
        vehs = traci.edge.getLastStepVehicleIDs(eId)
        if len(vehs) > 0:
           for vId in vehs:
               wait = traci.vehicle.
                      getAccumulatedWaitingTime(vId)
               if wait > 0:
                  wTime.append(wait)
    if len(wTime) > 0:
       return round((sum(wTime) / len(wTime)), 2)
    return 0
```

โปรแกรมที่ ข.5 Function ที่ใช้ในการหา Average Queue Length

```
def getAvgLength(self, edgeId):
lengthList = []
for edge in edgeId:
    lengthList.append(traci.lane
      .getLastStepHaltingNumber(edge+'_1') * 6.0)

    lengthList.append(traci.lane
      .getLastStepHaltingNumber(edge+'_0') * 6.0)
       avg_length = sum(lengthList) / len(lengthList)
       std_length = st.stdev(lengthList)
       return avg_length, round(std_length, 2)
```

โปรแกรมที่ ข.6 Function ที่ใช้ในการหาจำนวนยานพาหนะ ภายใน Last

```
def getLastLength(self, juncID):
    edgeID = self.edge[juncID]
    length = []
    for edge in edgeID:
        length.append(traci.lane
           .getLastStepHaltingNumber(edge+'_1') * 6.0)
        length.append(traci.lane
           .getLastStepHaltingNumber(edge+'_0') * 6.0)
    return length
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ข.7 Function ในการหา Arrival Rate จาก Detector

```
def addCarIn(self):
    juncIndex = 0
    for junc in self.edge.keys():
        for state in range(8):
            num = traci.inductionloop
                  .getLastStepVehicleNumber
                  ('a1_' + junc + '_' + str(state))
            curId = traci.inductionloop
                    .getLastStepVehicleIDs
                    ('a1_' + junc + '_' + str(state))
            if num > 0 and curId[0] !=
                     self.arrId[juncIndex][state]:
               self.arrId[juncIndex][state] = curId[0]
               self.carIn[juncIndex] += 1
        juncIndex += 1
```

โปรแกรมที่ ข.8 Function ในการหา Flow Rate จาก Detector

```
def addCarIn(self):
    juncIndex = 0
    for junc in self.edge.keys():
        for state in range(8):
            num = traci.inductionloop
                  .getLastStepVehicleNumber
                  ('e1_' + junc + '_' + str(state))
            curId = traci.inductionloop
                    .getLastStepVehicleIDs
                    ('e1_' + junc + '_' + str(state))
            if num > 0 and curId[0] !=
                     self.arrId[juncIndex][state]:
               self.arrId[juncIndex][state] = curId[0]
               self.carIn[juncIndex] += 1
        juncIndex += 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

ภาคผนวก ค

ส่วนประกอบของ Runner Module

โปรแกรมที่ ค.1 Function ที่ใช้ในการ Simulation แบบ FixedTime

```
def runNormal(api, tlsList, csvManage, csvRew):
    epoch = 0
    count = 0
    greenTime = 45
    phase = [0]*len(tlsList)
    tlsID = tlsList
    g_phase = ["rrrrrrrrrGGG", "rrrGGGrrrrrr",
               "rrrrrrGGGrrr", "GGGrrrrrrrrr"]
    y_phase = ["rrrrrrrrryyy", "rrryyyrrrrrr",
               "rrrrrryyyrrr", "yyyrrrrrrrrr"]
    traci.simulationStep()
    l_phase = [g_phase[phase[0]], y_phase[phase[0]]
               , greenTime]
    for tls in tlsID:
        tls.cycle = greenTime + 3
        tls.setLogic(l_phase)
    while (traci.simulation.getMinExpectedNumber()
           - traci.vehicle.getIDCount() != 0):
        try:
            result = api.simulate()
        except:
            print('Closed')
            return 0
        if result != None:
            csvManage.saveAvgResult(result[1])
            for tls in tlsID:
                tls.saveResult(result[0][tls.id])
        for tls in tlsID:
            index = tlsID.index(tls)
            if tls.isCycleEnd():
                phase[index] = (phase[index] + 1) % 4
                count += 1
                l_phase = [g_phase[phase[index]]
                           , y_phase[phase[index]]
                           , greenTime]
                tls.setLogic(l_phase)
                if count == len(phase):
                    count = 0
                    epoch += 1
                    csvRew.saveAvgResult([epoch,
                                          greenTime])
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ค.2 Function ที่ใช้ในการ Simulation แบบ TSCRL

```
def runPRL(api, agent, tlsList, csvManage, csvRew):
    traci.simulationStep()
    for tls in tlsList:
        tls.action = agent.getAction('e_greedy',
                     tls.currentState)
        phase, tls.moveState = agent.takeAction(tls
                           .action, tls.currentState)
        tls.setLogic(phase)
    while (traci.simulation.getMinExpectedNumber() -
           traci.vehicle.getIDCount() != 0):
        try:
            result = api.simulate()
        except:
            print('Closed')
            return 0
        if result != None:
            print(result)
            csvManage.saveAvgResult(result[1])
            for tls in tlsList:
                tls.saveResult(result[0][tls.id])
        for tls in tlsList:
            if tls.isCycleEnd():
                lastQueue = api.getLastLength(tls.id)
                if sum(lastQueue) == 0:
                    nextState = agent.getRandomState(
                                tls.moveState)
                else:
                    nextState = agent.getNextState(
                                lastQueue, tls.moveState)
                data = tls.getAvgResult()
                reward = agent.update(tls.currentState,
                         nextState, tls.action, data)
                if reward != None:
                    csvRew.saveAvgResult(reward)
                tls.currentState = nextState
                tls.action = agent.getAction('e_greedy'
                             , tls.currentState)
                printResult(tls.id, data, nextState
                  , tls.action)
                phase, tls.moveState = agent.takeAction(
                           tls.action, tls.currentState)
                tls.setLogic(phase)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
This material is reserved for educational use only, not allowed for commercial use.
Forbidden to modify the content, and cite the document when use.

โปรแกรมที่ ค.3 Function ที่ใช้ในการ Simulation แบบ Proposed

```
def runSRL(api, agent, tlsList, csvManage, csvRew):
    traci.simulationStep()
    for tls in tlsList:
        tls.action = agent.getAction('e_greedy',
                     tls.currentState)
        phase, tls.moveState = agent.takeAction(
                     tls.action, tls.currentState)
        phase[2] = agent.getGreenTime(tls.getMoveLane(),
                   tls.getNextLane())
        tls.setLogic(phase)
    while (traci.simulation.getMinExpectedNumber() -
           traci.vehicle.getIDCount() != 0):
        try:
            result = api.simulate()
        except:
            print('Closed')
            return 0
        if result != None:
            for tls in tlsList:
                tls.saveResult(result[0][tls.id])
        for tls in tlsList:
            if tls.isCycleEnd():
                lastQueue = api.getLastLength(tls.id)
                waitingTime = api.getLastWaiting(tls.id)
                print(lastQueue, waitingTime)
                if sum(lastQueue) == 0:
                    nextState = agent.getRandomState(
                                tls.moveState)
                else:
                    nextState = agent.getNextState(
                                lastQueue, tls.moveState
                                , waitingTime)
                data = tls.getAvgResult()
                reward = agent.update(tls.currentState,
                         nextState, tls.action, data)
                if reward != None:
                    csvRew.saveAvgResult(reward)
                tls.currentState = nextState
                tls.action = agent.getAction('e_greedy'
                             , tls.currentState)
                printResult(tls.id, data, nextState
                            , tls.action)
                phase, tls.moveState = agent.takeAction(
                           tls.action, tls.currentState)
                phase[2] = agent.getGreenTime(
                    tls.getMoveLane(), tls.getNextLane())
                tls.setLogic(phase)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.