รายงานสหกิจศึกษาฉบับสมบูรณ์

การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการวางตำแหน่งสถานีของรถไฟขนส่งสินค้าทางเดี่ยว

Development of Station Locating Software for Single-Track Freight Train

นายปาณินท์ ศรีสมบัติ

ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2560

รายงานสหกิจศึกษาฉบับสมบูรณ์

การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการวางตำแหน่งสถานีของรถไฟขนส่งสินค้าทางเดี่ยว

Development of Station Locating Software for Single-Track Freight Train

นายปาณินท์ ศรีสมบัติ

ภาควิชาวิศวกรรมเครื่องกล

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2560

ชื่อโครงงานสหกิจศึกษา การพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการวางตำแหน่งสถานีของรถไฟขนส่งสินค้าทางเดี่ยว

ชื่อ-สกุลนักศึกษา นายปาณินท์ ศรีสมบัติ

คณะ วิศวกรรมศาสตร์ **ภาควิชา** วิศวกรรมเครื่องกล

ชื่อ-สกุล อาจารย์นิเทศ ผศ.ดร.มนต์ศักดิ์ พิมสาร

ชื่อ-สกุล ผู้นิเทศงาน นายวสันต์ ฐิตวัฒน์

สถานประกอบการ บริษัท อิตาเลี่ยนไทย ดีเวล้อปเม้นต์ จำกัด (มหาชน)

บทคัดย่อ

การขนส่งรถไฟแบบทางเดี่ยวเป็นการเดินรถไฟในทั้งสองทิศทางโดยใช้รางรถไฟเพียงรางเดียวจึงทำให้มีโอกาสเกิดการชนกันของรถไฟสองขบวนที่สวนทางกัน ดังนั้นการวางตำแหน่งสถานีเพื่อใช้ในการหลีกของขบวนรถที่วิ่งสวนทางกันมาจึงจำเป็นอย่างมาก แต่เนื่องจากตำแหน่งและจำนวนสถานีก็เป็นปัจจัยหนึ่งที่ส่งผลกระทบต่อระยะเวลาการขนส่งของรถไฟแต่ละขบวน โปรแกรมตรวจสอบการวางตำแหน่งของสถานีจะช่วยในการตรวจสอบความปลอดภัยจากการชนกันของรถไฟที่สวนทางกันและช่วยในกาคำนวณหาระยะเวลาในการขนส่งได้สะดวกรวดเร็วกว่า และมีความแม่นยำการคำนวณและเขียนกราฟด้วยมือ อีกทั้งทำให้สามารถหาตำแหน่งการวางสถานีและระยะเวลาในการขนส่งได้อย่างที่ต้องการและพึงพอใจ ดังนั้นในโครงงานนี้ได้ทำการพัฒนาโปรแกรมการวางตำแหน่งของสถานีของรถไฟขนส่งสินค้าทางเดี่ยวด้วยโปรแกรม Matlab เพื่อใช้ในการออกแบบหาจำนวนรถและที่ตั้งสถานีที่เหมาะสมโดยมีเป้าหมายคือ ให้ใช้ระยะเวลาในการเดินรถต่อรอบของแต่ละขบวนน้อยที่สุด จากนั้นได้ยกตัวอย่างกรณีศึกษามา 1 กรณี และจากการใช้โปรแกรมที่พัฒนาขึ้นมาพบว่าในการออกแบบการจัดตั้งและจำนวนสถานีทำให้สามารถลดระยะเวลาการขนส่งได้จากเดิมที่คาดการณ์ไว้คือ 33 ชั่วโมง เป็น 32 ชั่วโมง 32 นาที ซึ่งลดลงไปจากเดิม 28.00 นาที (ลดไปร้อยละ 1.41) ซึ่งเป็นระยะเวลาในการขนส่งที่น้อยที่สุดและใช้จำนวนสถานีน้อยที่คือ 4 สถานี

Cooperative Title: Development of Station Locating Software for Single-Track Freight Train

Student intern name: Mr.Panin Srisombat

Faculty: Faculty of Engineering **Department:** Mechanical Engineering

Advisor name: Asst.Prof.Dr.Monsak Pimsarn

Mentor name: Mr.Vasan Thitawat

Company: Italian-Thai Development Public Company Limited

ABSTRACT

Single track railway transportation is the railway where trains traveling in both directions share the same track. Stations locating and train scheduling in a single track rail network are very important for controlling a train moving , to avoid train collision. But the number of the stations and location are the components that cause the cycle time to increase or decrease. Thus, station locating software for Single-Track freight train railway is used to detect the train collision, calculate the cycle time of each train and display a graph of train diagram. This software not only can compute an acceptable cycle time but also is quicker than the original way. Therefor, this project aims to develop a software for station locating using Matlab, and use it to find the optimal cycle time, number of trains and station locations. One example of case studies was carried out and showed that, the cycle time was decreased from 33 hr., original value , to 32 hr. and 32 minutes. The reduction time is 28 minutes or 1.41 percent. This value is minimal cycle time and the minimum number of stations is 4.

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดีจากการร่วมมือและความช่วยเหลือจากหลายฝ่ายโดยเฉพาะอย่างยิ่ง นายวสันต์ ฐิตวัฒน์ ผู้นิเทศสหกิจศึกษา และ อาจารย์มนต์ศักดิ์ พิมสาร อาจารย์ที่ปรึกษาสหกิจศึกษา ซึ่งท่านได้ให้ความใส่ใจ ความเมตตา และความรักแก่ข้าพเจ้าคณะผู้จัดทำเป็นอย่างมาก รวมทั้งให้คำปรึกษา คำแนะนำที่เปี่ยมไป ด้วยความรู้และประสบการณ์ซึ่งทำให้ข้าพเจ้าผู้จัดทำซาบซึ้งมากจึงขอขอบพระคุณมา ณ ที่นี้

ขอขอบคุณ อาจารย์จิตราภรณ์ วงศางาม ที่ให้ความรู้เพิ่มเติม คำปรึกษาและแนะนำเอกสารที่เกี่ยวข้องอันเป็นประโยชน์มากมายให้แก่ข้าพเจ้า

ขอขอบคุณสาขาวิชาวิศวกรรมขนส่งทางราง คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง สำหรับความรู้และแรงบันดาลใจสำหรับวิทยานิพนธ์ฉบับนี้ รวมไปถึงเพื่อนๆ จากสาขาวิศวกรรมขนส่งทางรางและวิศวกรรมเครื่องกลสำหรับกำลังใจและคำปรึกษาต่างๆ

ท้ายที่สุดนี้ขอขอบพระคุณบุคคลสำคัญที่สุดที่ทำให้ข้าพเจ้ามีวันนี้ ก็คือ บิดาและมารดา อันเป็นที่เคารพรักยิ่ง ซึ่งได้เลี้ยงดูผู้เขียนทั้งหมดมาเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และยังให้กำลังใจ เอาใจใส่เสมอมา ในทุก ๆ ด้านอันหาที่เปรียบมิได้ ข้าพเจ้าขอระลึกในพระคุณอันสูงสุดประมาณและขอกราบขอบพระคุณมา ณ ที่นี้

นายปาณินท์ ศรีสมบัติ

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญรูป.....VI

บทที่ 1 บทนำ.....1

1.1 ความเป็นมา.....1
1.2 วัตถุประสงค์.....1
1.3 ขอบเขตของโครงงาน.....2
1.4 วิธีการดำเนินงาน.....2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....2

บทที่ 2 งานวิจัย และทฤษฎีที่เกี่ยวข้อง.....3

2.1 งานวิจัยที่เกี่ยวข้อง.....3
2.2 ศึกษาข้อมูลเกี่ยวกับรถไฟขนแร่ถ่านหินทางเดี่ยว.....4
2.2.1 ความเร็ว.....4
2.2.2. ความเร่ง / ความหน่วง.....4
2.2.3. ระยะเบรก.....4
2.2.4. ความปลอดภัย.....4
2.3 สมการในการหาตำแหน่งและเวลาของรถไฟ ณ ตำแหน่งต่างๆ.....5
2.3.1 สมการหาระยะเวลาการเคลื่อนที่.....5
2.3.2 สมการหาจุดตัดของรถไฟสองขบวนที่สวนทางกัน.....6
2.4 สมการที่ใช้สำหรับเขียนฟังก์ชั่น.....7
2.5 การประมวลผลข้อมูลการทดลอง.....8
2.5.1 การคำนวณค่าความแตกต่าง.....8

บทที่ 3 วิธีดำเนินการ.....9

3.1. ลักษณะการขนส่งรถไฟขนส่งแร่ถ่านหิน.....9
3.2 วิเคราะห์เกี่ยวกับปัญหาที่ต้องการแก้ไขและปัจจัยที่เกี่ยวข้อง.....10
3.3 เริ่มต้นออกแบบการเขียนโปรแกรมด้วย Matlab.....14

สารบัญ

หน้า

3.3.1. สร้างตัวแปรที่เกี่ยวข้องและสามารถแทนค่าตัวแปรได้....16
3.3.2 กระบวนการหาตำแหน่งของรถไฟเที่ยวขาไป....16
3.2.3 กระบวนการหาตำแหน่งของรถไฟเที่ยวขากลับ....17
3.2.4. กระบวนการคำนวณหาระยะเวลาการเดินรถต่อรอบ....19
3.4 การใช้งานโปรแกรมและตัวอย่างการใช้งาน....19

บทที่ 4 ผลลัพธ์ของโครงงาน....21

4.1 บทนำ....21
4.2 ผลลัพธ์ของโครงงาน....21
4.3 เปรียบเทียบผลลัพธ์กับเป้าหมายโครงการของบริษัท....22
4.3 สรุปผล....34

บทที่ 5 สรุปผล และข้อเสนอแนะ....35

5.1 สรุปผล....24
5.2 ข้อเสนอแนะ....24

เอกสารอ้างอิง....25
ภาคผนวก....26

สารบัรูป

รูปที่ หน้า

3.1 รูปแบบการเดินรถและจุดสับหลีก....9

3.2 องค์ประกอบของระยะเวลาการเดินรถต่อรอบ....11

3.3 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 440+000....12

3.4 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 400+000....12

3.5 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 90+000, 210+000, 310+000, 410+000....13

3.6 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 120+000, 212+250+000, 312+502, 430+828....13

3.7 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000,160+000, 240+000, 320+000, 400+000....14

3.8 แผนภาพแสดงลำดับขั้นตอนการทำงานของโปรแกรม....15

3.9 แผนภาพแสดงลำดับขั้นตอนการหาตำแหน่งของรถเที่ยวขาไป....17

3.10 แผนภาพแสดงลำดับขั้นตอนการหาตำแหน่งของรถเที่ยวขากลับ....18

3.11 หน้าต่างอินเตอร์เฟสของโปรแกรม....19

3.12 หน้าอินเตอร์เฟสแสดงผลจากการทดลองคำนวณหาระยะเวลาการเดินรถต่อรอบ และกราฟแสดงการเดินรถ....20

4.1 หน้าต่างอินเตอร์เฟสแสดงผลลัพธ์จากการวางสถานีทั้ง 4 ในช่วงที่กำหนดไว้....21

บทที่ 1
บทนำ

1.1 ความเป็นมา

ในช่วงยุคแรกๆของการใช้รถไฟเป็นเครื่องมือในการขนส่งถ่านหินนั้น การออกแบบในการวางเส้นทางแต่ละเส้นได้ก่อสร้างขึ้นเพื่อให้เกิดประโยชน์สำหรับกิจการของบริษัทหรือองค์กรใดองค์กรหนึ่ง เมื่อการขนส่งแบบนี้มีการเติบโตมากยิ่งขึ้นการจัดการเส้นทางเหล่านี้ก็จะมีความซับซ้อนขึ้น มีการขยายตัวของเส้นทาง จากเหมืองไปสู่ท่าเรือขนส่ง และขนส่งไปยังโรงงานอุตสาหกรรมต่างๆ ไม่ใช่แค่เกิดการขยายตัวของเส้นทาง แต่ยังรวมไปถึงน้ำหนักและปริมาณในการขนส่งถ่านหินที่มากขึ้นตามไปด้วย ทำให้เกิดการใช้ขบวนรถที่มากขึ้นและการเดินรถที่ซับซ้อนมากขึ้น การจัดตารางการเดินรถไฟจึงเป็นสิ่งที่สำคัญมากอย่างหนึ่งเพื่อให้เกิดความปลอดภัยและตรงตามเป้าหมายของการเดินรถนั้น ๆ

ในการวางเส้นทางและออกแบบตารางการเดินรถของรถไฟทางเดี่ยว จึงจำเป็นต้องศึกษาและคำนึงถึงปัจจัยต่างๆที่เกี่ยวข้องกับโครงการนั้น ๆ ไม่ว่าจะเป็น จำนวนขบวนรถ ความเร็วของรถไฟ ความเร่งของรถไฟ ระยะเวลาในช่วงต่างๆของการเดินรถ ระยะเวลาและระยะห่างของการปล่อยรถออกแต่ละขบวน อีกทั้งยังรวมไปถึงความปลอดภัยต่างๆ เพราะการเดินรถไฟของเส้นทางรถไฟทางเดี่ยวจะเป็นการเดินรถแบบสวนทางกันโดยมีแค่รางคู่เดียวเท่านั้น นั่นจึงทำให้ต้องคำนึงถึงการตั้งสถานีหรือจุดต่างๆในการใช้เป็นช่องทางการหลีกของรถไฟเพื่อไม่ให้การชนของรถไฟหรืออุบัติเหตุเกิดขึ้น

โปรแกรมในการช่วยคำนวณและแสดงกราฟการเดินรถในช่วงต่างรถจึงเป็นส่วนหนึ่งในเพิ่มประสิทธิภาพในการออกแบบสถานีต่างๆของแนวเส้นทางรถไฟ อีกทั้งยังลดระยะเวลาในการคำนวณและเขียนกราฟ รวมไปถึงยังมีความแม่นยำมากกว่าการคำนวณและเขียนกราฟด้วยมือเป็นอย่างมาก

1.2 วัตถุประสงค์

1.2.1 เพื่อศึกษาวิธีการแก้ปัญหาเกี่ยวกับการวางตำแหน่งและจำนวนสถานีที่ส่งผลต่อระยะเวลาการเดินรถไฟ

1.2.2 เพื่อช่วยทางบริษัทในการลดระยะเวลาเดินรถไฟขนส่งแร่ถ่านหินและหาตำแหน่งการวางสถานีที่เหมาะสม

1.2.3 สามารถประยุกต์ใช้กับการออกแบบเส้นทางและตารางเดินรถของขบวนอื่นๆได้

1.3 ขอบเขตของโครงงาน

1.3.1. รถไฟขนส่งแร่ถ่านหิน

1.3.2. รางรถไฟทางเดี่ยว

1.4 วิธีการดำเนินงานวิจัย

1.4.1 ศึกษาข้อมูลเบื้องต้นเกี่ยวกับรถไฟขนส่งแร่ถ่านหิน

1.4.2 ศึกษาการออกแบบเส้นทางและการวางสถานี

1.4.3 ศึกษาและวิเคราะห์ข้อมูล จุดประสงค์ เงื่อนไขต่างๆ และความต้องการของโครงการ

1.4.4 นำข้อมูล สมการ และเงื่อนไขต่างๆมาเขียนเป็นฟังก์ชั่นในโปรแกรม Matlab

1.4.5 เรียบเรียง ทดลองประมวลผลและแสดงผล

1.4.6 ตรวจสอบและวิเคราะห์ข้อผิดพลาด

1.4.7 สรุปผลและจัดทำรายงานวิจัย

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1.5.1 ช่วยเพิ่มความรู้ความเข้าใจและแก้ปัญหาเกี่ยวการออกแบบเส้นทางและตารางเดินรถ

1.5.2 สามารถนำมาประยุกต์ใช้กับเส้นทางและตารางการเดินรถสำหรับโครงการอื่นได้

1.5.3 ได้พัฒนาทักษะการคิดวิเคราะห์ และการแก้ปัญหาอย่างเป็นระบบและมีเหตุผล

1.5.4 สามารถนำความรู้ การแก้ปัญหา และทักษะไปประยุกต์ใช้ในการทำงานได้

1.5.5 เป็นต้นแบบโปรแกรมเพื่อให้พัฒนาต่อในอนาคต

บทที่ 2

งานวิจัย และทฤษฎีที่เกี่ยวข้อง

ในการศึกษาและการพัฒนาโปรแกรมคอมพิวเตอร์สำหรับการวางตำแหน่งสถานีของรถไฟขนส่งสินค้าทางเดี่ยว จำเป็นต้องมีความรู้เบื้องต้นเกี่ยวกับระบบการเดินรถของรถไฟขนส่งสินค้าหรือแร่แบบทางเดี่ยว และการเขียนโปรแกรมในขั้นพื้นฐาน

2.1 งานวิจัยที่เกี่ยวข้อง

Owen Wong [1,2] ได้ศึกษาและพัฒนาโปรแกรมด้วย FORTRAN เพื่อใช้ในการศึกษารูปแบบการเคลื่อนที่รถไฟประเภทรถไฟทางเดี่ยว และตารางในการเดินรถ โดยมีจุดประสงค์เพื่อลดความล่าช้าของการเดินรถ

Szpigel [3] ศึกษาและอธิบายขั้นตอนวิธีการแตกกิ่งและกำหนดขอบเขตในการหาจุดตัดและการวางตำแหน่งจุดสับหลีกในระบบรถไฟทางเดี่ยว อีกทั้งยังพัฒนารูปแบบของปัญหาเกี่ยวกับตารางการเดินรถของรถไฟประเภททางเดี่ยวให้อยู่ในรูปกราฟดีสจังทีฟ (Disjunctive graph) เพื่อใช้ในการลดและแก้ปัญหาทางด้านความล่าช้าด้วยวิธีการที่แม่นยำ ผลที่ได้คือจุดตัดหรือตำแหน่งในการสับหลีกลดลงจากที่กำหนดหรือคาดการณ์ไว้ แต่ปัญหาส่วนใหญ่ที่สำคัญก็ยังคงได้ผลที่ยังไม่น่าพอใจนัก

Assad AA [4] ได้ศึกษาการวางตำแหน่งขอสิ่งก่อสร้าง รูปแบบเส้นทาง โครงสร้างและการจัดการตารางการเดินรถไฟเพื่อใช้ในการวางแผนขั้นตอนการทำงานในงานที่เกี่ยวข้องกับรถไฟ

Valentina Cacchiani [5] และคณะได้ทำการอธิบายปัญหาเกี่ยวกับการจัดการตารางการเดินรถ ปัญหาการจัดตั้งสถานี และได้ทำปรับแก้ให้เหมาะสมด้วยวิธีการ ILP (Interger Linear Programming) ในการเพิ่มประสิทธิภาพของตารางการเดินรถ

P. Tormos [6] และคณะได้ศึกษาและใช้ขั้นตอนวิธีการเชิงพันธุกรรม (Genetic Algorithm) ในการปรับแก้ให้เหมาะสมสำหรับปัญหาของ Spanish Manager of Infrastructure เกี่ยวกับการออกแบบเส้นทาง ตำแหน่งจุดสับหลีก และตารางการเดินรถไฟ

Diego Arenas [7] และคณะได้ศึกษาเกี่ยวกับวิธีการแก้ปัญหาของตารางเดินรถไฟให้มีความเหมาะสม โดยใช้วิธีการศึกษาแบบจำลองทางคณิตศาสตร์ที่เกี่ยวข้องกับการเดินรถ และ ขั้นตอนวิธีการเชิงพันธุกรรม (Genetic Algorithm) ในการปรับปรุงตารางการเดินรถ

Higgins [8] และคณะได้สร้างโปรแกรมรูปแบบทางเรขาคณิตเพื่อใช้ในการสร้างตารางการเดินรถของรถไฟทางเดี่ยว โดยศึกษาในระบบโครงข่ายของรถไฟในประเทศออสเตรเลีย และได้ทำการปรับแก้ให้เหมาะสม ด้วยวิธีการ Branch and bound

2.2 ข้อมูลเกี่ยวกับรถไฟขนแร่ถ่านหินทางเดี่ยว [9-11]

รถไฟขนส่งแร่ถ่านหินทางเดี่ยวนั้นเป็นการขนส่งแร่โดยมีรางเพียงคู่เดียวที่ใช้ในการเคลื่อนที่ทั้งไปและกลับของขบวนรถทุกขบวนและมีสถานีหรือจุดสับหลีกเพื่อหลีกทางไม่ให้เกิดอุบัติเหตุในการขนส่ง ในการเดินรถของรถไฟขนส่งแร่ถ่านหินจะมีการกำหนดความเร็ว ความเร่ง ระยะเบรก และวิธีการอื่น ๆ เพื่อให้มีความปลอดภัยในการเดินรถอีกด้วย

2.2.1 ความเร็ว

ความเร็วของรถไฟขนส่งสินค้าและรถไฟขนส่งแร่สามารถวิ่งสูงสุดได้ 80 ไมล์ต่อชั่วโมง (128-130 กิโลเมตรต่อชั่วโมง) แต่ในบางประเทศจะมีการกำหนดความเร็วอยู่ที่ 50-60 ไมล์ต่อชั่วโมง (80-96 กิโลเมตรต่อชั่วโมง)

2.2.2 ความเร่ง / ความหน่วง

ความเร่งของรถไฟขนส่งสินค้าและรถไฟขนส่งแร่จะมีค่าอยู่ที่ 0.2-0.4 $\frac{m}{s^2}$ ความหน่วงของรถไฟขนส่งสินค้าและรถไฟขนส่งแร่ที่ใช้ในการหยุดรถหรือชะลอรถจะมีค่าอยู่ที่ 0.1 $\frac{m}{s^2}$ เพื่อไม่ให้รถเกิดอุบัติเหตุเนื่องจากขบวนรถมีโมเมนตัมที่สูงมาก

2.2.3 ระยะเบรค

ระยะเบรกของรถนั้นจะขึ้นอยู่กับความเร็วที่รถกำลังเคลื่อนที่ โดยเฉลี่ยนั้น ระยะเบรกของรถไฟจะอยู่ที่ 2.1-2.5 กิโลเมตร

2.2.4 ความปลอดภัย

การเดินรถไฟของรถไฟทางเดี่ยวนั้นต้องคำนึงถึงความปลอดภัยเป็นอย่างมากเนื่องมีเส้นทางในการเดินรถแค่เส้นเดียวซึ่งเส้นทางนั้นรถต้องเดินทางไปได้ทั้ง 2 ทิศทางจึงต้องคำนึงถึง การวางจุดสับหลีกให้รถหลีกอีกขบวนที่สวนทางมา การเว้นระยะห่างของรถไฟสองขบวนที่เคลื่อนที่ในทิศทางเดียวกันเพื่อไม่ให้เกิดการชนหรืออุบัติเหตุในการเดินรถ รวมไปถึงการกำหนดความเร็วสูงสุดที่สามารถใช้ได้

2.3 สมการในการหาตำแหน่งและเวลาของรถไฟ ณ ตำแหน่งต่างๆ [12]

สมการที่ใช้ในการหาตำแหน่งและเวลาของรถไฟ ณ ตำแหน่งต่างๆ ได้กำหนดเงื่อนไขไว้โดยไม่ต้องคำนึงถึงความเร่ง การเบรค และแรงต้านต่างๆมาเกี่ยวข้องในการคำนวณจากสมการต่อไปนี้

2.3.1 สมการหาระยะเวลาการเคลื่อนที่

ในการหาตำแหน่งของรถไฟ ณ เวลาต่างๆจำเป็นต้องทราบถึงระยะเวลาในการเดินรถละช่วงของรถไฟ ซึ่งโครงงานที่ได้ทำนั้นได้กำหนดให้ความเร็วที่ใช้เป็นความเร็วเฉลี่ยหรือว่าไม่มีความเร่งระหว่างการเดินรถแต่ละช่วง เมื่อทราบถึงความเร็ว และตำแหน่งที่รถไฟจะเคลื่อนที่ ระยะเวลาการเคลื่อนที่จึงสามารถหาได้จากสมการดังต่อไปนี้

$$t = t_{arr,h} - t_{dep,h-1} = \frac{d_h - d_{h-1}}{v} \qquad (2.1)$$

โดยที่ t คือ ระยะเวลาการเดินรถ (ชั่วโมง)

$t_{arr,h}$ คือ เวลา ณ ที่รถเคลื่อนที่ถึงมาสถานี h (ชั่วโมง)

$t_{dep,h-1}$ คือ เวลา ณ ที่รถเริ่มเคลื่อนที่ออกจากสถานี h-1 (ชั่วโมง)

d_h คือ ตำแหน่งสถานี h (กิโลเมตร)

d_{h-1} คือ ตำแหน่งสถานี h-1 (กิโลเมตร)

v คือ ความเร็ว (กิโลเมตรต่อชั่วโมง)

จากสมการที่ 2.1 จะสามารถหาระยะทางที่เคลื่อนที่ได้ดังนี้

$$d = d_h - d_{h-1} = v \times t \qquad (2.2)$$

โดยที่ d คือ ระยะทางที่เคลื่อนที่ได้ (กิโลเมตร)

2.3.2. สมการหาจุดตัดของรถไฟสองขบวนที่สวนทางกัน

ในการเลือกสถานีที่ใช้ในการจอดเพื่อสับหลีกนั้น จำเป็นต้องทราบว่ารถไฟแต่ละขบวนนั้นมีเกิดการตัดกันกับรถไฟที่สวนทางมา ณ ตำแหน่งใดบ้าง ซึ่งวิธีการหาสามารถหาได้จากการนำสมการที่ 2.2 มาประยุกต์ใช้ดังนี้

จากสมการที่ 2.2 จะสามารถหาระยะทางการเคลื่อนที่ของรถเที่ยวขาไปได้ดังนี้

$$d_{i,forward} - d_f = v \times (t_f - t_{i,forward}) \quad ; d_{i,forward} > d_f \tag{2.3}$$

โดยที่ $d_h = d_f$ คือ ตำแหน่งระยะทางสุดท้าย (กิโลเมตร)

$d_{h-1} = d_i$ คือ ตำแหน่งระยะเริ่มต้น (กิโลเมตร)

$t_{arr,h} = t_f$ คือ เวลาที่รถขาไปเคลื่อนที่ถึงตำแหน่งสุดท้าย (ชั่วโมง)

$t_{dep,h-1} = t_i$ คือ เวลาที่รถขาไปเริ่มเคลื่อนที่ (ชั่วโมง)

จากสมการที่ 2.2 จะสามารถหาระยะทางการเคลื่อนที่ของรถเที่ยวขากลับได้ดังนี้

$$d_f - d_{i,back} = v \times (t_f - t_{i,back}) \quad ; d_{i,back} < d_f \tag{2.4}$$

นำสมการที่ 2.3 ลบด้วยสมการ 2.4 จะสามารถหาจุดตัดของรถไฟสองขบวนที่สวนทางกันได้ดังนี้

$$d_{conflict} = ((d_{i,forward} + d_{i,back}) v \times (t_{i,forward} - t_{i,back}))/2 \tag{2.5}$$

โดยที่ $d_f = d_{conflict}$ คือ ตำแหน่งของจุดที่รถไฟสองขบวนที่สวนทางตัดกัน (กิโลเมตร)

2.4 สมการที่ใช้สำหรับเขียนฟังก์ชั่น

สมการที่ใช้สำหรับเขียนฟังก์ชั่นคือสมการที่ได้มจากหัวข้อ 2.3 นำมาเขียนอยู่ในรูปอย่างง่ายเพื่อให้ใช้และศึกษาได้สะดวกขึ้น สามารถเขียนได้ดังนี้ได้ดังนี้

1. สมการหาระยะเวลาการเดินรถจากจุดเริ่มต้นไปจุดสุดท้าย และ สมการหาระยะทางที่รถเคลื่อนที่จากสมการที่ 2.1 และ 2.2 นำมาเขียนและจัดรูปตัวแปรใหม่ จึงได้สมการดังนี้

$$t=(S-S_o)/(d \times v) \quad (2.6)$$

$$S=S_o+(v \times t) \quad (2.7)$$

S = ตำแหน่งระยะทางสุดท้าย (กิโลเมตร)

S_o = ตำแหน่งระยะทางเริ่มต้น (กิโลเมตร)

t = ระยะเวลาการเดินรถจากจุดเริ่มต้นไปจุดสุดท้าย (ชั่วโมง)

v = ความเร็ว (กิโลเมตรต่อชั่วโมง)

d = สัมประสิทธิ์ที่เป็นเงื่อนไขเพื่อไม่ให้ระยะเวลามีค่าติดลบ

2. สมการจุดตัดของรถไฟสองขบวนที่วิ่งสวนทางกัน
จากสมการที่ 2.5 มาเขียนและจัดรูปตัวแปรใหม่ จึงได้สมการดังนี้

$$S = ((S_{o(unloaded)} + S_{o(loaded)}) + (v \times (t_{o(loaded)} - t_{o(unloaded)})))/2 \quad (2.8)$$

S = ตำแหน่งของจุดที่รถสองขบวนตัดกัน (กิโลเมตร)

$S_{o(unloaded)}$ = ตำแหน่งที่รถขากลับเริ่มออกเคลื่อนที่ (กิโลเมตร)

$S_{o(loaded)}$ = ตำแหน่งที่รถขาไปเริ่มออกเคลื่อนที่ (กิโลเมตร)

velocity = ความเร็ว (กิโลเมตรต่อชั่วโมง)

$t_{o(loaded)}$ = เวลาที่รถขาไปเริ่มเคลื่อนที่ (ชั่วโมง)

$t_{o(unloaded)}$ = เวลาที่รถขากลับเริ่มเคลื่อนที่ (ชั่วโมง)

3. สมการหาเวลา ณ ตำแหน่งที่รถสองขบวนที่สวนทางกันเคลื่อนที่มาตัดกัน

ในการหาเวลา ณ ตำแหน่งที่รถตัดกัน สามารถเขียนได้โดย แทนสมการที่ 2.7 ในสมการที่ 2.8 จะได้

$$S_o + (v \times t) = ((S_{o(unloaded)} + S_{o(loaded)}) + (v \times (t_{o(loaded)} - t_{o(unloaded)})))/2 \quad (2.9)$$

กำหนดให้ S_o มีค่าเท่ากับ 0 และ $t_{o(loaded)}$ คือ เวลาของรถเที่ยวขาไปเริ่มเคลื่อนที่ออกจากสถานีซึ่งหาค่าได้จาก $t_{o(loaded)} = (N-1) \times t_{(loading)}$ N คือลำดับของขบวนรถเที่ยวขาไป จากนั้นแทนค่าตามที่กำหนดในสมการที่ 2.9 จะสามารถหาสมการเวลา ณ ตำแหน่งที่รถสองขบวนที่สวนทางกันเคลื่อนที่มาตัดกันได้ดังนี้

$$t = ((S_{o(loaded)} - S_{o(unloaded)}) + (v \times ((N-1) \times t_{(loading)} - t_{o(unloaded)})))/(2 \times v) \quad (2.10)$$

t = เวลาที่รถสองขบวนตัดกันที่จุดใดจุดหนึ่ง (ชั่วโมง)

$S_{o(unloaded)}$ = ตำแหน่งที่รถขากลับเริ่มออกเคลื่อนที่ (กิโลเมตร)

$S_{o(loaded)}$ = ตำแหน่งที่รถขาไปเริ่มออกเคลื่อนที่ (กิโลเมตร)

v = ความเร็ว (กิโลเมตรต่อชั่วโมง)

N = ลำดับของขบวนรถ

$t_{(loading)}$ = ระยะเวลาการบรรทุกสินค้า หรือ Loading Time (ชั่วโมง)

$t_{o(unloaded)}$ =เวลาที่รถขากลับเริ่มเคลื่อนที่ (ชั่วโมง)

2.5 การประมวลผลข้อมูลการทดลอง

การคำนวณค่าความแตกต่างงจากการผลลัพธ์ที่ได้สามารถหาได้ดังสมการที่ 2.11

$$\frac{|\text{ค่าทฤษฎี-ค่าจากการทดลอง}|}{\text{ค่าทฤษฎี}} \times 100 \quad (2.11)$$

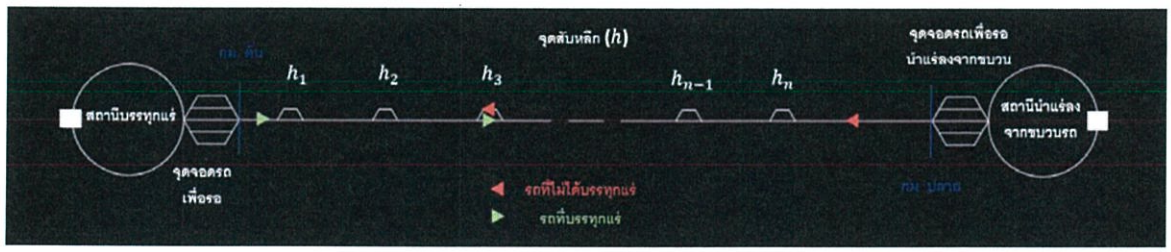
บทที่ 3

วิธีดำเนินการวิจัย

ขั้นตอนการดำเนินงานจะเริ่มตั้งแต่การศึกษาข้อมูลเกี่ยวกับรถไฟขนส่งเบื้องต้นและวิเคราะห์ปัจจัยกับองค์ประกอบของตัวแปรหรือสิ่งที่เกี่ยวข้องในการออกแบบเส้นทางและตารางการเดินรถ เมื่อศึกษาเสร็จแล้วจึงเริ่มเก็บข้อมูล กำหนดตัวแปรและนำมาประมวลผลเพื่อหาผลสรุป

3.1 ลักษณะการขนส่งรถไฟขนส่งแร่ถ่านหิน

เริ่มต้นรถแต่ละขบวนจะเริ่มจากจุดพักรถเพื่อรอการขนแร่ถ่านหินขึ้นขบวนรถ จุดพักรถมีลักษณะเป็นสถานีที่สามารถนำรถเข้าได้หลายขบวน เมื่อรถขบวนแรกทำการบรรทุกแร่ถ่านหินที่สถานีบรรทุกแร่ถ่านหินเสร็จ รถที่จอดรออยู่ที่จุดพักรถจะเคลื่อนที่เข้าสถานีบรรทุกแร่ถ่านหินได้ก็ต่อเมื่อขบวนก่อนหน้าทำการบรรทุกแร่และเคลื่อนที่ออกจากจุดบรรทุกแร่ถ่านหินเสียก่อนรถขบวนที่บรรทุกถ่านหินจะมีความสำคัญมากกว่ารถที่ไม่บรรทุกถ่านหิน โดยรถที่บรรทุกถ่านหินจะเคลื่อนที่โดยไม่จอดระหว่างทาง(ถ้าไม่จำเป็น) จนถึงจุดพักรถก่อนจะขนแร่ถ่านหินออกจากขบวน และให้รถไม่บรรทุกแร่ถ่านหินที่เคลื่อนที่สวนทางมาหลีกทาง ให้กับขบวนที่บรรทุกถ่านหิน ตามจุดสับหลีกต่างๆ เมื่อรถที่บรรทุกแร่ถ่านหินมาถึงจุดพักรถก่อนจะเข้าจุดขนแร่ถ่านหินออกจากขบวน ต้องตรวจสอบก่อนว่าขบวนก่อนหน้าขนแร่ถ่านหินเสร็จแล้ว จึงค่อยเคลื่อนที่เข้าเพื่อเข้าขนแร่ลง ดังรูปที่ 3.1



รูปที่ 3.1 รูปแบบการเดินรถและจุดสับหลีก

3.2 วิเคราะห์เกี่ยวกับปัญหาที่ต้องการแก้ไขและปัจจัยที่เกี่ยวข้อง

ระบบการเดินรถไฟขนส่งสินค้าแบบทางเดี่ยวนั้นปัญหาที่สนใจในการปรับปรุงเพื่อเพิ่มประสิทธิภาพ คือระยะเวลาในการขนต่อรอบ อีกทั้งยังคต้องคำนึงถึงความปลอดภัย เนื่องจากรางรถไฟนั้นจะต้องใช้วิ่งทั้งขบวนเที่ยวไปและเที่ยวกลับ จึงจำเป็นที่จะต้องมีสถานีเพื่อสับหลีกให้ไม่เกิดการชนกัน จำนวนและตำแหน่งของสถานีนั้น ๆ จึงมีผลต่อระยะเวลาในการขนต่อรอบเป็นอย่างมาก ดังนั้นปัญหาที่เราสนใจในการศึกษานี้ คือ

1.ระยะเวลาการเดินรถต่อรอบ (Cycle time)

ระยะเวลาในการเดินรถต่อรอบ คือระยะเวลาของรถหนึ่งขบวนที่ใช้ในการขนส่งแร่ถ่านหิน 1 รอบซึ่งแยกได้แบบช่วงดังนี้

1.1 ระยะเวลาในการเดินรถขณะบรรทุกแร่/ไม่บรรทุกแร่ (Running Time of Loaded/Unloaded Train) คือ ระยะเวลาที่รถเคลื่อนที่จากสถานีเริ่มต้นไปยังสถานีสุดท้าย (รถที่ไม่ได้บรรทุกแร่ (Unloaded Train) รถจะเคลื่อนที่จากสถานีสุดท้ายไปยังสถานีเริ่มต้น) ซึ่งหาได้จาก

$$RT = S/V \tag{3.1}$$

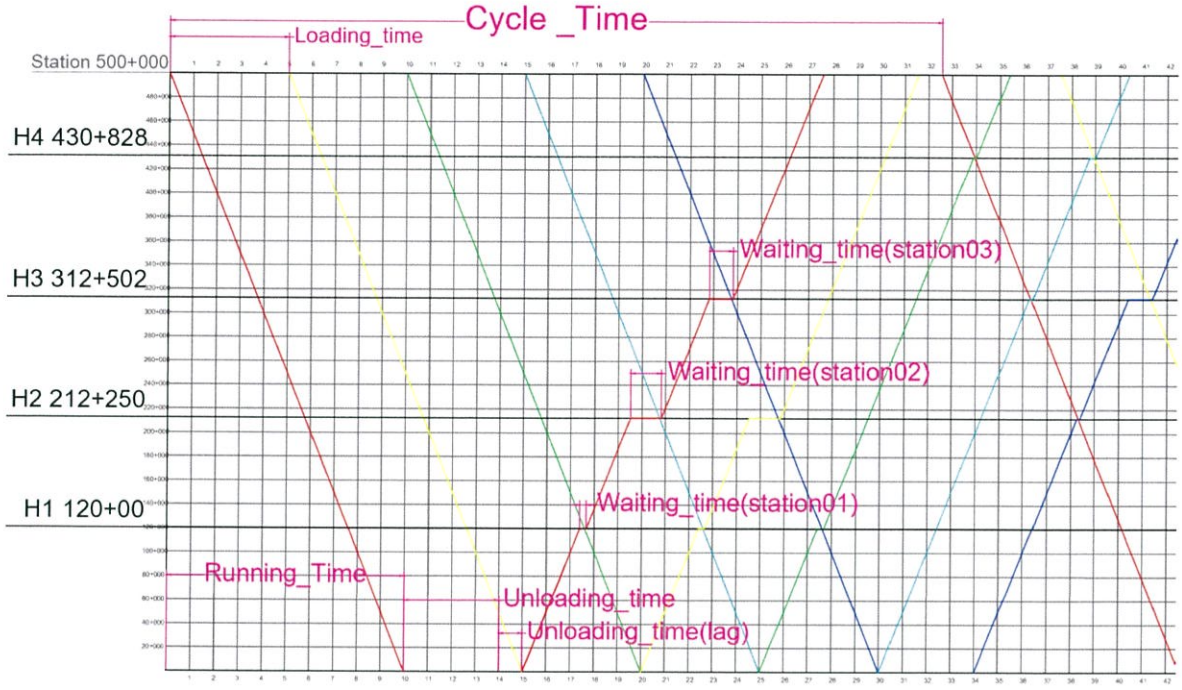
S = ระยะทางจากจากสถานีเริ่มต้นไปยังสถานีสุดท้าย (กิโลเมตร)

V = ความเร็ว (กิโลเมตรต่อชั่วโมง)

RT = ระยะเวลาในการเดินรถขณะบรรทุกแร่/ไม่บรรทุกแร่ (ชั่วโมง)

1.2 ระยะเวลาในการบรรทุกแร่ และ ระยะเวลาในการนำแร่ลง (Loading Time /Unloading Time) คือระยะเวลาในการบรรทุกแร่ขึ้นขบวนรถและระยะเวลาในการขนถ่ายแร่ลงจากขบวนรถ ซึ่งระยะเวลาที่กล่าวมานั้นจะขึ้นอยู่กับปริมาณแร่ถ่านหินที่รถไฟแต่ละขบวนบรรทุก

1.3 ระยะเวลาในการจอดรอ (Waiting Time) คือระยะเวลาหยุดรอหลีกให้รถไฟอีกขบวนที่สวนทางมาเคลื่อนที่ผ่านไปก่อน และ รวมไปถึงการจอดรอให้รถอีกขบวนบรรทุกแร่หรือนำแร่ลงให้เสร็จก่อน



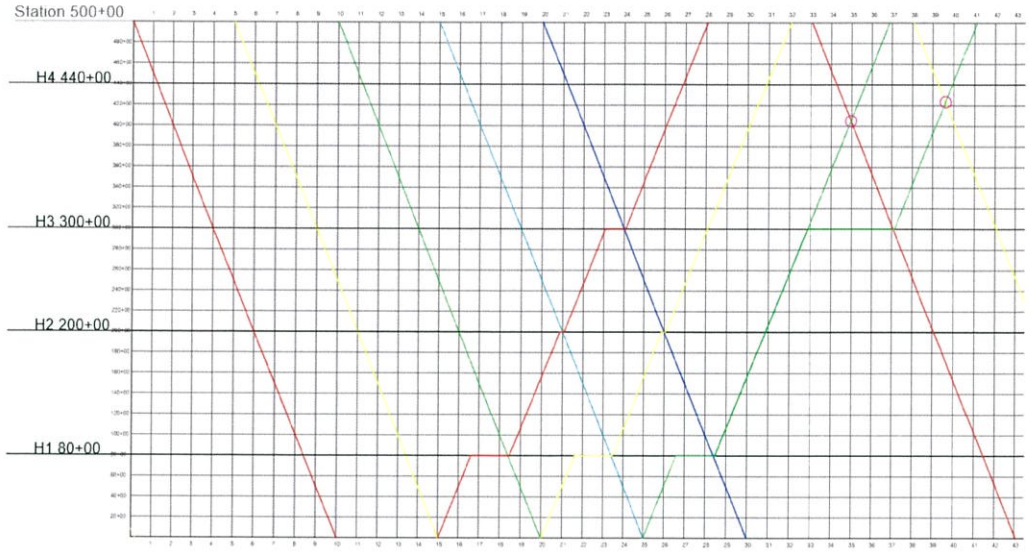
รูปที่ 3.2 องค์ประกอบของระยะเวลาการเดินรถต่อรอบ

2.จำนวนและตำแหน่งจุดสับหลีก

ตำแหน่งจุดสับหลีกและจำนวนจุดสับหลีก ส่งผลต่อเวลาการจอดรอหลีกรถไฟอีกขบวนที่สวนทางมา อีกทั้งถ้ายิ่งมีจำนวนจุดสับหลีกมาก ระยะเวลาในการจอดรอก็จะยิ่งมากขึ้นหรือลดลงตามไปด้วยเช่นกัน

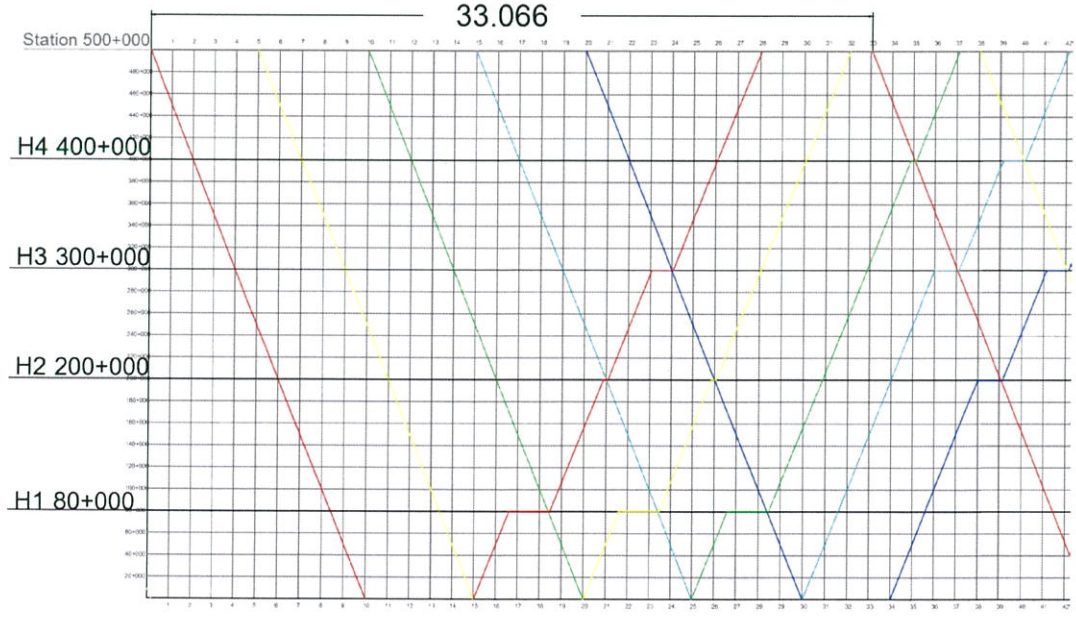
ตัวอย่างที่แสดงให้เห็นเกิดจากการใช้วิธีการลองผิดลองถูก (Trial and Error) ในการวางจุดสับหลีกที่กิโลเมตรที่ต่างกัน และคำนวณหาระยะเวลาการเดินรถต่อรอบด้วยวิธีการวาดกราฟในโปรแกรม AutoCAD เพื่อแสดงให้เห็นถึงความแตกต่างและการเปลี่ยนแปลงของระยะเวลาการเดินรถต่อรอบ

ในรูปที่ 3.3 เป็นการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 440+000 จะเห็นได้ว่าเส้นของรถขบวนที่ 3 (สีเขียว) เที่ยวขากลับ เกิดการชนกัน ไม่ว่าจะสับหลีกที่จุดไหนก็ตาม ถือว่าเป็นการวางตำแหน่งที่ผิดพลาด



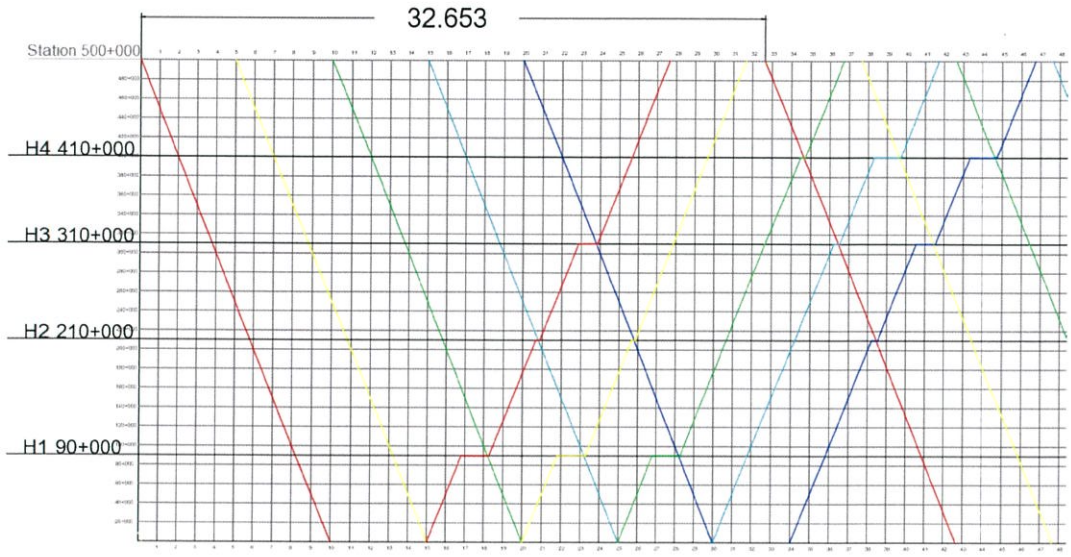
รูปที่ 3.3 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 440+000

ในรูปที่ 3.4 เป็นการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 400+000 จะเห็นได้ว่าทุกขบวนสามารถเดินรถได้อย่างปกติโดยมีช่วงเวลาระยะเวลาการเดินรถต่อรอบเท่ากับ 33.066 ชั่วโมง



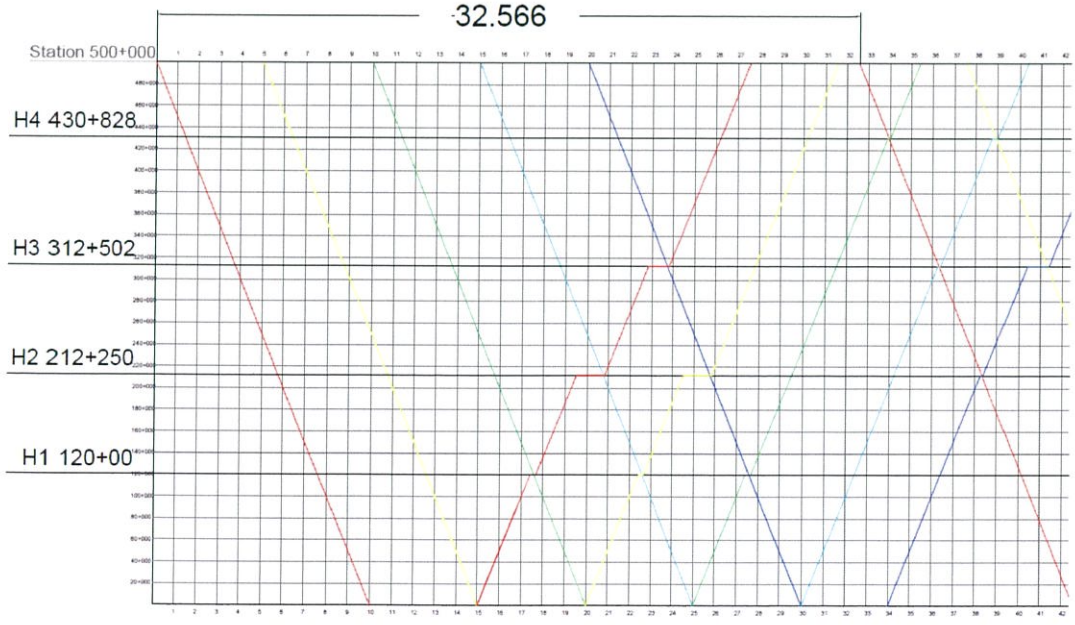
รูปที่ 3.4 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000, 200+000, 300+000, 400+000

ในรูปที่ 3.5 เป็นการวางจุดสับหลีกที่ กม.ที่ 90+000, 210+000, 310+000, 410+000 จะเห็นได้ว่ารถทุกขบวนสามารถเดินรถได้ปกติโดยมีช่วงเวลาระยะเวลาการเดินรถต่อรอบเท่ากับ 32.653 ชั่วโมง



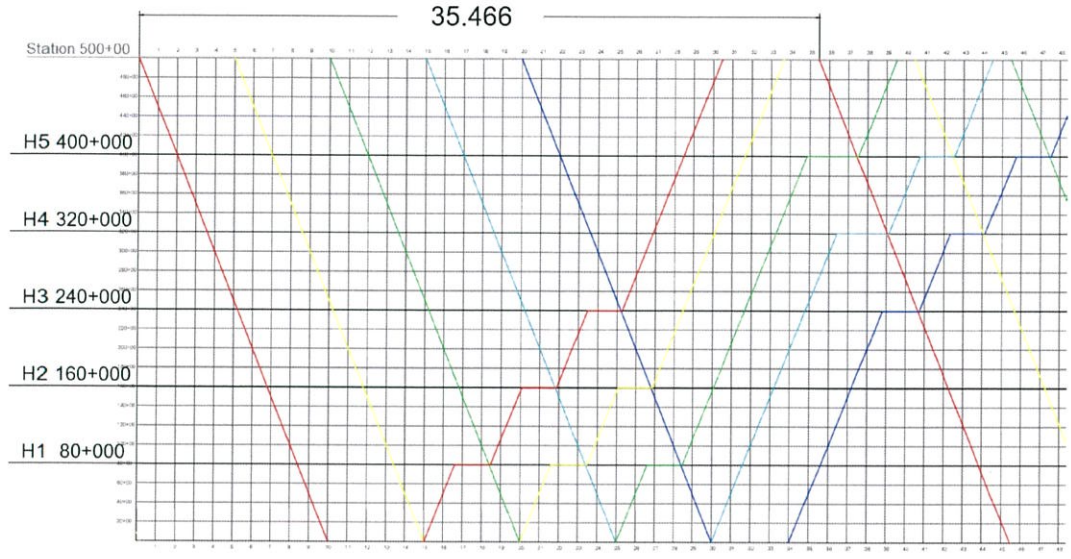
รูปที่ 3.5 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 90+000, 210+000, 310+000, 410+000

ในรูปที่ 3.6 เป็นการวางจุดสับหลีกที่ละเอียดขึ้น โดยวางที่กม.ที่ 120+000, 212+250+000, 312+502, 430+828 จะเห็นได้ระยะเวลาการเดินรถต่อรอบนั้นลดลงเมื่อเทียบกับรูปที่ 3.5 จาก 32.653 ชั่วโมง เป็น 32.566 ชั่วโมง ซึ่งลดลงไปประมาณ 5 นาที



รูปที่ 3.6 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 120+000, 212+250+000, 312+502, 430+828

ในรูปที่ 3.7 เป็นการวางจุดสับหลีกจำนวน 5 จุด โดยวางที่กม.ที่ 80+000,160+000, 240+000, 320+000, 400+000 จะเห็นได้ระยะเวลาการเดินรถต่อรอบนั้น จะเท่ากับ 35.466 ชั่วโมง เมื่อเทียบกับตัวอย่างที่ผ่านมาก็จะเห็นได้ว่าระยะเวลาในการเดินรถต่อรอบในกรณีนี้เพิ่มมากขึ้น



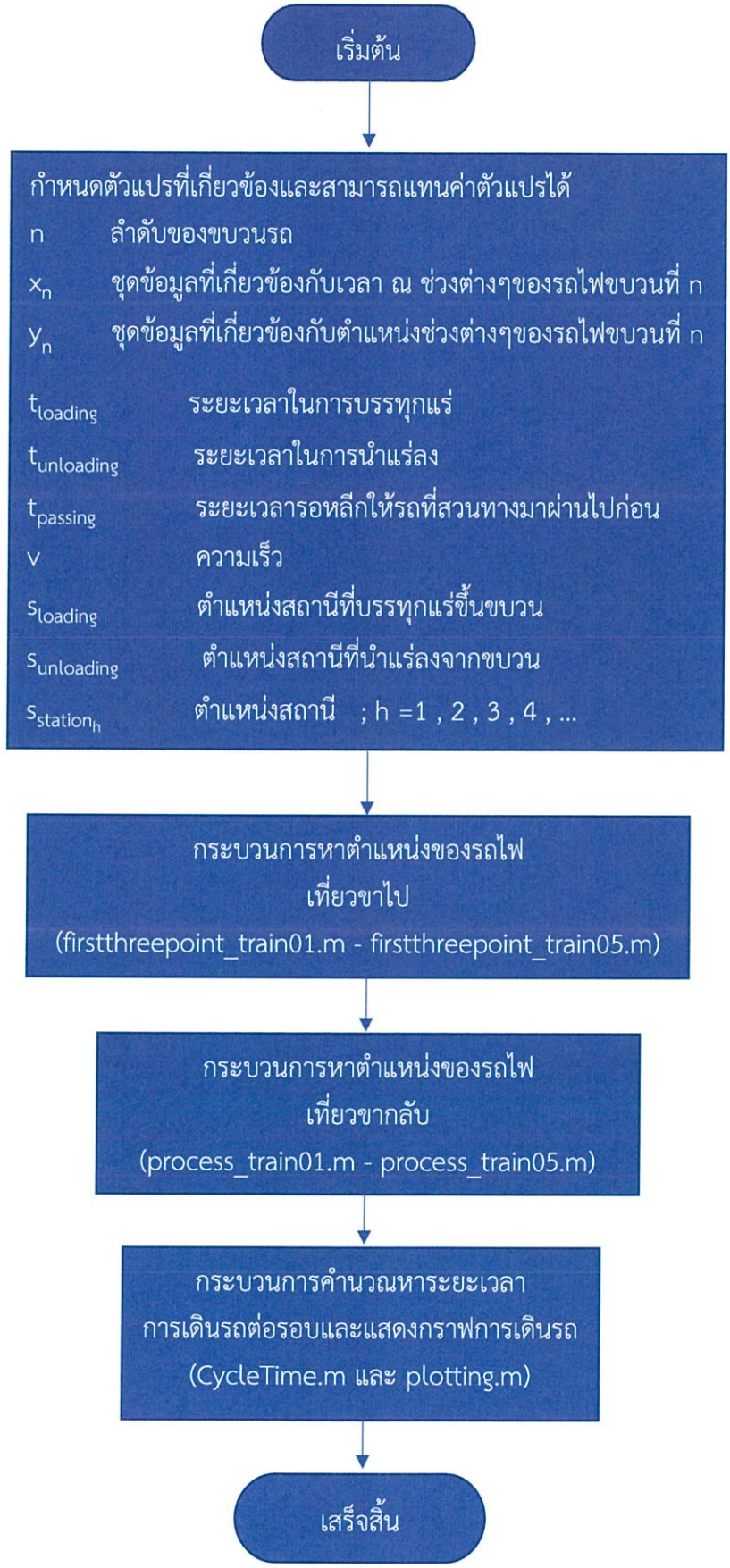
รูปที่ 3.7 กราฟแสดงการเดินรถจากการวางจุดสับหลีกที่ กม.ที่ 80+000,160+000, 240+000, 320+000, 400+000

3.3 การออกแบบและพัฒนาโปรแกรมด้วย Matlab

การออกแบบโปรแกรมที่ต้องการนั้นจะแบ่งออกเป็นการเขียนแผนภาพแสดงลำดับขั้นตอนการทำงาน (Flowchart) ,การสร้างฟังก์ชั่นและกำหนดเงื่อนไขต่างๆ โดยใช้เงื่อนไขที่ได้มากจากการศึกษาข้อมูลข้างต้น เพื่อให้โปรแกรมที่เขียนขึ้นนั้นใกล้เคียงความเป็นจริง อีกทั้งมีความถูกต้องในการคำนวณและประมวลผลมากที่สุด

แผนภาพแสดงลำดับขั้นตอนการทำงาน (Flowchart) เป็นการวางแผนแนวคิดและลำดับการทำงานของโปรแกรมเพื่อให้เข้าใจง่ายต่อการศึกษาและแก้ไข โดยภาพรวมลำดับการทำงานของโปรแกรมจะมี 4 ขั้นตอนดังต่อไปนี้

1. กำหนดตัวแปรที่เกี่ยวข้องและสามารถแทนค่าตัวแปรได้
2. กระบวนการหาตำแหน่งของรถไฟเที่ยวขาไป
3. กระบวนการหาตำแหน่งของรถไฟเที่ยวขากลับ
4. กระบวนการคำนวณหาระยะเวลาการเดินรถต่อรอบ
5. แสดงผลกราฟการเดินรถ



รูปที่ 3.8 แผนภาพแสดงลำดับขั้นตอนการทำงานของโปรแกรม

3.3.1 สร้างตัวแปรที่เกี่ยวข้องและสามารถแทนค่าตัวแปรได้

ขั้นตอนนี้เป็นการกำหนดตัวแปรที่เกี่ยวข้องกับการคำนวณหาตำแหน่งต่างๆของรถไฟ ซึ่งตัวแปรเหล่านี้จะเป็นตัวแปรที่สามารถปรับเปลี่ยนค่าที่จะแทนลงไปได้ เพื่อให้สามารถตรวจสอบได้ในหลายๆกรณี โดยตัวแปรที่ใช้ในกรณีศึกษานี้จะประกอบไปด้วย

n ลำดับของขบวนรถ

x_n ชุดข้อมูลที่เกี่ยวข้องกับเวลา ณ ช่วงต่างๆของรถไฟขบวนที่ n

y_n ชุดข้อมูลที่เกี่ยวข้องกับตำแหน่งช่วงต่างๆของรถไฟขบวนที่ n

$t_{loading}$ ระยะเวลาในการบรรทุกแร่

$t_{unloading}$ ระยะเวลาในการนำแร่ลง

$t_{passing}$ ระยะเวลารอหลีกให้รถที่สวนทางมาผ่านไปก่อน

v ความเร็ว

$s_{loading}$ ตำแหน่งสถานีที่บรรทุกแร่ขึ้นขบวน

$s_{unloading}$ ตำแหน่งสถานีที่นำแร่ลงจากขบวน

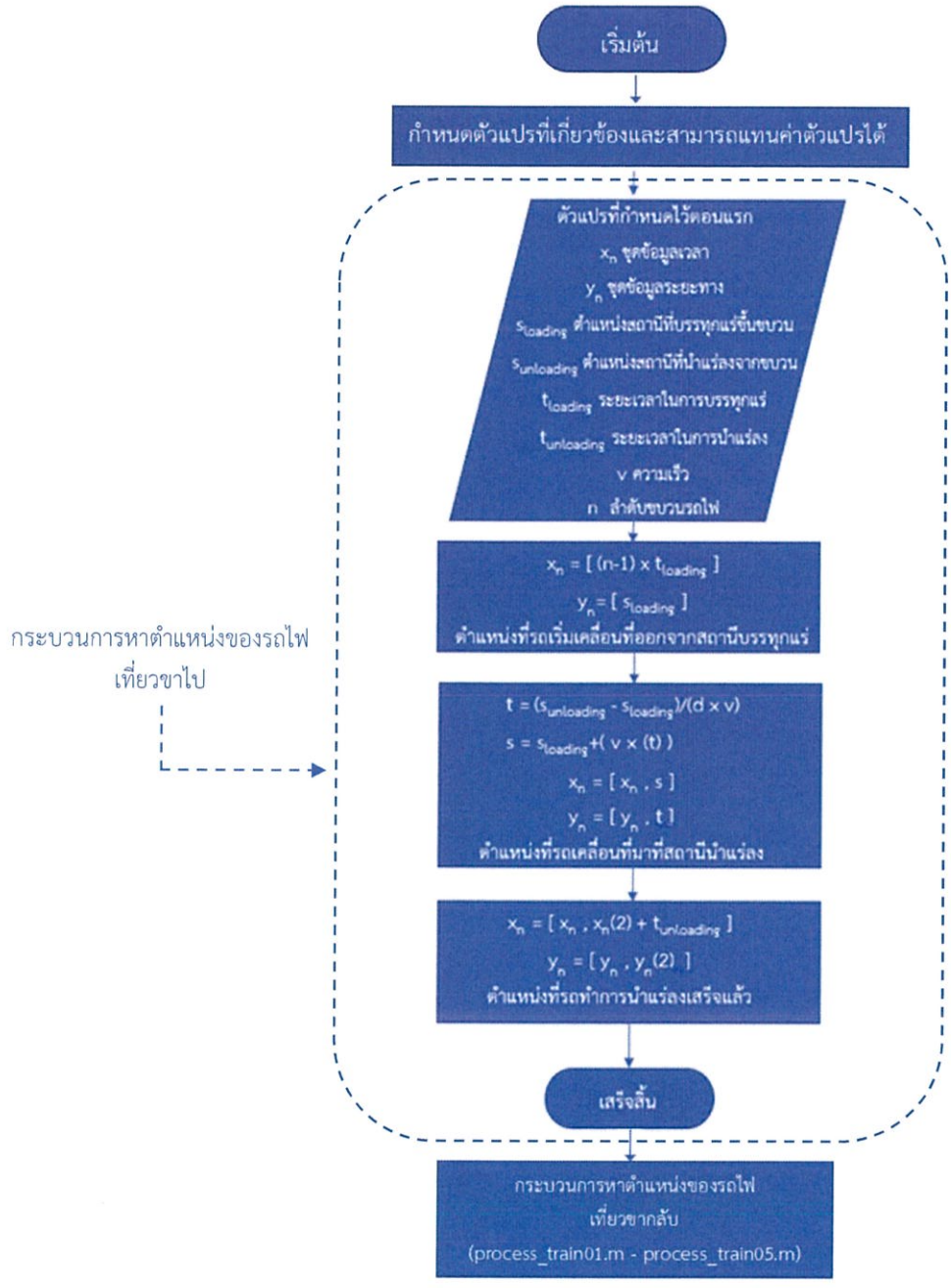
$s_{station_h}$ ตำแหน่งสถานี h ; h =1 , 2 , 3 , 4 , ...

3.3.2 กระบวนการหาตำแหน่งของรถไฟเที่ยวขาไป

ขั้นตอนในการหาตำแหน่งของรถไฟเที่ยวขาไป เนื่องจากรถไฟเที่ยวขาไปเป็นรถไฟที่กำลังบรรทุกแร่อยู่จึงทำให้มีเงื่อนไขในการเคลื่อนที่คือ ช่วงที่รถเคลื่อนที่จากสถานีบรรทุกแร่ไปยังจุดนำแร่ลงจากขบวนจะเป็นการเคลื่อนที่อย่างต่อเนื่องโดยไม่มีการหยุดเพื่อหลีกให้อีกขบวน จึงทำให้สามารถแบ่งตำแหน่งของรถในช่วงนี้ได้ออกเป็น 3 จุด คือ

1. จุดที่รถเริ่มเคลื่อนที่
2. จุดที่รถเคลื่อนที่มาถึงจุดนำแร่ลงจากขบวน
3. จุดที่รถทำการนำแร่ลงจากขบวนเสร็จเรียบร้อยแล้ว

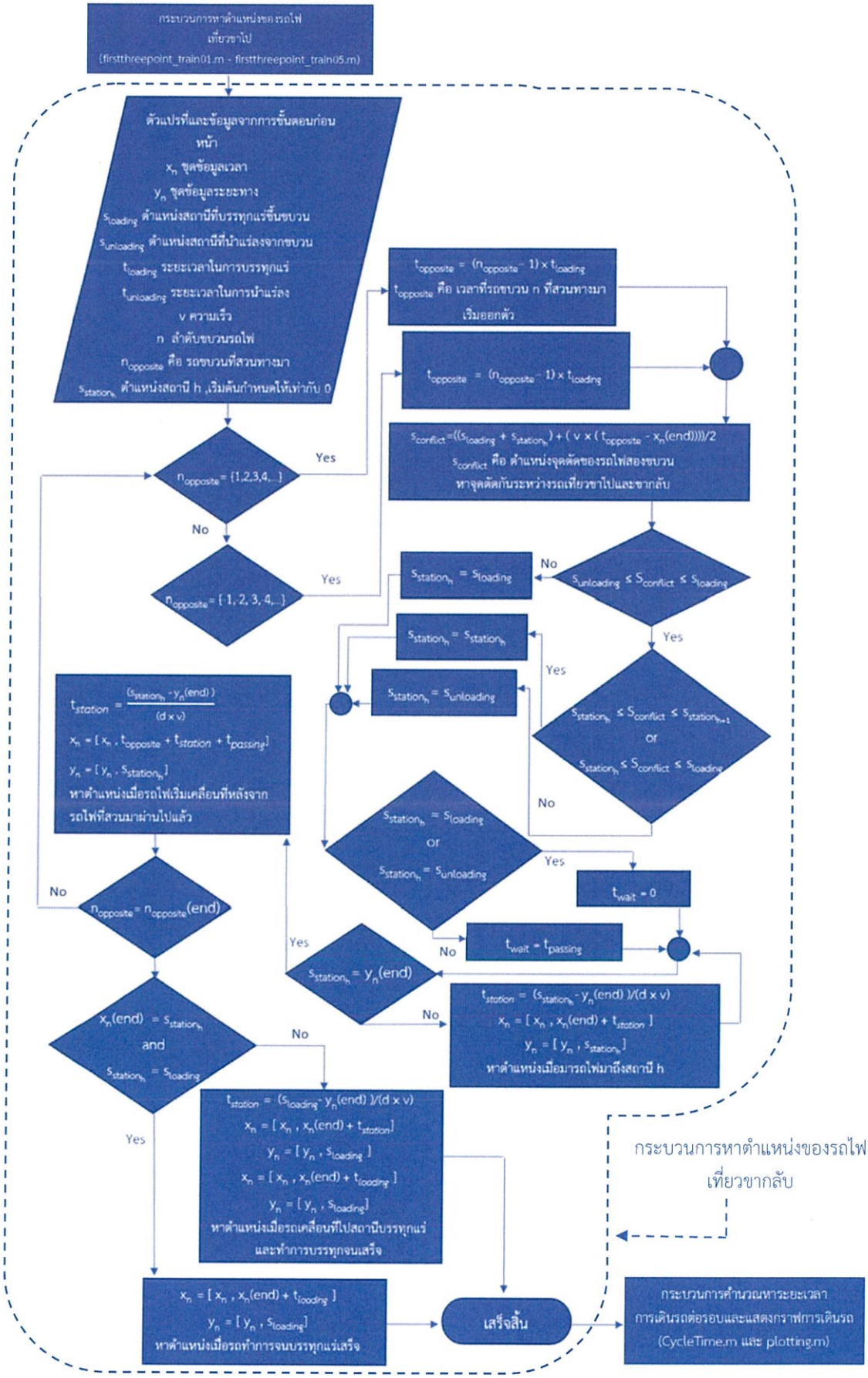
ซึ่งสามารถเขียนแผนภาพแสดงลำดับขั้นตอนการทำงานในขั้นตอนนี้ได้ดังรูปต่อไปนี้



รูป 3.9 แผนภาพแสดงลำดับขั้นตอนการหาตำแหน่งของรถเที่ยวขาไป

3.3.3 กระบวนการหาตำแหน่งของรถไฟเที่ยวขากลับ

ขั้นตอนในการหาตำแหน่งของรถไฟเที่ยวขากลับ ช่วงที่รถเคลื่อนที่จากสถานีที่นำแร่ลงจากขบวนไปยังจุดบรรทุกแร่ขึ้นขบวนจะเป็นการเคลื่อนที่โดยรถที่ไม่ได้บรรทุกแร่จะต้องคอยหลีกในสถานีต่างๆ เพื่อให้รถที่บรรทุกแร่มาผ่านไปก่อนจึงค่อยเคลื่อนที่อีกครั้งจนไปถึงจุดหมายปลายทาง ดังนั้นในการหาตำแหน่งจะซับซ้อนมากขึ้นและมีเงื่อนไขเข้ามาเกี่ยวข้องด้วย ซึ่งสามารถเขียนแผนภาพแสดงลำดับขั้นตอนการทำงานในขั้นตอนนี้ได้ดังรูปต่อไปนี้



รูป 3.10 แผนภาพแสดงลำดับขั้นตอนการหาตำแหน่งของรถเที่ยวขากลับ

3.3.4. กระบวนการคำนวณหาระยะเวลาการเดินรถต่อรอบ

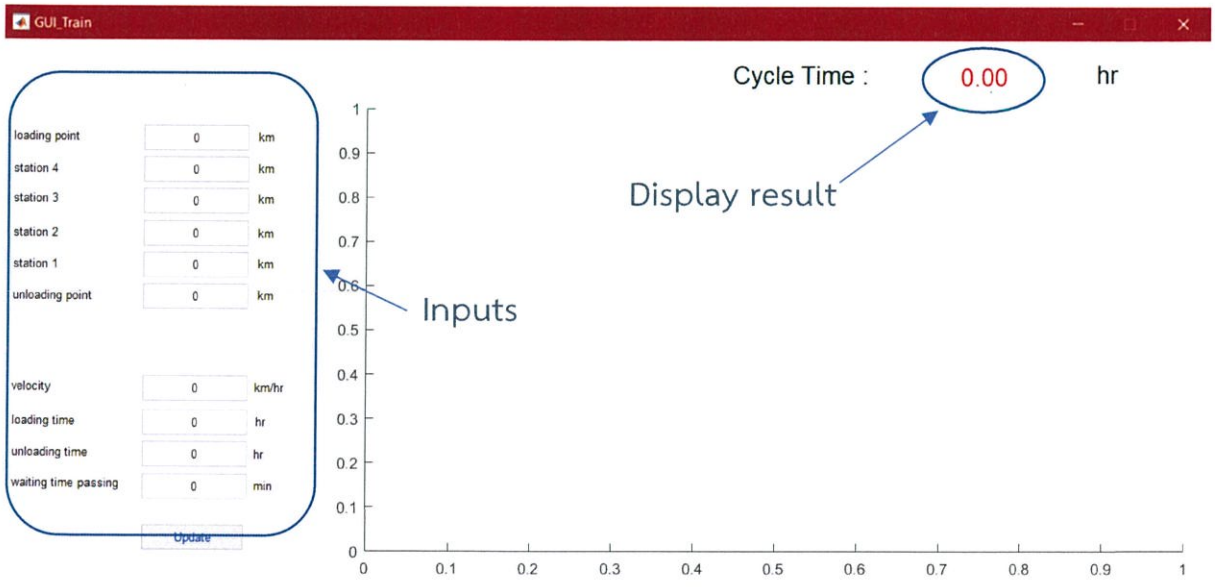
ขั้นตอนกระบวนการคำนวณหาระยะเวลาการเดินรถต่อรอบจากโปรแกรมนั้นจะเป็นการนำข้อมูล x_n ตัวสุดท้ายจากกระบวนการหาตำแหน่งของรถไฟเที่ยวขากลับมาลบกับ x_n ตัวแรก จะทำให้สามารถหาระยะเวลาในการเดินรถต่อรอบของแต่ละขบวนได้ จากนั้นก็จะทำการพลอตกราฟระหว่างตำแหน่งระยะทางกับเวลา (x_n, y_n) ของแต่ละขบวนและแสดงผลออกมาได้เป็นกราฟแสดงการเดินรถ

3.4 การใช้งานโปรแกรมและตัวอย่างการใช้งาน

โปรแกรมจะทำได้โดยเริ่มจากการใส่ค่าตัวแปรที่ได้กำหนดไว้ตามที่ต้องการซึ่งได้แก่

1. loading point คือ ตำแหน่งของจุดบรรทุกแร่หรือสินค้า (กิโลเมตร)
2. station4 คือ ตำแหน่งของสถานที่หรือจุดสับหลีกที่ 4 (กิโลเมตร)
3. station3 คือ ตำแหน่งของสถานที่หรือจุดสับหลีกที่ 3 (กิโลเมตร)
4. station2 คือ ตำแหน่งของสถานที่หรือจุดสับหลีกที่ 2 (กิโลเมตร)
5. station1 คือ ตำแหน่งของสถานที่หรือจุดสับหลีกที่ 1 (กิโลเมตร)
6. unloading point คือ ตำแหน่งของจุดที่นำแร่หรือสินค้าลง (กิโลเมตร)
7. velocity คือ ความเร็วเฉลี่ยของรถไฟ (กิโลเมตรต่อชั่วโมง)
8. loading time คือ ระยะเวลาในการบรรทุกแร่หรือสินค้า (ชั่วโมง)
9. unloading time คือ ระยะเวลาในการนำแร่หรือสินค้าลง (ชั่วโมง)
10. waiting time passing คือ ระยะเวลาจอดรอเพื่อให้รถไฟที่สวนทางมาผ่านไปก่อน (นาที)

หลังจากทำการแทนค่าตัวแปรตามที่ต้องการแล้ว ทำการคลิกที่ปุ่ม Update เพื่อประมวลผล โปรแกรมจะทำการคำนวณหาระยะเวลาในการขนส่งต่อรอบและแสดงกราฟแสดงการเดินรถของรถไฟขบวนต่างๆ



รูป 3.11 หน้าต่างอินเตอร์เฟสของโปรแกรม

เมื่อลองแทนค่าตัวแปร ทั้งหมดตามที่ต้องการและทำการประมวลผล ทุกฟังก์ชั่นจะทำงานตามลำดับขั้นตอนที่เรียบเรียงไว้จนขั้นตอนสุดท้ายจะเป็นการแสดงผลและกราฟของกราฟการเดินรถ ดังรูปที่ 3.12 เป็นแสดงถึงตัวอย่าเมื่อแทนค่าตัวแปรดังนี้

loading time = 5

unloading time = 4

waiting time passing = 4

velocity = 50

unloading_P = 0

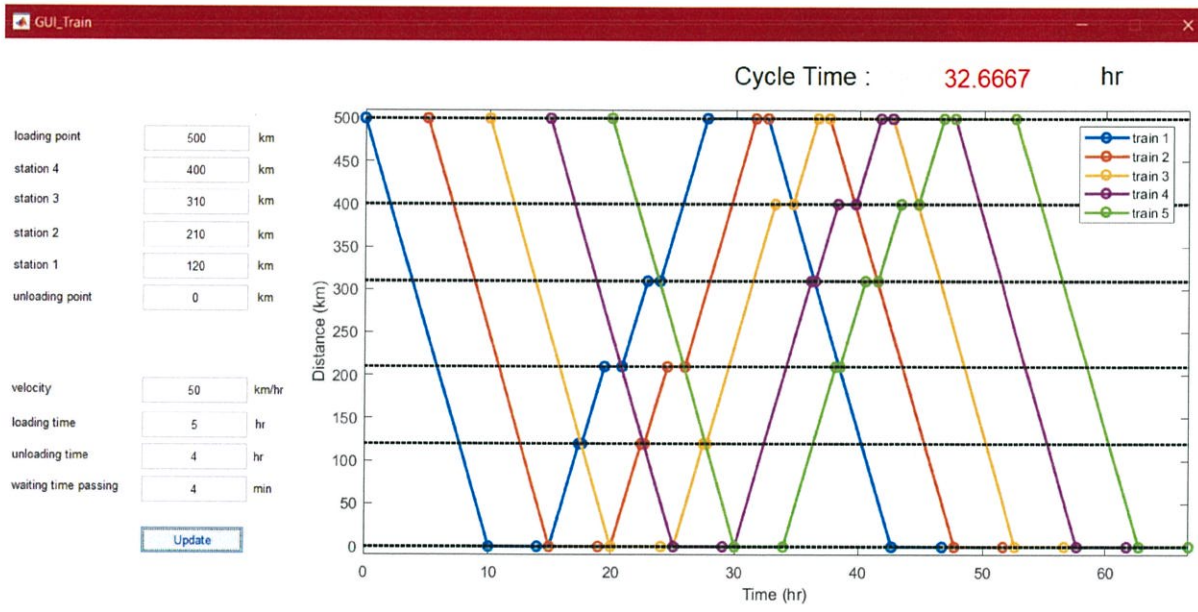
station1 = 120

station2 = 210

station3 = 310

station4 = 400

loading_P = 500



รูปที่ 3.12 หน้าอินเตอร์เฟสแสดงผลจากการทดลองคำนวณหาระยะเวลาการเดินรถต่อรอบ และกราฟแสดงการเดินรถ

บทที่ 4

ผลลัพธ์ของโครงงาน

4.1 บทนำ

หลังจากที่ทราบขั้นตอน วิธีการใช้และการทำงานโปรแกรม ในบทนี้จะทราบถึงการหาผลของการใช้โปรแกรมด้วยการตรวจสอบการวางสถานีเพื่อใช้เป็นจุดสับหลีกในตำแหน่งต่างๆ เพื่อให้ทราบถึงระยะเวลาของระยะเวลาการเดินรถต่อรอบที่ดีที่สุด

4.2 ผลลัพธ์ของโครงงาน

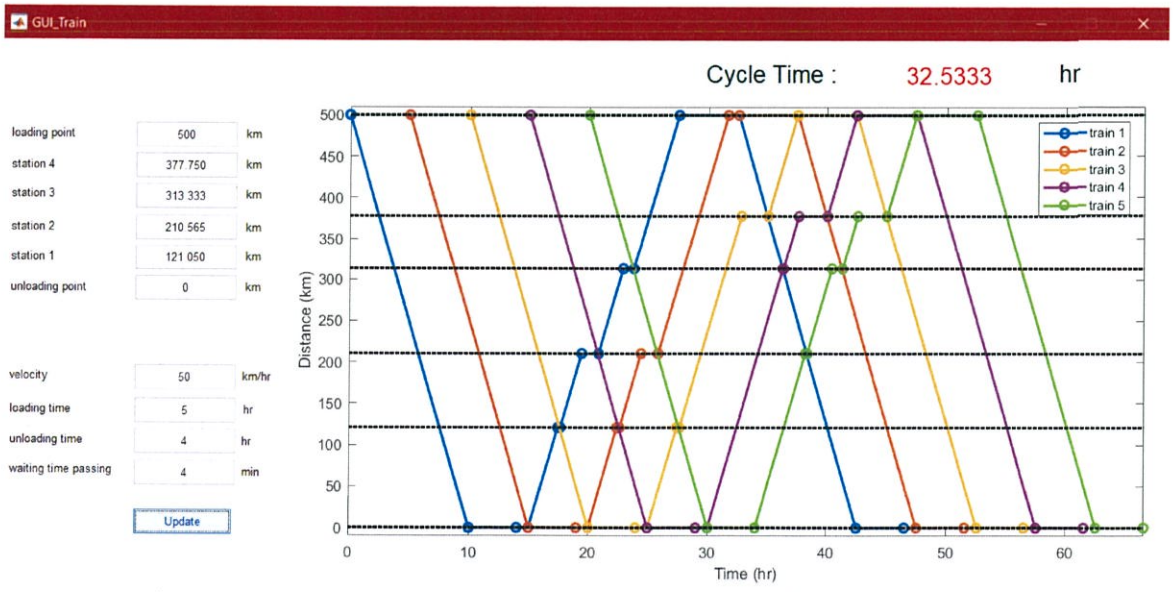
จากการทดลองแทนค่าจุดสับหลีกและสังเกตความเปลี่ยนแปลง และความเป็นไปได้ในการเดินรถ จากการคำนวณและการแสดงผลเป็นกราฟของโปรแกรมพบว่าระยะเวลาการเดินรถต่อรอบที่น้อยที่สุดที่ทำได้คือ 32.53 ชั่วโมง (32 ชั่วโมง 32 นาที) เป็นระยะเวลาที่ใช้ในการขนส่งน้อยที่สุด และ เป็นไปได้มากที่สุดสำหรับโครงการนี้ อีกทั้งตำแหน่งของการวางสถานีนั้นมีช่วงในการขยับตำแหน่งได้มากที่สุดซึ่งสามารถแสดงได้ดังนี้

ตำแหน่งของสถานีที่ 1 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [113+334 , 125+000]

ตำแหน่งของสถานีที่ 2 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [190+000 , 213+333]

ตำแหน่งของสถานีที่ 3 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [313+333]

ตำแหน่งของสถานีที่ 4 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [376+667 , 425+000]



รูปที่ 4.1 หน้าต่างอินเตอร์เฟสแสดงผลลัพธ์จากการวางสถานีทั้ง 4 ในช่วงที่กำหนดไว้

4.3 เปรียบเทียบผลลัพธ์กับเป้าหมายโครงการของบริษัท

โครงงานนี้ศึกษาที่ความต้องการของปริมาณแร่ถ่านหินคือ 25 ล้านตันต่อปี

ระยะเวลาการทำงาน 344 วันต่อปี

ดังนั้น ปริมาณแร่ถ่านหินที่ต้องขนส่งต่อวันคือ 25,000,000 / 344 = 72,675 ตันต่อวัน

จำนวนพ่วงบรรทุกแร่ 216 คันต่อขบวน

ความสามารถบรรทุกของพ่วงบรรทุกแร่ 100 ตันต่อคัน ประสิทธิภาพของพ่วงบรรทุกแร่ 0.96

ความสามารถในการบรรทุกต่อขบวน 216 × 100 × 0.96 = 20,736 ตันต่อขบวน

ความเร็วเฉลี่ยของขบวนรถ 50 กิโลเมตรต่อชั่วโมง ระยะทางในการขนส่งประมาณ 500 กิโลเมตร

ระยะเวลาการเดินรถขณะบรรทุก,ไม่บรรทุก 500/50 = 10 ชั่วโมง

ระยะเวลาในการบรรทุกแร่ขึ้นขบวนรถ 5 ชั่วโมง

ระยะเวลาในการนำแร่ลงจากขบวนรถ 4 ชั่วโมง

ระยะเวลาอื่นๆ 4 ชั่วโมง (* ระยะเวลาอื่นๆ เช่น ระยะเวลารอเพื่อให้อีกขบวนบรรทุกหรือนำแร่ลง)

ระยะเวลาในการขนส่งต่อรอบที่คาดการณ์ไว้จะเท่ากับ 10+10+5+4+4 = 33 ชั่วโมง

$$\text{จำนวนขบวนรถที่ใช้} = \frac{\text{ปริมาณแร่ถ่านหินที่ต้องขนต่อวัน} \times \text{ระยะเวลาในการขนส่งต่อรอบ}}{\text{ความสามารถในการบรรทุกต่อขบวน} \times 24}$$

$$= \frac{72{,}675 \times 33}{20{,}736 \times 24} = 4.819 = 5 \text{ ขบวนรถ}$$

ซึ่งเมื่อเทียบกับค่าที่หามาได้จากการทดลองคือ 32.53 ชั่วโมง จะทำให้เห็นได้ว่าสามารถทำให้ปัญหาทางด้านระยะเวลาการขนส่งลดลงไปได้ 33 - 32.53 = 0.467 ชั่วโมง (28.00 นาที) หรือ สามารถทำให้ระยะเวลาขนส่งต่อรอบลดลงได้ร้อยละ 1.41

4.4 สรุปผล

จากการลองแทนค่าของจุดสับหลีกต่างๆสามารถทำให้เห็นความเปลี่ยนแปลงของระยะเวลาการเดินรถต่อรอบ และกราฟของการเดินรถอย่างเห็นได้ชัดว่าสามารถเดินเป็นไปได้หรือไม่ในการเดินรถ ซึ่งจากการแทนค่าตำแหน่งของสถานีที่ใช้เป็นจุดสับหลีกให้กับขบวนรถนั้นมีหลายตำแหน่งที่สามารถทำให้ขบวนรถมีระยะเวลาการเดินรถต่อรอบน้อยที่สุด โดยปัจจัยในการเลือกตำแหน่งในการตั้งสถานีนั้นควรสามารถที่จะยืดหยุ่นได้จึงเลือกช่วงของการวางตำแหน่งสถานีที่กว้างที่สุดนั่นคือ ตำแหน่งของสถานีที่ 1 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [113+334 , 125+000] ตำแหน่งของสถานีที่ 2 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [190+000 , 213+333] ตำแหน่งของสถานีที่ 3 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [313+333] และ ตำแหน่งของสถานีที่ 4 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [376+667,425+000] ซึ่งเป็นการวางตำแหน่งสถานีที่ทำให้ระยะเวลาในการขนส่งน้อยที่สุดและสามารถขยับตำแหน่งได้มากที่สุดตามสภาพแวดล้อมและภูมิประเทศนั้นๆ ซึ่งระยะเวลาการเดินรถต่อรอบเมื่อทำการวางตำแหน่งของสถานีในช่วงที่กำหนดนี้ จะสามารถหาระยะเวลาการเดินรถต่อรอบได้เท่ากับ 32 ชั่วโมง 32 นาที ซึ่งลดลงจากการคาดการ์ของโครงการที่มีระยะเวลาการเดินรถต่อรอบเท่ากับ 33 ชั่วโมงได้ถึง 0.467 ชั่วโมง (28.00 นาที) หรือ คิดเป็นร้อยละ 1.41

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

1. ปัญหาของการเดินรถไฟขนส่งแร่ถ่านหินแบบทางเดี่ยวนั้นคือ ความปลอดภัย และ ระยะเวลาในการขนส่ง เนื่องจากการที่รถไฟแบบทางเดี่ยวนั้นเป็นการเดินรถได้ในทั้ง 2 ทิศทางด้วยรางเดียวกัน จึงจำเป็นต้องมีสถานีไว้ใช้ในการสำหลีกขบวนรถไฟที่สวนทางมาเพื่อไม่ให้เกิดอุบัติเหตุ ปัญหาต่อมาเมื่อจะทำการก่อสร้างสถานีแต่ละสถานีนั้นปัจจัยหลักคือความคุ้มค่าในการสร้าง เนื่องจากการสร้างสถานีนั้นมีค่าใช้จ่ายค่อนข้างสูง อีกทั้งจำนวนของสถานีและตำแหน่งของการวางสถานีที่แตกต่างกันก็ยังส่งผลต่อระยะเวลาในการขนส่งและความเป็นไปได้ในการเดินรถนั้นยังส่งผลต่อระยะเวลาในการขนส่งเช่นกัน การศึกษาและแก้ปัญหาในการหาจำนวนและตำแหน่งของการวางสถานีนั้น ได้ใช้วิธีการออกแบบโปรแกรมในการตรวจสอบความเป็นไปได้ในการเดินรถและคำนวณหาระยะเวลาขนส่งจากการแทนตำแหน่งของสถานีต่างๆ รวมไปถึงการแสดงกราฟของตารางการเดินรถอีกด้วย

2.จากการทดลองและปรับค่าเพื่อแก้ปัญหาด้วยโปรแกรมทำให้หาผลสรุปได้ว่า ระยะเวลาในการขนส่งต่อหนึ่งรอบนั้นมีค่าน้อยที่สุดและเป็นไปได้ที่จะเดินรถอย่างปลอดภัยนั้นมีค่าเท่ากับ 32 ชั่วโมง 32 นาที (32.53 ชั่วโมง) ซึ่งมีค่าน้อยกว่าค่าของที่บริษัทได้คาดการณ์ไว้อยู่ 28.00 นาที (0.4667 ชั่วโมง) หรือลดลงไปร้อยละ 1.4142 โดยใช้จำนวนสถานีน้อยที่สุดคือ 4 สถานี และตำแหน่งของสถานีแต่มีความยืดหยุ่นในขยับตำแหน่งได้มากที่สุด โดยตำแหน่งของสถานีต่างๆ มีดังนี้ ตำแหน่งของสถานีที่ 1 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [113+334 , 125+000] ตำแหน่งของสถานีที่ 2 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [190+000 , 213+333] ตำแหน่งของสถานีที่ 3 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [313+333] และ ตำแหน่งของสถานีที่ 4 ที่สามารถใช้ได้จะอยู่ในช่วง กิโลเมตรที่ [376+667 , 425+000]

3. ในการพัฒนาโปรแกรมนี้ได้มีการออกแบบในการใช้ตัวแปรต่างๆ ที่สามารถปรับค่าได้ เช่น ระยะทางทั้งหมดในการขนส่ง ความเร็ว ระยะเวลาบรรทุกสินค้า ระยะเวลานำสินค้าลง ตำแหน่งในการวางสถานี เป็นต้น ตัวแปรเหล่านี้ทำให้โปรแกรมสามารถใช้คำนวณระยะเวลาการเดินรถต่อรอบและช่วยในการตรวจสอบความเป็นไปได้ในการเดินรถจากการตั้งสถานีในตำแหน่งต่างๆของโครงการอื่นได้เช่นกัน

5.2 ข้อเสนอแนะ

โปรแกรมที่เขียนขึ้นเป็นแค่ต้นแบบเบื้องต้น ซึ่งอาจจะมีข้อผิดพลาดหรือเงื่อนไขที่ยังไม่ครอบคลุมกับโครงการหรืองานที่ซับซ้อนมากนัก สามารถนำไปต่อยอดและปรับปรุงเพื่อให้โปรแกรมมีประสิทธิภาพเพิ่มมากขึ้น อีกทั้งยังสามารถนำใช้งานได้กับโครงการอื่นที่มีความซับซ้อนและขอบเขตที่กว้างขึ้นได้ในอนาคต

เอกสารอ้างอิง

[1] WONG OWEN (1975), A Computer Simulation of a Single-Line Railway System, Unpublished M.E. thesis, University of Auckland

[2] WONG OWEN (1975), IMPROVING SYSTEM PERFORMANCE FOR A SINGLE LINE RAILWAY WITH PASSING LOOPS, University of Auckland

[3] B. Szpigel (1973), Optimal train scheduling on a single line railway.

[4] Assad AA (1980), Models for rail transportation, Transportation Research Part A

[5] Valentina Cacchiani, Laura Galli and Paolo Toth ,"A Tutorial on Train Timetabling and Train Platforming Problems", University of Pisa

[6] P. Tormos et al (2008), A Genetic Algorithm for Railway Scheduling Problems.

[7] Diego Arenas, R´emy Chevrier, Said Hanafi and Joaquin Rodriguez (2015), Solving the Train Timetabling Problem, a mathematical model and a genetic algorithm solution approach, University of Valenciennes

[8] A. Higgins and L. Ferreira (1996), Optimal scheduling of trains on a single line track," transportation Research Part B

[9] James Ekmann and Patrick Le (2004) , Coal Storage and Transportation

[10] Leonard Victor Pearson, Moving block railway signaling, Loughborough University

[11] Cramer, Barton Emmet (2007), "North American freight rail: regulatory evolution, strategic rejuvenation, and the revival of an ailing industry", University of Iowa

[12] กรณ์พงษ์ อึงสถิตถาวร (2014), การหาค่าที่เหมาะสมที่สุดในการจัดตารางเวลารถไฟในระบบรางเดี่ยว

ภาคผนวก

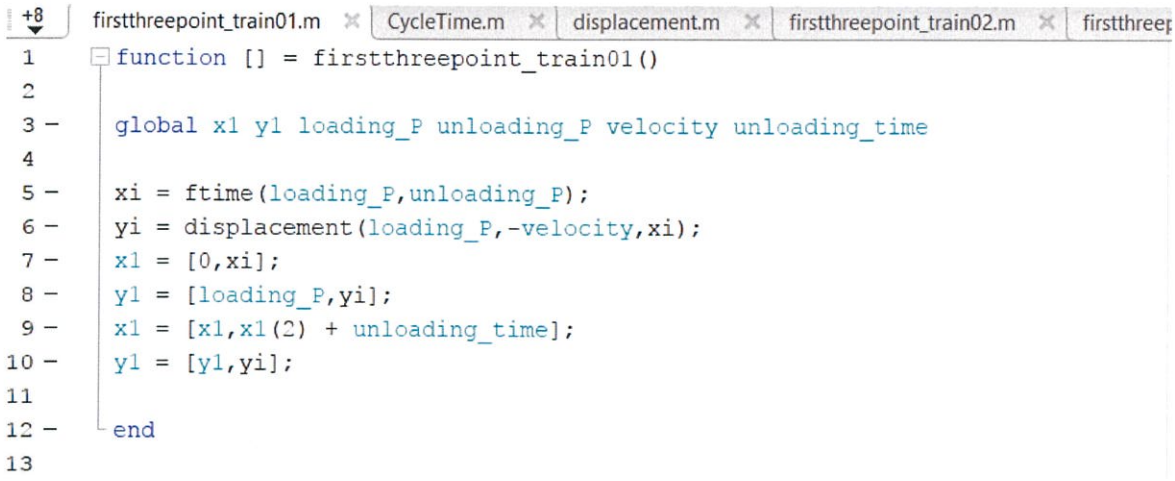
ภาคผนวก

รายละเอียดของโปรแกรมที่ใช้พัฒนาขึ้นมา

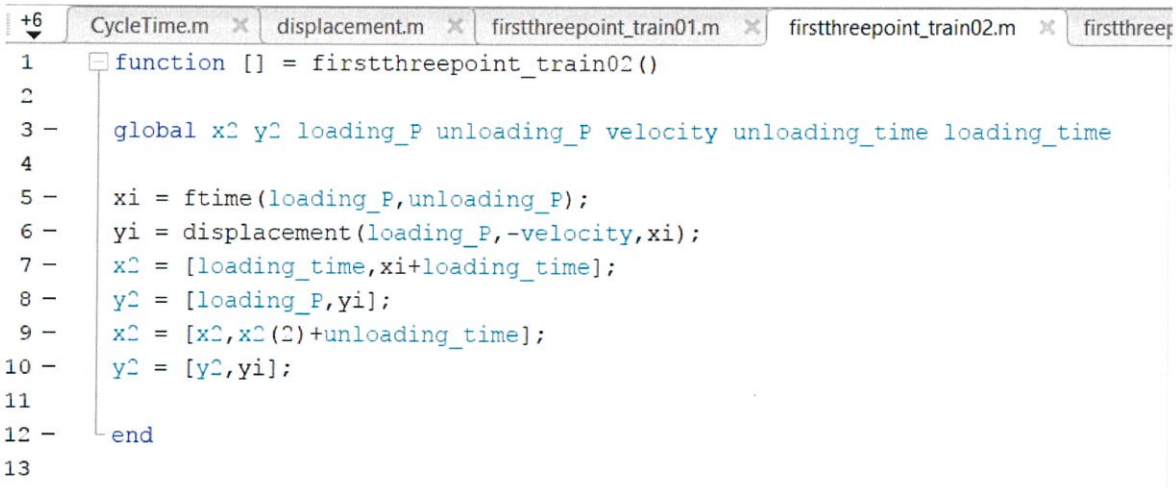
process_train01.m | waiting_P.m | firstthreepoint_train04.m | firstthreepoint_train03.m | firstthree

```
clc;clear;
global  loading_time unloading_time waiting_time_passing velocity
loading_time = 0;
unloading_time = 0;
waiting_time_passing = 0;
velocity = 0;
global  unloading_P station01 station02 station03 station04 loading_P
unloading_P = 0;
station01   = 0;
station02   = 0;
station03   = 0;
station04   = 0;
loading_P   = 500;
global x1 y1 x2 y2 x3 y3 x4 y4 x5 y5
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0;
y1 = 0; y2 = 0; y3 = 0; y4 = 0; y5 = 0;
%-------  Initialize -------
firstthreepoint_train01();
firstthreepoint_train02();
firstthreepoint_train03();
firstthreepoint_train04();
firstthreepoint_train05();
%--------  Processing ------
process_train01()
process_train02()
process_train03()
process_train04()
process_train05()
fprintf('Cycle Time Train1 = %g hr\n',x1(end-2)-x1(1));
fprintf('Cycle Time Train2 = %g hr\n',x2(end-2)-x2(1));
fprintf('Cycle Time Train3 = %g hr\n',x3(end-2)-x3(1));
fprintf('Cycle Time Train4 = %g hr\n',x4(end-2)-x4(1));
fprintf('Cycle Time Train5 = %g hr\n',x5(end-2)-x5(1));
%--------  Plotting ------
plotting()
```

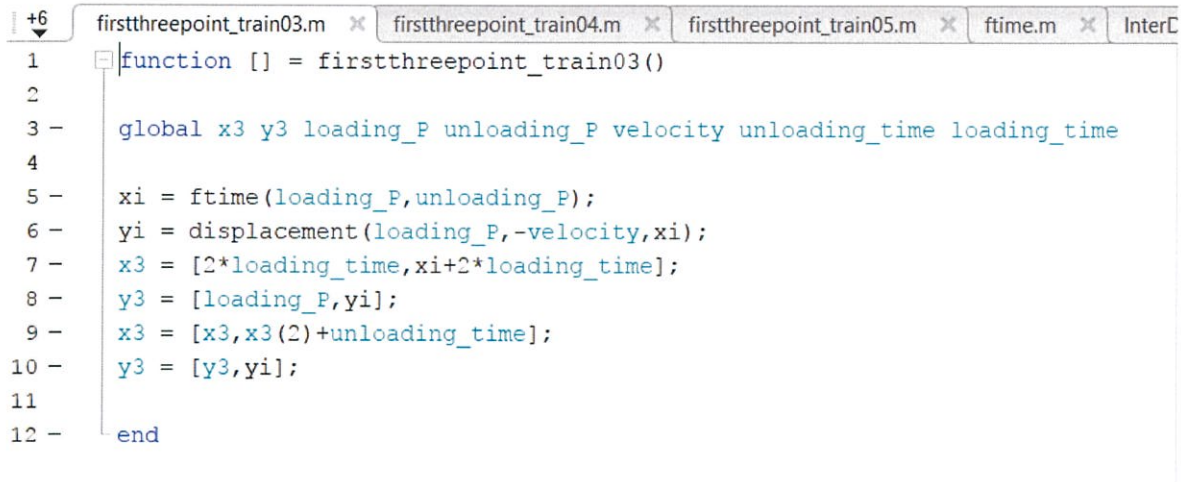
ภาพที่ 1 หน้าต่างของไฟล์ Cycletime.m ซึ่งแสดงถึงตัวแปรและลำดับการทำงานของฟังก์ชั่นอื่นๆ



ภาพที่ 2 หน้าต่าง firsthreepoint_train01.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขาไปของขบวนที่ 1



ภาพที่ 3 หน้าต่าง firsthreepoint_train02.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขาไปของขบวนที่ 2

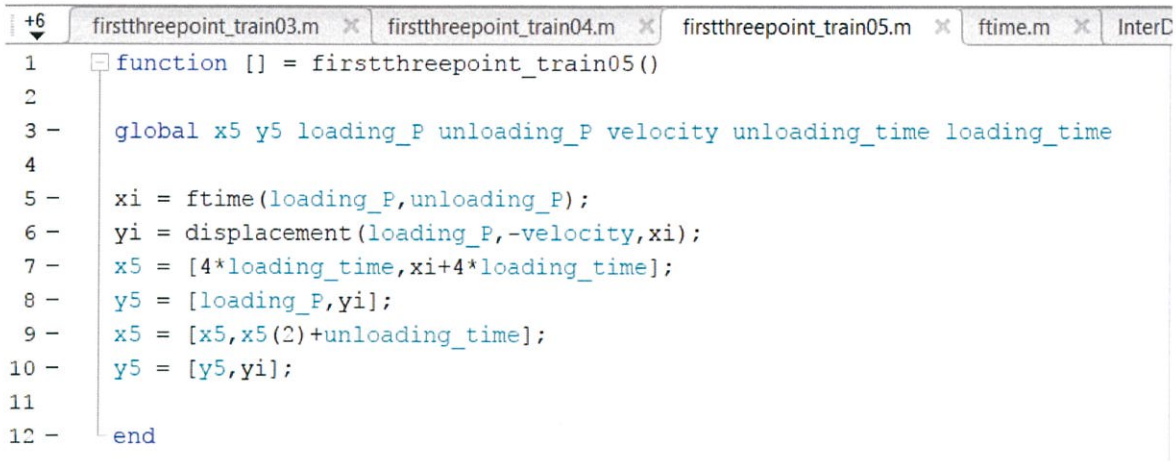


```
+6  firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterD
1   function [] = firstthreepoint_train03()
2
3 –  global x3 y3 loading_P unloading_P velocity unloading_time loading_time
4
5 –  xi = ftime(loading_P,unloading_P);
6 –  yi = displacement(loading_P,-velocity,xi);
7 –  x3 = [2*loading_time,xi+2*loading_time];
8 –  y3 = [loading_P,yi];
9 –  x3 = [x3,x3(2)+unloading_time];
10 –  y3 = [y3,yi];
11
12 –  end
```

ภาพที่ 4 หน้าต่าง firsthreepoint_train03.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขาไปของขบวนที่ 3

```
+6  firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterD
1   function [] = firstthreepoint_train04()
2
3 –  global x4 y4 loading_P unloading_P velocity unloading_time loading_time
4
5 –  xi = ftime(loading_P,unloading_P);
6 –  yi = displacement(loading_P,-velocity,xi);
7 –  x4 = [3*loading_time,xi+3*loading_time];
8 –  y4 = [loading_P,yi];
9 –  x4 = [x4,x4(2)+unloading_time];
10 –  y4 = [y4,yi];
11
12 –  end
```

ภาพที่ 5 หน้าต่าง firsthreepoint_train04.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขาไปของขบวนที่ 4



ภาพที่ 6 หน้าต่าง firsthreepoint_train05.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขาไปของขบวนที่ 5

+6 | firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterDisp_v2.m

```
function [] = process_train01()

global x1  y1 x2 y2 x3 y3 x4 y4 x5 y5  loading_P loading_time waiting_time_passing
X = x1;
Y = y1;
Waiting_Point = 0;
for train_N = [2 3 4 5] % edit
    switch train_N
        case {1,2,3,4,5}
            timeoffset = (train_N-1)*loading_time;
        case -1
            timeoffset = x1(end-2);
        case -2
            timeoffset = x2(end-2);
        case -3
            timeoffset = x3(end-2);
        case -4
           timeoffset = x4(end-2);
    end

    Intersection_D = InterDisp_v2(Waiting_Point,timeoffset,X(end));
    Waiting_Point = waiting_P(Intersection_D);

    if Waiting_Point == 0 || Waiting_Point == 500
        wtp = 0;
    else
        wtp = waiting_time_passing;
    end
    if  Waiting_Point ~= Y(end)
        timetostation = ftime(Waiting_Point,Y(end));
        X = [X, X(end)+ timetostation];
        Y = [Y, Waiting_Point];
    end
    tt = ftime(Waiting_Point,loading_P);
    X = [X, timeoffset + tt + wtp];
    Y = [Y, Waiting_Point];

    if Waiting_Point == loading_P
        X = [X, X(end)+ loading_time];  % edit
        Y = [Y, loading_P];
        break;
    end

end
if Waiting_Point ~= loading_P
    timetostation = ftime(loading_P,Y(end));
    X = [X, X(end)+ timetostation];
    Y = [Y, loading_P];
    X = [X,X(end)+loading_time];
    Y = [Y, loading_P];
end

X = [X,X(end)+x1(2:3)];
Y = [Y, y1(2:3)];

x1 = X;
y1 = Y;
```

ภาพที่ 7 หน้าต่าง process_train01.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขากลับของขบวนที่ 1

+6 | firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterDisp_v2.m

```
function [] = process_train02()

global x1  y1 x2 y2 x3 y3 x4 y4 x5 y5  loading_P loading_time waiting_time_passing
X = x2;
Y = y2;
Waiting_Point = 0;
for train_N = [3 4 5 -1] % edit
    switch train_N
        case {1,2,3,4,5}
            timeoffset = (train_N-1)*loading_time;
        case -1
            timeoffset = x1(end-2);
        case -2
            timeoffset = x2(end-2);
        case -3
            timeoffset = x3(end-2);
        case -4
            timeoffset = x4(end-2);
    end

    Intersection_D = InterDisp_v2(Waiting_Point,timeoffset,X(end));
    Waiting_Point = waiting_P(Intersection_D);

    if Waiting_Point == 0 || Waiting_Point == 500
        wtp = 0;
    else
        wtp = waiting_time_passing;
    end
    if  Waiting_Point ~= Y(end)
        timetostation = ftime(Waiting_Point,Y(end));
        X = [X, X(end)+ timetostation];
        Y = [Y, Waiting_Point];
    end
    tt = ftime(Waiting_Point,loading_P);
    X = [X, timeoffset + tt+ wtp];
    Y = [Y, Waiting_Point];

    if Waiting_Point == loading_P
        X = [X, x1(end-2)+ loading_time]; % edit
        Y = [Y, loading_P];
        break;
    end

end
if Waiting_Point ~= loading_P
    timetostation = ftime(loading_P,Y(end));
    X = [X, X(end)+ timetostation];
    Y = [Y, loading_P];
    X = [X,X(end)+loading_time];
    Y = [Y, loading_P];
end

X = [X,X(end)+x1(2:3)];
Y = [Y, y1(2:3)];

x2 = X;
y2 = Y;
```

ภาพที่ 8 หน้าต่าง process_train02.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขากลับของขบวนที่ 2

+6 | firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterDisp_v2.m

```
1   function [] = process_train03()
2   global x1  y1 x2 y2 x3 y3 x4 y4 x5 y5  loading_P loading_time waiting_time_passing
3   X = x3;
4   Y = y3;
5   Waiting_Point = 0;
6   for train_N = [4 5 -1 -2] % edit
7       switch train_N
8           case {1,2,3,4,5}
9               timeoffset = (train_N-1)*loading_time;
10          case -1
11              timeoffset = x1(end-2);
12          case -2
13              timeoffset = x2(end-2);
14          case -3
15              timeoffset = x3(end-2);
16          case -4
17              timeoffset = x4(end-2);
18      end
19
20      Intersection_D = InterDisp_v2(Waiting_Point,timeoffset,X(end));
21      Waiting_Point = waiting_P(Intersection_D);
22
23      if Waiting_Point == 0 || Waiting_Point == 500
24          wtp = 0;
25      else
26          wtp = waiting_time_passing;
27      end
28      if  Waiting_Point ~= Y(end)
29          timetostation = ftime(Waiting_Point,Y(end));
30          X = [X, X(end)+ timetostation];
31          Y = [Y, Waiting_Point];
32      end
33      tt = ftime(Waiting_Point,loading_P);
34      X = [X, timeoffset + tt + wtp];
35      Y = [Y, Waiting_Point];
36
37      if Waiting_Point == loading_P
38          X = [X, x2(end-2)+ loading_time]; % edit
39          Y = [Y, loading_P];
40          break;
41      end
42
43  end
44  if Waiting_Point ~= loading_P
45      timetostation = ftime(loading_P,Y(end));
46      X = [X, X(end)+ timetostation];
47      Y = [Y, loading_P];
48      X = [X,X(end)+loading_time];
49      Y = [Y, loading_P];
50  end
51
52  X = [X,X(end)+x1(2:3)];
53  Y = [Y, y1(2:3)];
54
55  x3 = X;
56  y3 = Y;
```

ภาพที่ 9 หน้าต่าง process_train03.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขากลับของขบวนที่ 3

+6 | firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterDisp_v2.m

```
function [] = process_train04()
global x1  y1 x2 y2 x3 y3 x4 y4 x5 y5  loading_P loading_time waiting_time_passing
X = x4;
Y = y4;
Waiting_Point = 0;
for train_N = [5 -1 -2 -3] % edit
    switch train_N
        case {1,2,3,4,5}
            timeoffset = (train_N-1)*loading_time;
        case -1
            timeoffset = x1(end-2);
        case -2
            timeoffset = x2(end-2);
        case -3
            timeoffset = x3(end-2);
        case -4
            timeoffset = x4(end-2);
    end

    Intersection_D = InterDisp_v2(Waiting_Point,timeoffset,X(end));
    Waiting_Point = waiting_P(Intersection_D);
    if Waiting_Point == 0 || Waiting_Point == 500
        wtp = 0;
    else
        wtp = waiting_time_passing;
    end
    if  Waiting_Point ~= Y(end)
        timetostation = ftime(Waiting_Point,Y(end));
        X = [X, X(end)+ timetostation];
        Y = [Y, Waiting_Point];
    end
    tt = ftime(Waiting_Point,loading_P);
    X = [X, timeoffset + tt + wtp];
    Y = [Y, Waiting_Point];

    if Waiting_Point == loading_P
        X = [X, x3(end-2)+ loading_time]; % edit
        Y = [Y, loading_P];
        break;
    end

end
if Waiting_Point ~= loading_P
    timetostation = ftime(loading_P,Y(end));
    X = [X, X(end)+ timetostation];
    Y = [Y, loading_P];
    X = [X,X(end)+loading_time];
    Y = [Y, loading_P];
end

X = [X,X(end)+x1(2:3)];
Y = [Y, y1(2:3)];

x4 = X;
y4 = Y;
```

ภาพที่ 10 หน้าต่าง process_train04.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขากลับของขบวนที่ 4

+6 | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | InterDisp_v2.m | InterTime.m | plotting.m

```
function [] = process_train05()
global x1  y1 x2 y2 x3 y3 x4 y4 x5 y5  loading_P loading_time waiting_time_passing
X = x5;
Y = y5;
Waiting_Point = 0;
for train_N = [-1 -2 -3 -4] %
    switch train_N
        case {1,2,3,4,5}
            timeoffset = (train_N-1)*loading_time;
        case -1
            timeoffset = x1(end-2);
        case -2
            timeoffset = x2(end-2);
        case -3
            timeoffset = x3(end-2);
        case -4
            timeoffset = x4(end-2);
    end

    Intersection_D = InterDisp_v2(Waiting_Point,timeoffset,X(end));
    Waiting_Point = waiting_P(Intersection_D);

    if Waiting_Point == 0 || Waiting_Point == 500
        wtp = 0;
    else
        wtp = waiting_time_passing;
    end
    if  Waiting_Point ~= Y(end)
        timetostation = ftime(Waiting_Point,Y(end));
        X = [X, X(end)+ timetostation];
        Y = [Y, Waiting_Point];
    end
    tt = ftime(Waiting_Point,loading_P);
    X = [X, timeoffset + tt + wtp];
    Y = [Y, Waiting_Point];

    if Waiting_Point == loading_P
        X = [X, x4(end-2)+ loading_time]; % edit
        Y = [Y, loading_P];
        break;
    end

end
if Waiting_Point ~= loading_P
    timetostation = ftime(loading_P,Y(end));
    X = [X, X(end)+ timetostation];
    Y = [Y, loading_P];
    X = [X,X(end)+loading_time];
    Y = [Y, loading_P];
end

X = [X,X(end)+x1(2:3)];
Y = [Y, y1(2:3)];

x5 = X;
y5 = Y;
```

ภาพที่ 11 หน้าต่าง process_train05.m
แสดงการเขียนโค้ดของฟังก์ชั่นการหาตำแหน่งของรถช่วงขากลับของขบวนที่ 5

ftime.m | displacement.m | InterDisp_v2.m | InterTime.m

```
function [ ti ] = ftime( end_P,start_P)
global velocity
if end_P - start_P > 0
    d = 1;
else
    d = -1;
end
ti = (end_P - start_P)/(d*velocity);
end
```

ภาพที่ 12 หน้าต่าง ftime.m
แสดงการเขียนโค้ดของฟังก์ชั่นหาระยะเวลาการเดินรถจากจุดเริ่มต้นไปจุดสุดท้าย

displacement.m | ftime.m | InterDisp_v2.m | InterTime.m

```
function [ S ] = displacement( So,V,t)
S = So + V*t;
end
```

ภาพที่ 13 หน้าต่าง displacement.m
แสดงการเขียนโค้ดของฟังก์ชั่นหาระยะทางที่รถเคลื่อนที่จากจุดเริ่มต้นไปยังปลายทาง

InterDisp_v2.m | displacement.m | ftime.m | InterTime.m

```
function [S] = InterDisp_v2( station_P,timeoffset,ti)

global loading_P velocity

S = ((loading_P + station_P) + velocity*(timeoffset-ti))/2;

end
```

ภาพที่ 14 หน้าต่าง InterDisp_v2.m
แสดงการเขียนโค้ดของฟังก์ชั่นหาจุดตัดของรถไฟสองขบวนที่วิ่งสวนทางกัน

InterTime.m | InterDisp_v2.m | displacement.m | ftime.m

```
function [t] = InterTime( station_P,N,ti)
global loading_P velocity loading_time
t = ((loading_P - station_P) + velocity*((N-1)*loading_time+ti))/(2*velocity);
end
```

ภาพที่ 15 หน้าต่าง InterTime.m
แสดงการเขียนโค้ดของฟังก์ชั่นหาเวลาที่รถสองขบวนที่สวนทางกันเคลื่อนที่มาตัดกัน

waiting_P.m | process_train01.m | firstthreepoint_train04.m | firstthreepoint_train03.m

```
function [ Pi ] = waiting_P( s )
    global loading_P unloading_P station01 station02 station03 station04
    if s <= loading_P && s >= unloading_P
        if s == loading_P
            Pi = loading_P;
        elseif s >= station04
            Pi = station04;
        elseif s >= station03
            Pi = station03;
        elseif s >= station02
            Pi = station02;
        elseif s >= station01
            Pi = station01;
        else
            Pi = unloading_P;
        end
    else
        Pi = loading_P;
    end
end
```

ภาพที่ 16 หน้าต่าง waiting_P.m
แสดงการเขียนโค้ดของฟังก์ชั่นเงื่อนไขในการเลือกสถานี

plotting.m | CycleTime.m | displacement.m | firstthreepoint_train01.m | firstthreepoint_train02.m | firstthreepoint_train03.m | firstthreepoint_train04.m | firstthreepoint_train05.m | ftime.m | GUI_Train.m | InterDisp.m

```
function [] = plotting()
global  station01 station02 station03 station04 unloading_P loading_P
global x1 y1 x2 y2 x3 y3 x4 y4 x5 y5

T_range = [0 x5(end)];
S0 = [unloading_P ,unloading_P ];
S1 = [station01 ,station01 ];
S2 = [station02 ,station02 ];
S3 = [station03 ,station03 ];
S4 = [station04 ,station04 ];
S5 = [loading_P ,loading_P ];

plot(x1,y1,'-o',x2,y2,'-o',x3,y3,'-o',x4,y4,'-o',x5,y5,'-o',T_range,S0,'k:',T_range,S1,'k:',T_range,S2,'k:',T_range,S3,'k:',T_range,S4,'k:',T_range,S5,'k:','lineWidth',1.5);
legend('train 1','train 2','train 3','train 4','train 5')
axis([0,x5(end),-10,loading_P+10])
ylabel('Distance (km)');
xlabel('Time (hr)');
end
```

ภาพที่ 17 หน้าต่าง plotting.m
แสดงการเขียนโค้ดของฟังก์ชั่นในการสร้างกราฟแสดงตำแหน่งต่างๆของรถแต่ละขบวน