บทที่ 3
วิธีการและการพัฒนาโปรแกรม

จากแนวคิดและทฤษฎีที่เกี่ยวข้องในบทที่ 2 แสดงให้เห็นว่า วิธีการประเมินค่าความปลอดภัยของแนวเส้นทางจากความเร็วและระยะมองเห็น สามารถพัฒนาซอฟแวร์ทางด้าน CAD มาเพื่อประยุกต์ใช้ในการตรวจสอบแนวเส้นทางที่ออกแบบ โดยการวิจัยนี้เลือกโปรแกรม AutoCAD และ Softdesk มาใช้ในการพัฒนา เนื่องจากโปรแกรม AutoCAD นั้นมี AutoLISP ที่เป็นเครื่องมือในการเขียนโปรแกรมคำสั่งขึ้นมาใหม่ภายใน AutoCAD ซึ่งการวิจัยได้นำความสามารถนี้มาใช้ในการพัฒนารหัสคำสั่งของโปรแกรม และในส่วนของโปรแกรม Softdesk นั้นนำมาใช้ในการเตรียมข้อมูลที่จำเป็นต่อการคำนวณ ป้อนเข้าสู่โปรแกรม ได้แก่ ข้อมูลตำแหน่ง และข้อมูลแนวเส้นทาง เป็นต้น

โดยในบทนี้ จะนำเสนอแนวคิด วิธีการ และหลักการทำงานของโปรแกรมที่พัฒนา โดยมีเนื้อหาสำคัญอยู่ 4 ส่วน

1) โปรแกรมที่นำมาใช้ในการพัฒนา
2) โปรแกรมที่พัฒนา
3) แนวคิดและหลักการทำงานของโปรแกรมหาความเร็ว
4) แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็น

ซึ่งรายละเอียดในแต่ละส่วนมีดังนี้

3.1 <u>โปรแกรมที่นำมาใช้ในการพัฒนา</u>

3.1.1 โปรแกรม AutoCAD

การที่ซอฟแวร์ทางด้าน CAD สามารถนำมาประยุกต์ใช้ในงานด้านต่างๆอย่างแพร่หลายนั้น เนื่องจากสามารถแสดงข้อมูลรายละเอียดคุณสมบัติของวัตถุคือ ตำแหน่ง และขนาด ซึ่งเป็นข้อมูลพื้นฐานประจำตัวของวัตถุที่วาดได้ จึงทำให้ปัจจุบันความสามารถของซอฟแวร์นี้มีการพัฒนาเพิ่มขึ้นอย่างรวดเร็วและนำมาประยุกต์ใช้งานหลายด้าน เช่น ด้าน Graphical Information System (GIS) ด้านโครงสร้าง งานสำรวจ งานท่อ งานปฏิรูปที่ดิน และงานทาง เป็นต้น เกิดเป็นโปรแกรมต่างๆ เฉพาะด้านเชิงพาณิชย์ เช่น โปรแกรม AutoCAD 3D-Studio MAX Microstation Bentley แต่ AutoCAD เป็นโปรแกรมด้าน CAD ที่ได้รับความนิยมมากที่สุด เนื่องจากมีการประยุกต์ใช้งานอย่างกว้างขวาง สามารถนำเอาข้อมูลที่ส่งออก (Output) ไปใช้ร่วมกับโปรแกรมอื่นๆได้

นอกจากนี้ AutoCAD ยังสนับสนุนให้สามารถสร้างคำสั่งใหม่ขึ้นมา เพื่อเป็นเครื่องมือและดัดแปลงให้สะดวกแก่การใช้งานที่ต้องการหลายวิธี สรุปได้ดังนี้

1) การสร้างเมนูใหม่ โดยการสร้างไฟล์ .mnu ด้วยการใช้มาโคร โดยการนำคำสั่งที่มีอยู่แล้วมารวมกัน พร้อมทั้งกำหนดตัวเลือก หรือรอรับข้อมูล
2) การทำไฟล์ Script โดยการสร้างไฟล์ .scr ซึ่งเป็นไฟล์ที่รวบรวมชุดคำสั่ง และค่าพารามิเตอร์ต่างๆไว้
3) DIESEL (Direct Interpretively Evaluated String Expression Language) เป็นรูปแบบที่ใช้แสดงข้อความสตริง สามารถใช้ร่วมกับ AutoLISP
4) การสร้าง Dialog Box ขึ้นมาใหม่ เพื่อใช้ร่วมกับ AutoLISP หรือ ADS ซึ่งจะเป็นไฟล์ .dcl
5) โปรแกรม AutoLISP เป็นเครื่องมือที่ใช้ในการเขียนรหัสคำสั่ง หรือฟังก์ชันใช้งานขึ้นมาใหม่ภายใน AutoCAD ซึ่งมีรูปแบบของรหัสคำสั่งที่ไม่ยาก และมีจำนวนรหัสคำสั่งไม่มากนัก และค่อนข้างง่ายต่อการวิจัย จึงเป็นเครื่องมือที่นิยมใช้ในการดัดแปลงหรือปรับปรุง AutoCAD
6) ADS เป็นการเขียนโปรแกรมในภาษา C ซึ่งค่อนข้างจะมีความซับซ้อนกว่าการใช้งาน AutoLISP มาก แต่ข้อดีของ ADS ก็คือเป็นภาษาที่ถูกคอมไพล์ ทำให้สามารถทำงานได้เร็วกว่า AutoLISP ซึ่งเป็นภาษาที่ถูกอินเตอร์พรีท

โดยปัจจุบันนั้นโปรแกรม AutoCAD ได้พัฒนาเครื่องมืออีก 2 ส่วนที่สำคัญคือ

Visual Basic Application (VBA) เป็นเครื่องมือซึ่งมีประโยชน์อย่างมากในการสร้าง Dialog Box ที่สามารถโต้ตอบแบบ Interactive กับ AutoCAD ได้เป็นอย่างดี ทำให้สร้าง Dialog Box ได้รวดเร็วกว่าการสร้างโดย AutoLISP ที่เป็นไฟล์ .dcl

VisualLISP เป็นเครื่องมือที่ช่วยในการสร้างโปรแกรม AutoLISP ให้สะดวกขึ้นเป็นอย่างมาก และสามารถคอมไพล์ โปรแกรม AutoLISP ให้เป็นไฟล์ .arx และ .vlx สามารถประมวลผลได้เร็วกว่าโปรแกรม AutoLISP เนื่องจากเป็นภาษาที่ถูกคอมไพล์ ไม่ต้องถูกอินเตอร์พรีทเมื่อประมวลผล

เกียรติขจร ไชยแสงสุขกุล (2544) ให้ข้อมูลว่าโปรแกรมภาษา AutoLISP เริ่มรวมอยู่ในโปรแกรม AutoCAD ตั้งแต่เวอร์ชัน 2.18 ในปี ค.ศ. 1986 เป็นต้นมา และมีการพัฒนาควบคู่ไปกับโปรแกรม AutoCAD อย่างต่อเนื่องทุกเวอร์ชันอันจะสามารถลำดับพัฒนาการของภาษา AutoLISP ใน AutoCAD ดังตารางที่ 3.1

ตารางที่ 3.1 การพัฒนา AutoLISP ในโปรแกรม AutoCAD เวอร์ชันต่างๆ

เวอร์ชัน	การเปลี่ยนแปลง
Release 7 (2.5)	เพิ่มฟังก์ชันในการจัดการฐานข้อมูลของ Drawing
Release 8 (2.6)	เริ่มมีระบบจุดสามมิติและเพิ่มฟังก์ชัน getcorner, getword และ intget
Release 9	เพิ่มฟังก์ชัน Command และ ssget
Release 10	เพิ่มความสามารถของฟังก์ชัน findfile และ load และเพิ่มฟังก์ชัน getenv, handent และ vports
Release 11	เพิ่มความสามารถของฟังก์ชัน entget และเพิ่มฟังก์ชัน cvunits และ entmake
Release 12	เพิ่มความสามารถของฟังก์ชัน ssget และ intget และเพิ่มฟังก์ชัน alert, getfield และ textbox รวมทั้งเพิ่มความสามารถการโปรแกรม Dialog Box ด้วยภาษา DCL (Dialog Box Control Language) และฟังก์ชันAutoLISP ในการจัดการ Dialog Box ทั้งหมดด้วย
Release 13	เพิ่มฟังก์ชันอีกหลายตัว เช่น acad_strlsort, Autoload, Help และ Acad_colordlg
Release 14	มีฟังก์ชัน VisualLISP เพิ่มเข้ามาและโปรแกรม VisualLISP ที่เป็นเครื่องมือในการพัฒนา Application ของ AutoLISP (แต่จำหน่ายแยกต่างหาก)
Release 2000	รวมเอา VisualLISP เข้าไว้ใน AutoCAD

ที่มา : เกียรติขจร ไชยแสงสุขกุล (2544: 4)

โดยที่การพัฒนา AutoLISP ในโปรแกรม AutoCAD เป็นการพัฒนาที่ยังคงรักษารูปแบบและความสามารถของฟังก์ชันเดิมไว้มีการเปลี่ยนแปลงเงื่อนไขข้อกำหนดของฟังก์ชันน้อยมาก ดังนั้นจึงไม่เป็นอุปสรรคมากนักต่อการแก้ไขรหัสคำสั่งของเวอร์ชันเดิม เพื่อให้ทำงาน AutoCAD เวอร์ชันใหม่แต่อย่างใด จึงนับได้ว่าภาษา AutoLISP เป็นภาษาที่มีความมั่นคงอย่างมาก และถึงแม้ว่าใน AutoCAD ตั้งแต่ Release 14 เป็นต้นมา Autodesk ได้เสนอแนวทางการพัฒนาโปรแกรมในรูปแบบใหม่ทั้ง VBA และ ObjectARX (ที่พัฒนาจาก ADS) แต่ Autodesk ก็ยังให้คำมั่นว่าจะยังคงพัฒนา AutoLISP อย่างต่อเนื่องต่อไป เพราะ AutoLISP มีความแข็งแกร่งในเรื่องของความง่ายในการพัฒนาโปรแกรม รวมทั้งออกแบบวิธีการใช้งานผ่าน Dialog Box ด้วยภาษา DCL นั้นก็ง่ายกว่าการออกแบบใน VBA มาก รวมทั้งเรื่องของความเร็วที่ AutoLISP เหนือกว่าด้วยหาก Compile ด้วย VisualLISP จึงทำให้ AutoLISP จะยังคงเป็นภาษาที่ใช้ในการพัฒนา Application บน AutoCAD อย่างต่อเนื่องอีกต่อๆไป

การวิจัยจึงเลือกพัฒนาโปรแกรม โดยใช้ AutoLISP เป็นภาษาที่ใช้ในการพัฒนารหัสคำสั่งโปรแกรม ในการวิจัยครั้งนี้

3.1.2 โปรแกรม Softdesk

โปรแกรม Softdesk เป็นโปรแกรมที่นำเอาความสามารถของโปรแกรม AutoCAD มาพัฒนาเพื่อใช้ในการออกแบบทางเรขาคณิตของถนนโดยเฉพาะ เริ่มแรกได้พัฒนาให้ใช้กับโปรแกรม AutoCAD Release 12 โดยใช้ชื่อว่า "AdCADD Civil/Survey Release 12 " และในปัจจุบันได้พัฒนาใช้กับโปรแกรม AutoCAD Release 14 โดยใช้ชื่อโปรแกรมว่า "Softdesk 8" และปัจจุบัน Autodesk ซึ่งเป็นเจ้าของลิขสิทธิ์ AutoCAD ได้ซื้อลิขสิทธิ์จาก Softdesk แล้วนำมาพัฒนาแบ่งออกเป็น 2 โปรแกรมย่อย คือ Survey for Land กับ Civil Design

ในการพัฒนาโปรแกรมได้นำโปรแกรม Softdesk 8 มาใช้ในเตรียมข้อมูลแนวเส้นทางและตำแหน่งต่างๆบนแนวเส้นทาง ซึ่งโปรแกรม Softdesk 8 นั้นประกอบด้วย Module ต่างๆ ที่ใช้สำหรับงานวิศวกรรมเฉพาะด้าน จำนวน 8 Module มีรายละเอียดโดยสังเขปดังต่อไปนี้

1) **COGO** (Coordinate Geometry) มีหน้าที่ในการจัดการกับระบบข้อมูลที่เกี่ยวกับตำแหน่งของจุดต่างๆ แล้วเก็บไว้ในฐานข้อมูล (Point Database) ใช้ในงานระบบพิกัดสำหรับงานทางและงานสำรวจ เป็น Module พื้นฐานที่จะประกอบอยู่ในทุกๆ Module
2) **SURVEY** ใช้ในงานด้านสำรวจ ปรับแก้วงรอบ การจัดเก็บข้อมูลสำรวจจากภาคสนาม สามารถนำข้อมูลการสำรวจเข้าสู่โปรแกรมได้หลายทาง ซึ่งอาจใช้จะชุดคำสั่งภายในของโปรแกรมที่ Survey Command Line, Batch file หรือ โดยการป้อนผ่านทางตารางป้อนข้อมูล ซึ่งต้องอยู่ในรูปแบบมาตรฐานของโปรแกรม และสามารถนำข้อมูลป้อนเข้าโปรแกรมได้โดยตรงจากกล้องสำรวจอิเล็คโทรนิค (Total Station) บางยี่ห้อได้
3) **DTM** (Digital Terrain Modeling) ใช้ในการสร้างโครงข่ายรูปสามเหลี่ยม (Triangulated Irregular Networks : TIN) จากข้อมูลพิกัด และค่าระดับของจุดต่างๆ นำมาสร้างเป็นพื้นผิวภูมิประเทศ 3 มิติในลักษณะ Grid และคำนวณเพื่อแสดงเส้นชั้นความสูงจาก TIN
4) **Advance Design** ใช้ในการออกแบบแนวเส้นทางของถนน, สร้างเส้นแนวตัดขวางตามยาว, สร้างรูปหน้าตัด, การยกโค้ง, คำนวณปริมาณวัสดุ รวมทั้งรูปแบบและข้อมูลที่ได้จากการคำนวณของโปรแกรมที่สามารถนำไปใช้ก่อสร้างทางได้ เป็น Module ที่นำมาใช้มากที่สุดในการพัฒนาโปรแกรมครั้งนี้

5) **Design** ใช้ในการออกแบบถนน และมีความสามารถคล้ายกันกับ Advance Design แต่มีฟังก์ชันบางอย่างที่ไม่มี คือ ฟังก์ชันการคำนวณยกโค้ง การสร้างเส้นโค้ง Spiral แต่จะมีฟังก์ชันนอกเหนือจากฟังก์ชันของ Advance Design ในการออกแบบท่อของถนนเพิ่มเติม

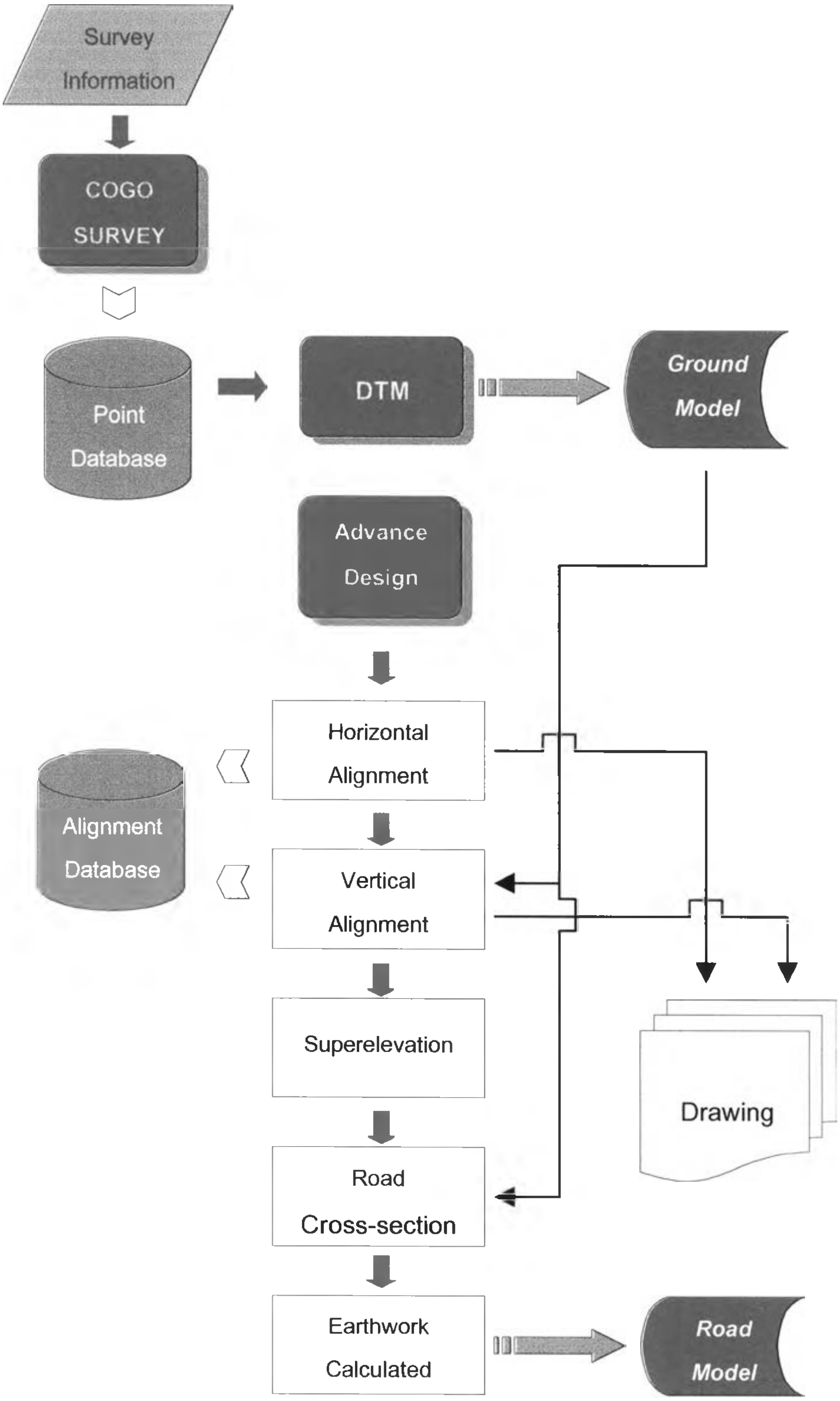
6) **Landscape** ใช้ในการออกแบบผังโครงการ ภูมิทัศน์ เช่นเรื่องของทางเดิน สนามกีฬาทางจักรยาน สวนต้นไม้ ฯลฯ จัดเก็บข้อมูลต่างๆอยู่ในฐานข้อมูลแบบตาราง

7) **Earthworks** ใช้ในการคำนวณหาปริมาณงานดินขุด-ถม เป็น Module ที่มีความสามารถคล้ายกับ DTM Module แต่มีฟังก์ชันที่ช่วยในการคำนวณหาปริมาณงานดินขุด-ถมเพิ่มเติม

8) **Hydrology** ใช้ในการออกแบบชลศาสตร์ ช่วยในการคำนวณค่าต่างๆทางชลศาสตร์ เช่น ค่า Rational SCS TR-55 Pond Parameter เป็นต้น

จุดเด่นของโปรแกรมที่เลือกใช้นำสำหรับการวิจัยในครั้งนี้คือ รูปแบบผลการคำนวณที่ได้จากโปรแกรม เป็นรูปแบบมาตรฐานซึ่งสามารถนำข้อมูลไปพัฒนาได้ง่าย ดังแสดงในภาคผนวก ก ผลการคำนวณที่ได้จากโปรแกรม Softdesk

ขั้นตอนการใช้ Module ต่างๆ เพื่อช่วยในการออกแบบถนน แสดงได้ดังรูปที่ 3.1



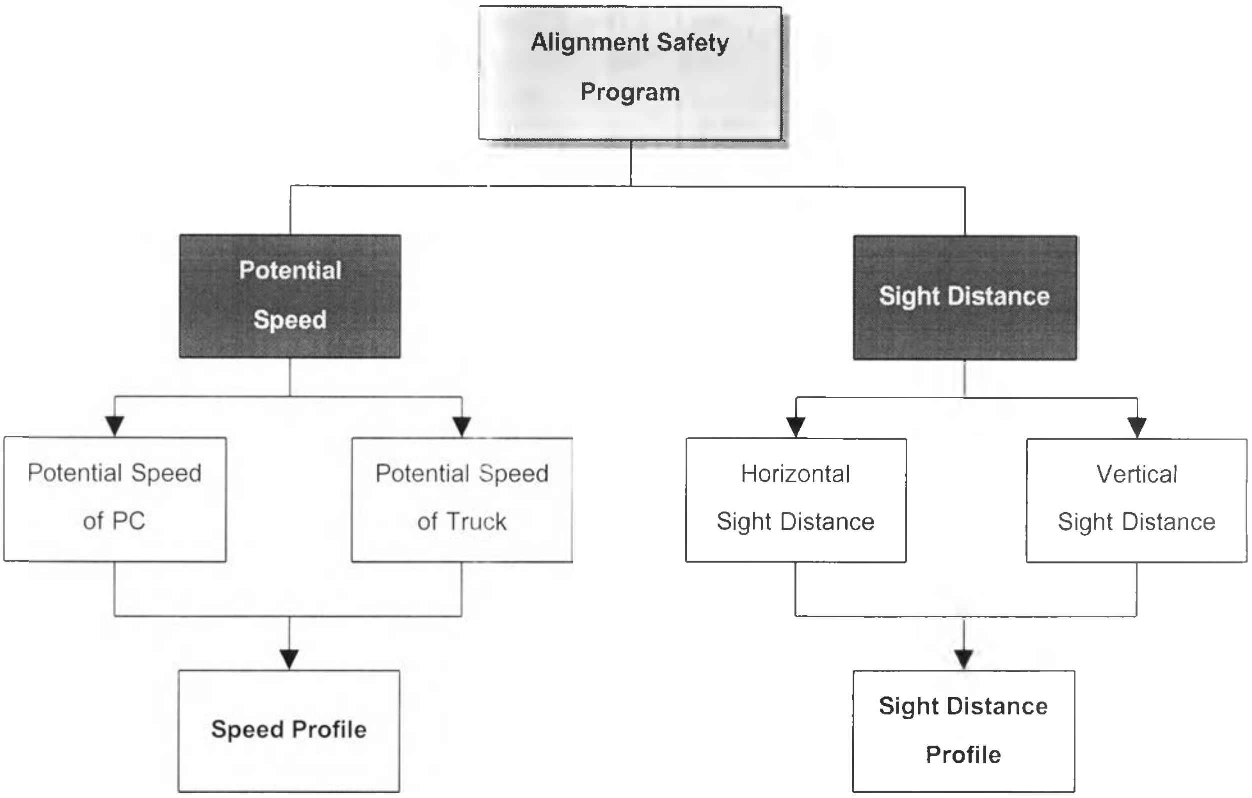
รูปที่ 3.1 แผนผังการใช้โปรแกรม Softdesk ช่วยในการออกแบบถนน

3.2 <u>โปรแกรมที่พัฒนา</u>

โปรแกรมที่พัฒนามีโครงสร้างหลักแบ่งออกเป็น 2 ส่วน คือ

1) โปรแกรมหาความเร็วที่เป็นไปได้ (Potential Speed) แบ่งเป็น 3 โปรแกรมย่อย
    - โปรแกรมหาความเร็วที่เป็นไปได้ของรถยนต์นั่ง (Potential Speed of PC)
    - โปรแกรมหาความเร็วที่เป็นไปได้ของรถบรรทุก (Potential Speed of Truck)
    - โปรแกรมแสดงเส้นผังความเร็ว (Speed Profile)
2) โปรแกรมหาระยะมองเห็น (Sight Distance) แบ่งเป็น 3 โปรแกรมย่อย
    - โปรแกรมหาระยะมองเห็นในแนวราบ (Horizontal Sight Distance)
    - โปรแกรมหาระยะมองเห็นในแนวดิ่ง (Vertical Sight Distance)
    - โปรแกรมแสดงเส้นผังระยะมองเห็น (Sight Distance Profile)

ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 แผนผังหลักแสดงส่วนประกอบของโปรแกรมที่พัฒนา

3.3 แนวคิดและหลักการทำงานของโปรแกรมหาความเร็ว

3.3.1 ความเร็วที่เป็นไปได้ของรถยนต์นั่ง

การคำนวณหาความเร็วที่เป็นไปได้ของรถยนต์นั่ง โดยวิธีการของ Ruediger ที่คำนวณหาความเร็วที่ 85 เปอร์เซ็นต์ไทล์ หรือ V85 นั้น เป็นวิธีการที่เหมาะสมแก่การพัฒนาโปรแกรม เนื่องจากมีข้อดี ดังนี้

1) แนวทางราบสามารถแปลงเป็นค่า CRR ทุกประเภท ไม่ว่าจะเป็น ทางตรง โค้งกลม โค้งสไปรอล โค้งกลับ และโค้งประกอบ เพื่อนำไปคำนวณหาความเร็ว V85 จากแบบจำลองความสัมพันธ์ระหว่าง CRR กับความเร็ว V85 ของประเทศต่างๆ
2) เกณฑ์ประเมินความปลอดภัยของแนวเส้นทาง จากความเร็ว V85 ที่คำนวณจากองค์ประกอบแนวเส้นทาง มีหลายวิธีดังที่เสนอมาแล้วในบทที่ 2

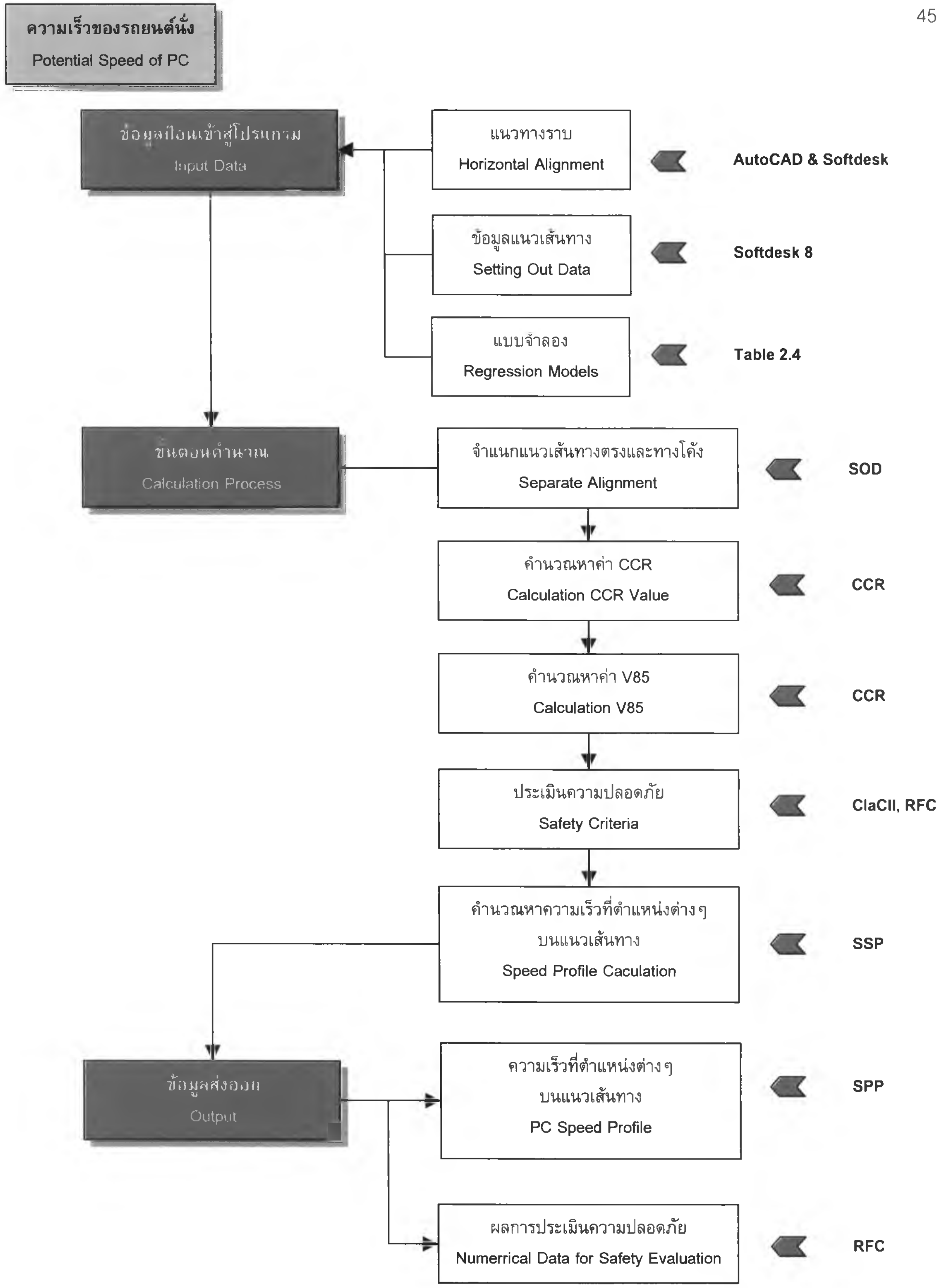
แต่มีข้อจำกัดของวิธีการคือ ไม่เหมาะสมสำหรับแนวทางราบซึ่งมีความลาดชันมากกว่า 6 % เพราะความลาดชันเริ่มมีผลกระทบต่อความเร็วที่เป็นไปได้ของรถยนต์นั่ง ซึ่งจะทำให้แบบจำลองความสัมพันธ์ระหว่าง CRR กับความเร็ว V85 ไม่สอดคล้องกับพฤติกรรมเกิดขึ้นจริง

โปรแกรมที่พัฒนามีแนวคิดและหลักการทำงานในการหาความเร็ว V85 ของรถยนต์นั่งตามแผนผังแสดงดังรูปที่ 3.3 โดยมีรายละเอียดดังนี้

ข้อมูลป้อนเข้าสู่โปรแกรม

❑ **แนวทางราบ (Horizontal Alignment)**

เป็นแนวทางราบที่ต้องการหาความเร็ว V85 ของรถยนต์นั่ง ได้แก่ส่วนประกอบของแนวเส้นทาง ทางตรง ทางโค้ง ซึ่งเตรียมในโปรแกรม AutoCAD ได้จากคำสั่ง Line และ Arc ในเมนู Draw ในส่วนของโปรแกรม Softdesk นั้นมีเมนู Entity ซึ่งรวบรวมคำสั่งสร้าง Tangent, Circular Curve และ Spiral อยู่ใน Advanced Design Module เมื่อเตรียมแนวทางราบแล้วเสร็จต้องแปลงส่วนของเส้นตรง และเส้นโค้งต่างๆ ให้โปรแกรม Softdesk ทราบว่าเส้นต่างๆ (Entities) เหล่านั้นเป็นแนวเส้นทางของถนน จากคำสั่ง Define Alignment จากเมนู Align ของโปรแกรม Softdesk



รูปที่ 3.3 แผนผังแสดงแนวคิดและหลักการทำงานของโปรแกรม
ประเมินความปลอดภัยในการออกแบบความเร็วของรถยนต์นั่ง

❑ **ข้อมูลแนวเส้นทาง (Setting Out Data)**

เมื่อ Define Alignment แล้วเสร็จ โปรแกรม Softdesk จะตรวจสอบคุณสมบัติของเส้น ได้แก่ความยาว และรัศมีโค้ง ฯลฯ แปลงเป็นข้อมูลแนวเส้นทาง ประกอบด้วยข้อมูล Description Point Station Spiral/Curve Data Tangent Data Azimuth Northing และ Easting ข้อมูลต่างๆเหล่านี้โปรแกรม Softdesk สามารถส่งข้อมูลออก (Export) ในรูปไฟล์ข้อความ (Text File) จากคำสั่ง Alignment Commands > Edit... จากเมนู Align ซึ่งจะนำข้อมูลต่างๆของแนวเส้นทางนี้ไปคำนวณหาค่า CCR ต่อไป

❑ **แบบจำลอง (Regression Models)**

แบบจำลองความสัมพันธ์ระหว่าง CCR กับ ความเร็ว V85 ของประเทศต่างๆ โดยในการพัฒนาโปรแกรมครั้งนี้ได้นำแบบจำลองของประเทศเยอรมัน สหรัฐอเมริกา และออสเตรเลีย มาใช้ในการพัฒนา เนื่องจากเยอรมันเป็นประเทศแรกที่พัฒนาวิธีการนี้ สหรัฐอเมริกามีมาตรฐานการออกแบบทางที่กรมทางหลวงและหน่วยงานส่วนใหญ่นำมาใช้ และออสเตรเลียมีมาตรฐานการออกแบบทางที่สำนักงานเร่งรัดพัฒนาชนบทนำมาใช้ ซึ่งทั้งกรมทางหลวงและสำนักงานเร่งรัดพัฒนาชนบท เป็นหน่วยงานที่มีก่อสร้างทางเข้าสู่หมู่บ้านในชนบทซึ่งส่วนใหญ่เป็นทางหลวงนอกเมือง 2 ช่องจราจร โดยแบบจำลองความสัมพันธ์ระหว่าง CCR กับ ความเร็ว V85 ของทั้งประเทศเยอรมัน สหรัฐอเมริกา และออสเตรเลีย เป็นไปตามตารางที่ 2.4

ขั้นตอนการคำนวณ

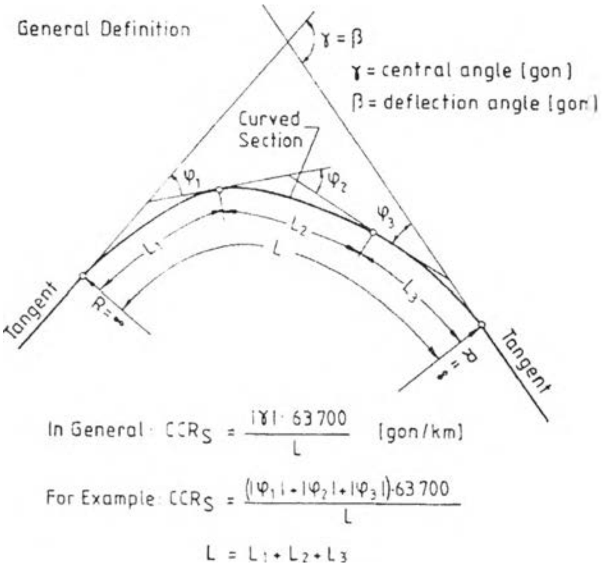
ในขั้นตอนการคำนวณ ผู้วิจัยดำเนินการพัฒนาโปรแกรมจาก AutoLISP โดยตั้งชื่อไฟล์ที่พัฒนาว่า Psp.lsp ประกอบด้วยฟังก์ชัน(defun)หลักย่อยๆ ในไฟล์ ได้แก่ Setting Out Data [SOD] Curvature Change Rate [CCR] Criteria I and Criteria II [ClaCII] Report for Calculation [RFC] และ Speed Profile PC [SPP]

ขั้นตอนที่ 1 จำแนกแนวทางตรงและทางโค้ง (Separate Alignment)

จากข้อมูลแนวเส้นทาง ซึ่งเป็นไฟล์ข้อความที่ได้จากโปรแกรม Softdesk นำมาจำแนกทางโค้งออกจากแนวเส้นทางเป็นส่วนๆ โดยในส่วนโค้งที่มีโค้งต่อเนื่องกันหลายโค้ง เช่นโค้ง Full Spiral และ Compound Curve จะสามารถคำนวณหาค่า CCR ได้เพียง 1 ค่าตามวิธีการของ Ruediger ดังรูปที่ 3.4 โดยรูปแบบคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Setting Out Data [SOD]

ขั้นตอนที่ 2 คำนวณหาค่า CCR (Calculation CCR Value)

คำนวณหาค่า CCR ของแต่ละส่วนโค้งที่ได้ทำการจำแนก โดยใช้ค่ารัศมีโค้ง และความยาวโค้ง คำนวณตามสมการ 2.1 ดังรูปที่ 3.4 โดยรูปแบบคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Curvature Chang Rate [CCR]



รูปที่ 3.4 สมการสำหรับคำนวณหาค่า CCR ของโค้งทั่วไป

ที่มา : Ruediger, Basil, และ Theodor (1999: 8.4)

ขั้นตอนที่ 3 คำนวณหาความเร็ว V85 (Calculation V85)

จากข้อมูล CCR ที่คำนวณได้ในขั้นตอนที่ 2 และแบบจำลองความสัมพันธ์ระหว่าง CCR กับความเร็ว V85 ตามตารางที่ 2.4 สามารถคำนวณหาความเร็ว V85 ของรถยนต์นั่ง ของแต่ละองค์ประกอบได้ โดยรูปแบบคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน CCR

ขั้นตอนที่ 4 ประเมินความปลอดภัย (Safety Evaluation)

เมื่อได้ความเร็ว V85 ของแต่ละองค์ประกอบ สามารถประเมินความปลอดภัยของแนวเส้นทางที่ตรวจสอบ โดยใช้เกณฑ์การประเมิน 3 เกณฑ์ คือ

1) ค่าอัตราการเปลี่ยนแปลงความโค้ง (Curvature Change Rate ; CCR)
2) ความสอดคล้องในการออกแบบ (Design Consistency) คือตรวจสอบความแตกต่างระหว่างความเร็ว V85 กับค่าความเร็วออกแบบ (Vd)
3) ความสอดคล้องของความเร็ว (Speed Consistency) คือตรวจสอบความแตกต่าง ความเร็ว V85 ระหว่างองค์ประกอบที่ต่อเนื่องกัน

ดังรายละเอียดตามตารางที่ 3.2 โดยรูปแบบรหัสคำสั่งที่พัฒนาอยู่ในฟังก์ชัน Criteria I and Criteria II [ClaCII] และ Report for Calculation [RFC]

ตารางที่ 3.2 เกณฑ์การประเมินความปลอดภัยของแนวเส้นทาง จากความเร็ว V85 ของรถยนต์นั่ง ตามวิธีการของ Ruediger

เกณฑ์	ดี	พอใช้	ไม่เหมาะสม
1. CCR (gon/km)	CCR $\leq$ 180	180 < CCR $\leq$ 360	360 < CCR
2. Design Consistency (km/h)	$\lvert V85_i - Vd \rvert \leq 10$	$10 < \lvert V85_i - Vd \rvert \leq 20$	$\lvert V85_i - Vd \rvert > 20$
3. Speed Consistency (km/h)	$\lvert V85_i - V85_{i+1} \rvert \leq 10$	$10 < \lvert V85_i - V85_{i+1} \rvert \leq 20$	$\lvert V85_i - V85_{i+1} \rvert > 20$

ที่มา : Ruediger, Basil, และ Theodor (1999: 11.3)

ขั้นตอนที่ 5 คำนวณหาความเร็วที่ตำแหน่งต่าง ๆบนแนวเส้นทาง (Speed Profile Caculation)

เมื่อมีข้อมูลความเร็ว V85 ของแต่ละองค์ประกอบ แล้วนำมาเขียนเส้นผังความเร็วโดยวิธีการของ Ruediger ซึ่งจะคำนวณหาระยะในแนวทางตรงที่ใช้เปลี่ยนความเร็วจากส่วนโค้งหนึ่งไปยังอีกส่วนโค้งหนึ่ง ตามที่ได้นำเสนอไว้ในบทที่ 2 โดยรูปแบบรหัสคำสั่งที่พัฒนาอยู่ในฟังก์ชัน Speed Profile PC [SPP]

ข้อมูลส่งออก

- **ความเร็วที่ตำแหน่งต่าง ๆบนแนวเส้นทาง (PC Speed Profile)**

เป็นข้อมูลพิกัดความเร็วที่มีการเปลี่ยนแปลงที่ตำแหน่งต่างๆบนแนวเส้นทาง แล้วนำพิกัดดังกล่าวมาพล็อตเป็นผังเส้นความเร็ว โดยรูปแบบรหัสคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Speed Profile PC [SPP]

- **ผลการประเมินความปลอดภัย (Numerical Data for Safety Evaluation)**

Numerical Data for Safety Evaluation เป็นตารางข้อมูลแสดงผลการประเมินความปลอดภัยโดยใช้เกณฑ์การประเมิน 3 ลักษณะ ดังกล่าวในข้างต้น โดยรูปแบบรหัสคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Report for Calculation [RFC]

3.3.2 ความเร็วที่เป็นไปได้ของรถบรรทุก

จากการวิจัยพบว่าความลาดชันของถนนจะมีผลอย่างมากต่อความเร็วของรถบรรทุก ฉะนั้น จึงต้องแบ่งเป็นผลเนื่องจากแนวทางดิ่ง และแนวทางราบ ในส่วนของผลเนื่องจากแนวทางราบนั้นจะใช้ค่าความเร็วที่ตำแหน่งต่างๆบนถนนของรถยนต์นั่งลบด้วย 5 mph (8 km/h) ซึ่งวิธีการหาความเร็วของรถยนต์นั่งนั้นได้นำเสนอไปแล้วในข้างต้น และในส่วนของผลเนื่องจากแนวทางดิ่งจะใช้ข้อมูลความลาดชัน และระยะความลาดชัน ร่วมกับ แผนภูมิความสัมพันธ์ระหว่างระยะทาง และความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน คำนวณหาความเร็วที่ตำแหน่งต่างๆ ดังที่ได้เสนอไว้ในบทที่ 2 แล้วใช้ค่าความเร็วต่ำสุดระหว่างผลกระทบเนื่องจากแนวทางราบและแนวทางดิ่ง เป็นความเร็วของรถบรรทุกที่ตำแหน่งนั้น

ในหัวข้อนี้จะนำเสนอแนวคิดและหลักการทำงานของโปรแกรมหาความเร็วที่เป็นไปได้ของรถบรรทุก เนื่องจากผลกระทบของแนวทางดิ่ง ตามแผนผังแสดงดังรูปที่ 3.5 โดยมีรายละเอียดดังนี้

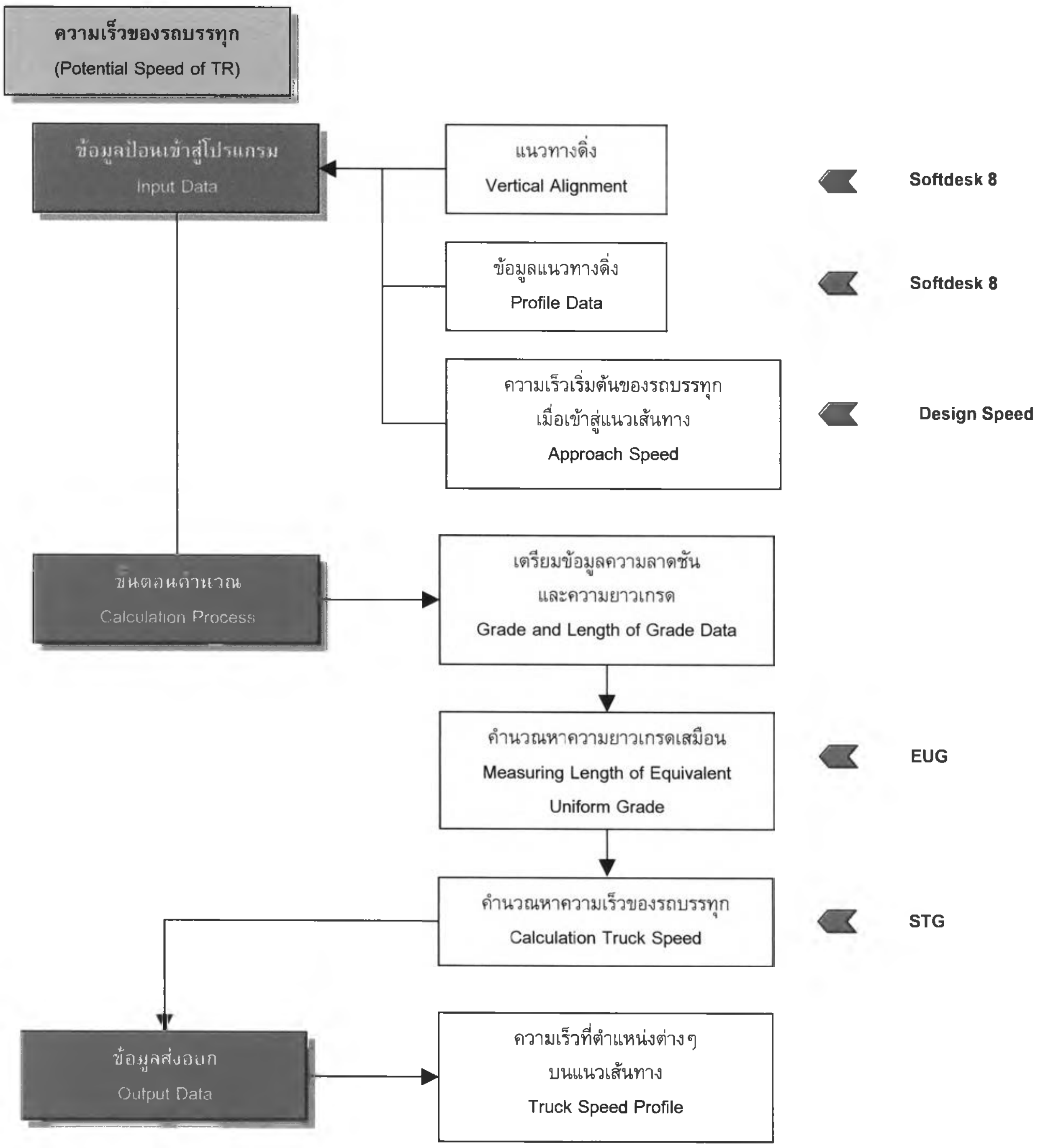
ข้อมูลป้อนเข้าสู่โปรแกรม

- **แนวทางดิ่ง (Vertical Alignment)**

เป็นแนวทางดิ่งที่ต้องการหาความเร็วที่เป็นไปได้ของรถบรรทุก ซึ่งสามารถเตรียมแนวเส้นดังกล่าวได้จากโปรแกรม Softdesk โดยใช้เมนูคำสั่ง Profile อยู่ใน Advanced Design Module เนื่องจากมีคำสั่งที่ช่วยในการสร้างส่วนประกอบต่างๆของแนวทางดิ่งเช่น เส้นความลาดชัน โค้งทางดิ่ง โดยคำนึงถึงอัตราส่วนทางดิ่ง (Vertical Scale) ที่ต้องการนำเสนอในแบบก่อสร้างเพื่อให้เห็นความแตกต่างของค่าระดับที่ตำแหน่งต่างๆบนเส้นทางอย่างชัดเจน เมื่อเตรียมแนวทางดิ่งแล้วเสร็จ ต้องแปลงส่วนของเส้นความลาดชัน และเส้นโค้งดิ่ง ต่างๆ ให้โปรแกรม Softdesk ทราบว่าเส้นต่างๆเหล่านั้นคือแนวทางดิ่งของถนน จากคำสั่ง Define FG Centerline จากเมนู Profile > Vertical Alignments ของโปรแกรม Softdesk

- **ข้อมูลแนวทางดิ่ง (Profile Data)**

เมื่อ Define FG Centerline แล้วเสร็จ โปรแกรม Softdesk จะตรวจสอบคุณสมบัติของเส้นได้แก่ความยาว ความลาดชัน และพารามิเตอร์ของโค้งดิ่ง แปลงเป็นข้อมูลแนวทางดิ่งประกอบด้วยข้อมูล Station Elevation Grade in Grade Out Vertical Curve Information และ Low-Height Station Elevation ซึ่งข้อมูลต่างๆเหล่านี้โปรแกรม Softdesk สามารถส่งข้อมูลออกในรูปไฟล์ข้อความจากคำสั่ง Vertical Alignments > Edit... จากเมนู...



รูปที่ 3.5 แผนผังแสดงแนวคิดและหลักการทำงานของโปรแกรม
ประเมินความปลอดภัยในการออกแบบความเร็วของรถบรรทุก

จากเมนู Profile ในการพัฒนาจะนำข้อมูลระยะทางลาดชัน ความลาดชัน ความยาวโค้งดิ่ง และค่าระดับ ไปคำนวณหาค่าความเร็วของรถบรรทุก จากแผนภูมิความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน

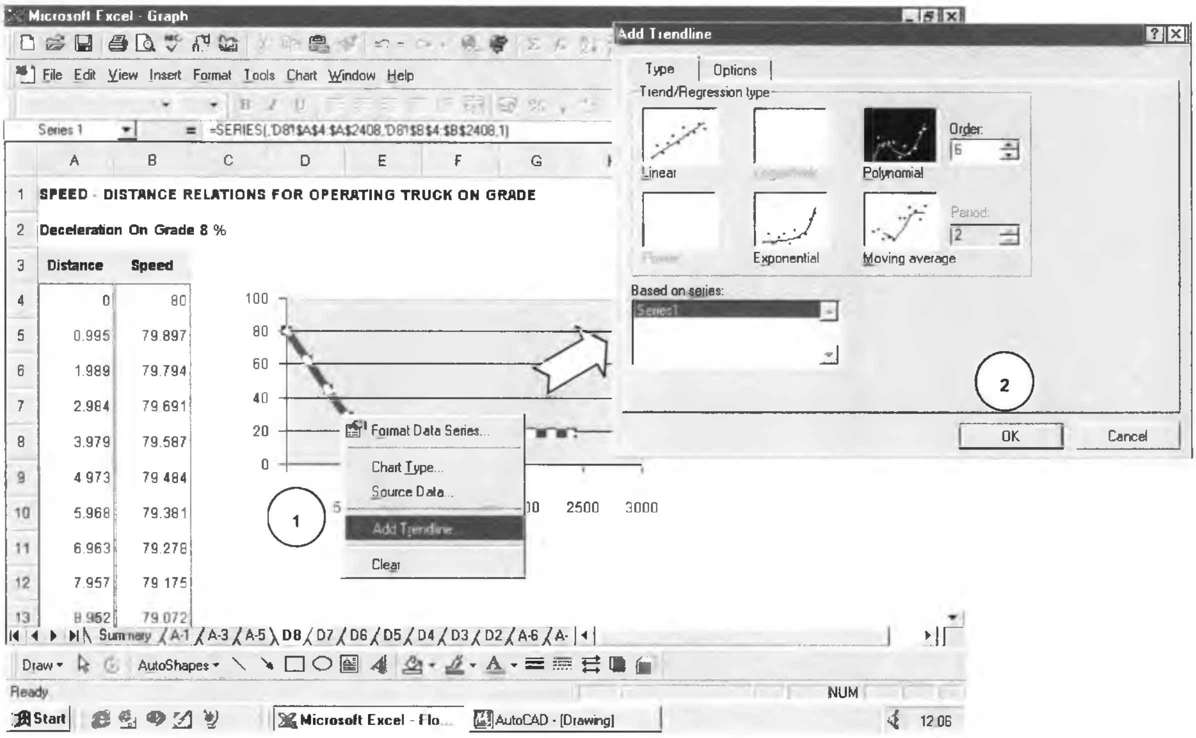
จากแผนภูมิความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน ดังรูปที่ 2.5 ผู้วิจัยได้ดำเนินการแปลงความสัมพันธ์ดังกล่าวมาเป็นสมการทางคณิตศาสตร์ โดยการใช้ฟังก์ชัน Add Trend line ในโปรแกรม Microsoft Excel และเลือกรูปแบบสมการถดถอยเป็นสมการ Polynomial ยกกำลัง 6 ดังแสดงในรูปที่ 3.6 โดยได้สมการแสดงความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน ด้วยค่าความลาดชันต่างๆ เป็น

$$Y = aX^6 + bX^5 + cX^4 + dX^3 + eX^2 + fX + g \qquad 3.1$$

เมื่อ Y = ความเร็วที่เป็นไปได้ของรถบรรทุก km/h.

X = ระยะที่รถบรรทุกวิ่งในทางลาดชัน m.

โดยที่ a, b, c, d, e, f และ g เป็นค่าคงที่ขึ้นกับความลาดชัน แสดงดัง ภาคผนวก ข สัมประสิทธิ์ของสมการความเร็วของรถบรรทุกที่วิ่งบนทางลาดชัน



รูปที่ 3.6 การแปลงความสัมพันธ์เป็นสมการทางคณิตศาสตร์ ด้วยฟังก์ชัน Add Trend line โดยเลือกรูปแบบสมการถดถอยเป็นสมการ Polynomial ยกกำลัง 6

❑ **ความเร็วเริ่มต้นของรถบรรทุกเมื่อเข้าสู่แนวทางดิ่ง (Approach Speed)**

เป็นค่าความเร็วเริ่มต้นของรถบรรทุกเมื่อเข้าสู่แนวทางดิ่ง ซึ่งโดยมากมักกำหนดให้น้อยกว่าความเร็วออกแบบในช่วงเข้าสู่ทางดิ่งนั้นอยู่ประมาณ 5 mph (8 km/h)

ขั้นตอนการคำนวณ

ในขั้นตอนการคำนวณ ผู้วิจัยพัฒนาโปรแกรมจาก AutoLISP โดยตั้งชื่อไฟล์ที่พัฒนาว่า Pst.lsp ประกอบด้วยฟังก์ชัน(defun)หลักย่อยๆ ในไฟล์ ได้แก่ Equivalent Uniform Grades [EUG] และ Speed Trucks on Grades [STG]

ขั้นตอนที่ 1 เตรียมข้อมูลความลาดชัน และความยาวเกรด (Grade and Length of Grade Data)

จากข้อมูล Profile Data ซึ่งเป็นไฟล์ข้อความที่ได้จากโปรแกรม Softdesk นำมาจำแนกข้อมูลเฉพาะข้อมูลระยะความลาดชัน ความลาดชัน ความยาวโค้งดิ่ง และค่าระดับ เพื่อนำไปคำนวณหาค่าความเร็วของรถบรรทุก โดยอาศัยสมการความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชันที่ได้จัดเตรียมไว้

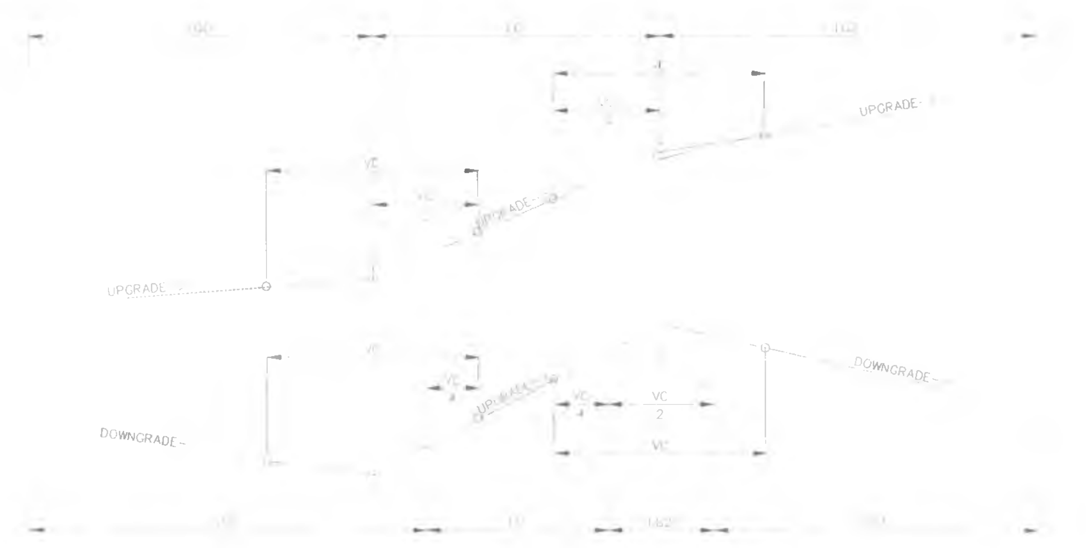
ขั้นตอนที่ 2 คำนวณหาระยะความลาดชันเสมือน (Measuring length of equivalent uniform grade)

Leisch (1977) เสนอวิธีการแปลงโค้งทางดิ่งให้เป็นเพียงเส้นแนวความลาดชัน เพื่อให้สามารถหาความเร็วของรถบรรทุกที่วิ่งผ่านโค้งดิ่งดังกล่าวจากแผนภูมิความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน มีแนวคิดดังนี้

❑ หากผลคูณระหว่างความลาดชันมีค่าเป็น บวก จะแบ่งความยาวโค้งดิ่งออกเป็น 2 ส่วนเท่าๆกันค่าระยะความลาดชันเสมือน (L_G) จะวัดจากจุด PVI ดังรูปที่ 3.7

❑ หากผลคูณระหว่างความลาดชันมีค่าเป็น ลบ จะแบ่งความยาวโค้งดิ่งออกเป็น 4 ส่วนเท่าๆ กันค่าระยะความลาดชันเสมือน (L_G) จะวัดจากจุด PVI ออกไปด้านและ L/4 และจะเกิดระยะความลาดชันสั้นๆ อีกหนึ่งความลาดชัน ดังรูปที่ 3.7

ผู้วิจัยได้นำแนวคิดดังกล่าวมาใช้ในการแปลงโค้งทางดิ่ง จากข้อมูลซึ่งเตรียมไว้ในขั้นตอนที่ 1 ซึ่งจะได้ข้อมูลระยะความลาดชันเสมือน และ ความลาดชันไปใช้ในการคำนวณหาความเร็วของรถบรรทุก โดยรูปแบบคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Equivalent Uniform Grades [EUG]



VC = Vertical Curve Length   LG = Length of Equivalent Uniform Grade

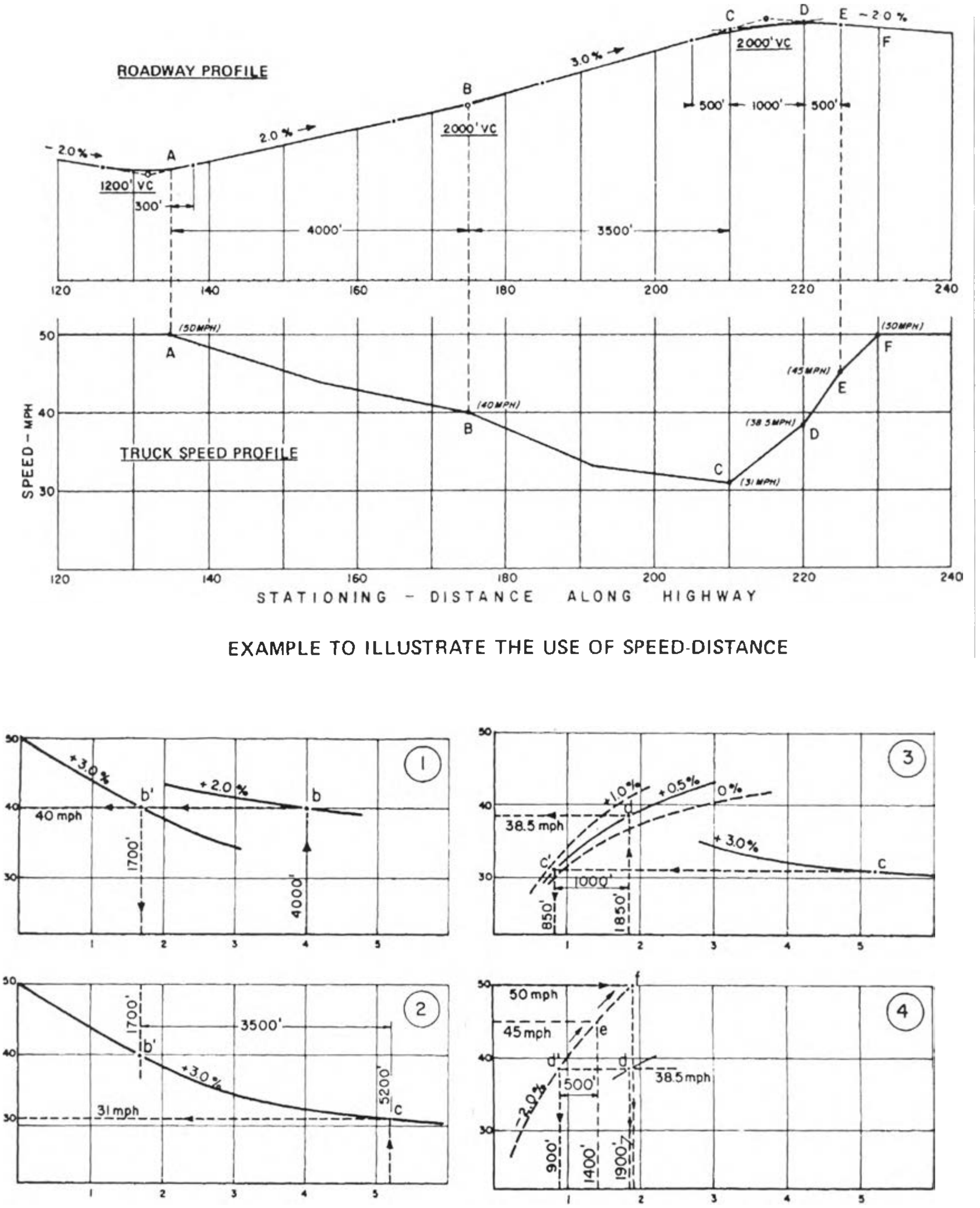
รูปที่ 3.7 แนวคิดในการหาระยะความลาดชันเสมือน โดยวิธีการของ Leisch

ขั้นตอนที่ 3 คำนวณหาความเร็วของรถบรรทุก (Calculation Truck Speed)

จากข้อมูลระยะความลาดชันเสมือน และ ความลาดชัน สามารถนำมาใช้ในการคำนวณหาความเร็วของรถบรรทุกที่วิ่งบนทางลาดชัน จากสมการความสัมพันธ์ระหว่างระยะทางและความเร็วสำหรับรถบรรทุกที่วิ่งบนทางลาดชัน ด้วยค่าความลาดชันต่างๆ ซึ่งความลาดชันจะเป็นตัวกำหนดสมการที่จะเลือกนำมาใช้ หลังจากได้สมการแล้ว แทนค่าระยะความลาดชันเสมือนลงในสมการเพื่อให้ได้ความเร็วของรถบรรทุกเมื่อวิ่งสุดความลาดชันนั้น ซึ่งถ้าหากความลาดชันเป็นตัวเลขไม่ลงตัว จะกำหนดสมการจากความลาดชันที่เป็นตัวเลขลงตัวใกล้เคียงกับทางด้านมากหนึ่งค่าและด้านน้อยหนึ่งค่า นำไปคำนวณหาความเร็วแล้วเทียบบัญญัติไตรยางค์เพื่อให้ได้ความเร็วของรถบรรทุกที่ความลาดชันนั้น

จากรูปที่ 3.8 ในช่วงการเปลี่ยนแปลงความลาดชัน +2.00% ไปยังความลาดชัน +3.00% นั้นจะต้องคำนวณหาระยะที่ใช้วิ่งบนความลาดชัน +3.00% ให้ได้ความเร็วเท่ากับจุด B ที่ 40 mph ก่อนซึ่งมีค่าเท่ากับ 1,700 ฟุต แล้วนำระยะที่ได้ดังกล่าวไปบวกกับระยะความลาดชันเสมือน 3,500 ฟุต เพื่อนำไปคำนวณหาความเร็วของจุด C ที่ 31 mph ต่อไป

ในขั้นตอนที่คำนวณหาระยะในช่วงการเปลี่ยนแปลงความลาดชันนั้นจะต้องนำความเร็วที่จุด B ซึ่งเท่ากับ 40 mph ไปแทนในสมการความสัมพันธ์ของค่าความลาดชัน +3.00% แล้วแก้สมการเพื่อหาระยะทาง ผู้วิจัยนำเอาวิธีการ Newton Ralpson Method มาใช้ โดยรูปแบบรหัสคำสั่งที่พัฒนานั้นอยู่ในฟังก์ชัน Speed Trucks on Grades [STG]



รูปที่ 3.8 ตัวอย่างการคำนวณหาความเร็วของรถบรรทุกที่วิ่งบนทางลาดชัน

ที่มา : Leisch (1977: 32-33)

Newton Ralpson Method

จากสมการ 3.1 ได้

$$Y = g(x) = aX^6 + bX^5 + cX^4 + dX^3 + eX^2 + fX + g$$

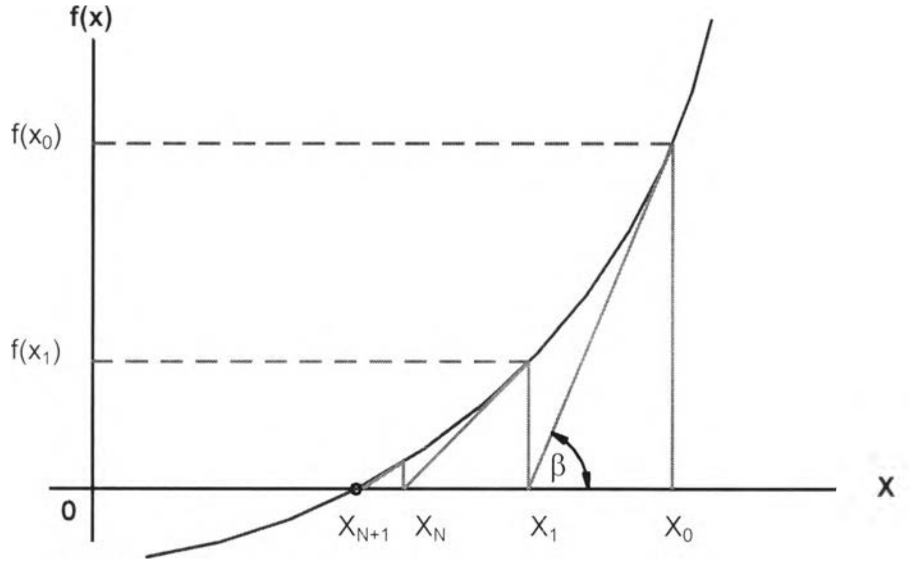
$$f(x) = g(x) - Y = aX^6 + bX^5 + cX^4 + dX^3 + eX^2 + fX + g - Y = 0$$

$$f'(x) = 6aX^5 + 5bX^4 + 4cX^3 + 3dX^2 + 2eX+f \qquad 3.2$$

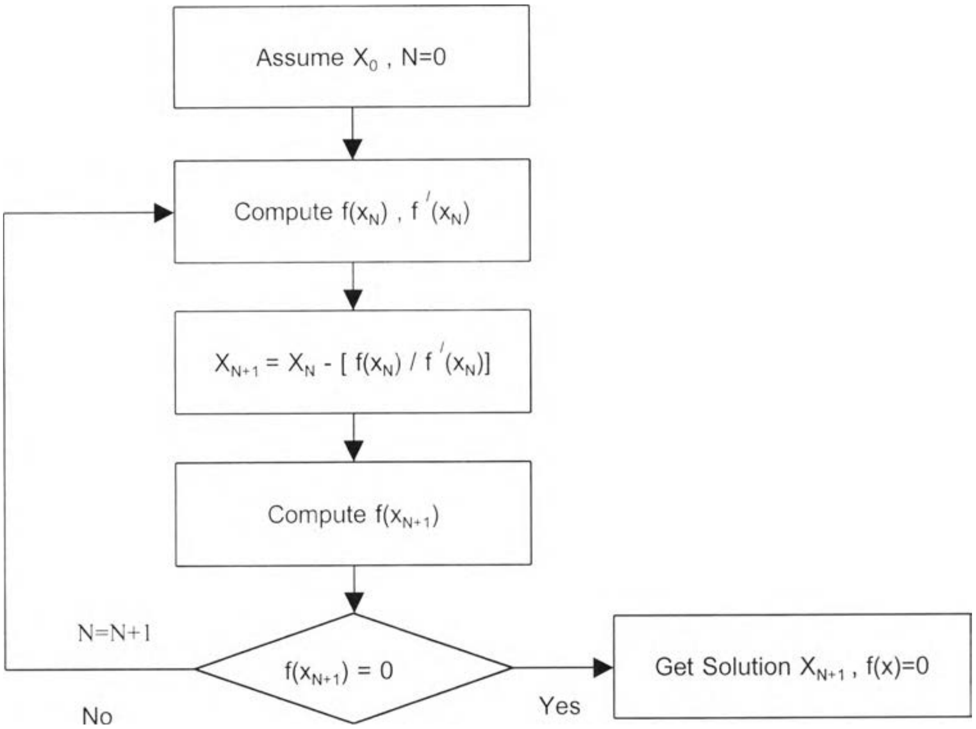
จากรูปที่ 3.9 ได้ $\tan\beta = f'(x_0) = \dfrac{f(x_0)}{X_0 - X_1}$

หรือ $X_1 = X_0 - \dfrac{f(x_0)}{f'(x_0)}$

ฉะนั้น $X_{N+1} = X_N - \dfrac{f(x_N)}{f'(x_N)}$ 3.3



รูปที่ 3.9 การแก้สมการโดยใช้วิธีของ Newton Ralpson Method



รูปที่ 3.10 ผังแสดงวิธีการแก้สมการด้วยวิธี New Ralpson Method

ข้อมูลส่งออก

- **ความเร็วที่ตำแหน่งต่าง ๆบนแนวเส้นทาง (Speed Profile)**

เป็นข้อมูลพิกัดความเร็วที่มีการเปลี่ยนแปลงที่ตำแหน่งต่าง ๆบนแนวเส้นทาง แล้วนำพิกัดดังกล่าวมาพล็อตเป็นผังเส้นความเร็ว

3.4 แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็น

3.4.1 แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็นในแนวราบ

ผู้วิจัยพัฒนาโปรแกรม โดยใช้แนวคิดและหลักการหาระยะมองเห็นในแนวทางราบ ตามมาตรฐาน AASHTO Geometric Design Policy (1994) ซึ่งได้นำเสนอไว้ในบทที่ 2 แสดงดังแผนผังรูปที่ 3.11 โดยมีรายละเอียดดังนี้

ข้อมูลป้อนเข้าสู่โปรแกรม

- **แนวทางราบ (Horizontal Alignment)**

เป็นแนวทางราบที่ต้องการหาระยะมองเห็นในแนวราบ ซึ่งสามารถเตรียมจากโปรแกรม AutoCAD และ Softdesk เมื่อเตรียมแนวทางราบแล้วเสร็จ ต้องแปลงแนวเส้นต่างๆ ให้โปรแกรม Softdesk ทราบว่าแนวเส้นเหล่านั้นคือแนวเส้นทางของถนน จากคำสั่ง Define Alignment จากเมนู Align ของโปรแกรม Softdesk ดังรายละเอียดที่นำเสนอมาแล้วในหัวข้อ 3.3.1

- **ข้อมูลพิกัดบนแนวเส้นทาง (Data Point On Alignment)**

เมื่อ Define Alignment แล้ว สามารถหาค่าพิกัดต่างๆบนแนวเส้นทางจากคำสั่ง Align>Locate Point>Measure Alignment ในเมนู Align ของโปรแกรม Softdesk โดยระยะห่างระหว่างจุดจะเป็นความละเอียดในการคำนวณหาระยะการมองเห็น ในเบื้องต้น AASHTO แนะนำว่ากรณีถ้าระยะมองเห็นน้อยกว่า 500 เมตร ควรตรวจสอบระยะมองเห็นทุกระยะ 10 เมตร กรณีที่มีระยะมองเห็นมากกว่า 500 เมตร อาจจะตรวจสอบทุกระยะ 50 เมตร ผู้วิจัยได้กำหนดระยะตรวจสอบในเบื้องต้นคือทุกระยะ 10 เมตร ซึ่งสามารถเปลี่ยนแปลงค่าได้ตามความเหมาะสม

- **แนวสิ่งกีดขวาง (Obstruction Line)**

เป็นข้อมูลสิ่งปลูกสร้าง แนวเขตทาง หรือ สิ่งบดบังมองเห็น ซึ่งได้จากการสำรวจ หรือภาพถ่ายทางอากาศ ฯลฯ เป็นต้น นำมาเตรียมโดยใช้เฉพาะคำสั่ง Line ในโปรแกรม AutoCAD สร้างลงบน Obstruction Layer

ระยะมองเห็นในแนวราบ
Horizontal Sight Distance

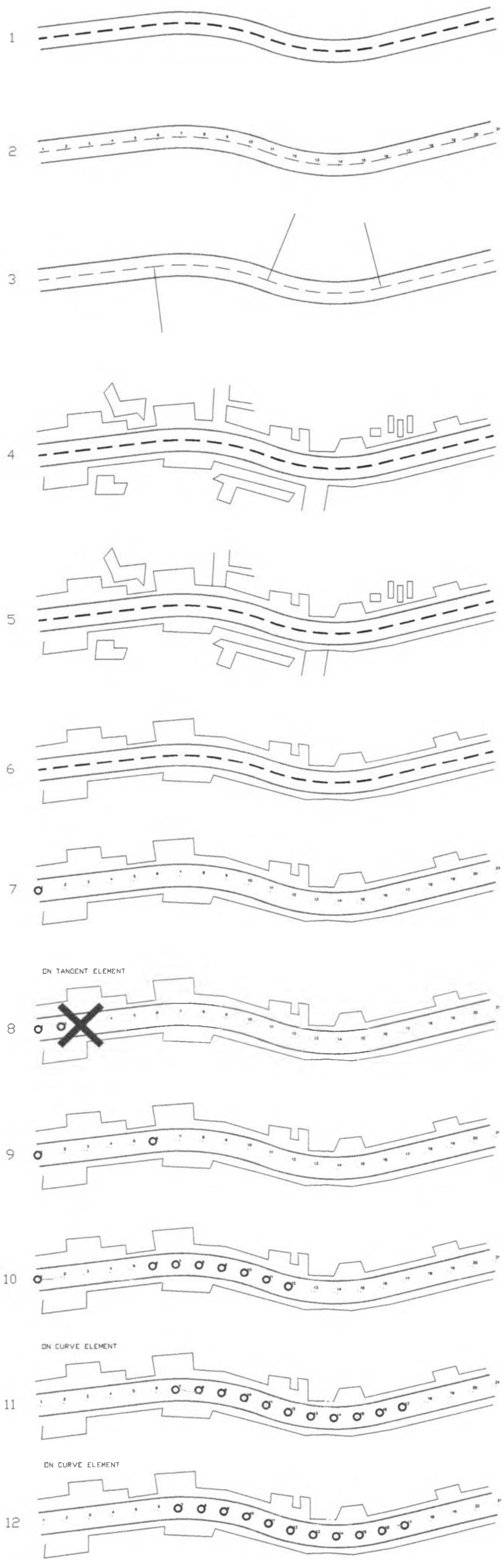
ข้อมูลนำเข้า
Input Data

- แนวเส้นทางราบ / Horizontal Alignment — จาก AutoCAD ได้แก่ส่วนประกอบของแนวเส้นทาง ทางตรง ทางโค้ง (1)
- ข้อมูลพิกัดบนแนวเส้นทาง — จากฟังก์ชันคำสั่งของโปรแกรม Softdesk *Align>Locate Point>Measure Alignment* โดยระยะห่างระหว่างจุดจะเป็นความละเอียดในการคำนวณหาระยะการมองเห็น (2)
- ข้อมูลแนวเส้นทาง / Setting Out Data — เตรียมจาก Softdesk โดยใช้คำสั่ง *Align> Alingment Commands>Edit...* (3)
- แนวสิ่งกีดขวาง / Obstruction Line — จาก AutoCAD โดยใช้ข้อมูลการสำรวจ ภาพถ่ายทางอากาศ สร้างลงบน Obstruction Layer (4) (5) และ (6)

ขั้นตอนคำนวณ
Calculation Process

- จุดหาระยะมองเห็น / Find Point — จากข้อมูลพิกัดบนแนวเส้นทาง หาระยะมองเห็นทุกพิกัด (7)
- จุดตรวจสอบ / Check Point — จากจุดหาระยะมองเห็นตรวจสอบว่าจุดดังกล่าวอยู่ในช่วงทางตรงหรือทางโค้ง จากข้อมูลแนวเส้นทาง
  - จุดหาอยู่ในทางตรง / Find Point On Tangent — จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุด TC หรือ TS Point (8) (9)
  - จุดหาอยู่ในทางโค้ง / Find Point On Curve — จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุด หาระยะมองเห็น (11)
- สร้างเส้นแนวมองเห็น / Create Sight Line — จากจุดหาและจุดตรวจสอบสร้างเส้นแนวมองเห็นระหว่างจุดดังกล่าว (10) (11)
- ตรวจสอบเส้นแนวมองเห็น / Check Sight Line — จากเส้นแนวมองเห็นตรวจสอบว่าแนวดังกล่าวตัดกับแนวสิ่งกีดขวางหรือไม่
  - ไม่ตัดกับสิ่งกีดขวาง / Not Intersect — จะหาจุดตรวจสอบเป็นจุดถัดไปหลังจุดตรวจสอบแรก แล้วดำเนินการตรวจสอบไปเรื่อยๆ จนกระทั่งแนวการมองเห็นตัดกับสิ่งกีดขวาง หรือ สิ้นสุดแนวเส้นทาง
  - ตัดกับสิ่งกีดขวาง / Intersect — จะกำหนดระยะจากจุดหาและจุดตรวจสอบก่อนหน้าเป็นระยะการมองเห็นในแนวราบ และจะเปลี่ยนจุดหาระยะมองเห็นเป็นจุดถัดไปและดำเนินการซ้ำไปเรื่อยๆจนกระทั่งสิ้นสุดแนวเส้น
- ลำดับจุดย้อนกลับ / inverse point — ลำดับจุดย้อนกลับ จากจุดสิ้นสุดโครงการไปยังจุดเริ่มต้นโครงการ และดำเนินการตรวจสอบระยะมองเห็นอีกครั้ง

ข้อมูลนำออก
Output Data — เป็นข้อมูล Text File แสดงข้อมูลจุดหาระยะมองเห็นบนแนวเส้นทางและ ระยะมองเห็นทางราบของจุดนั้น



รูปที่ 3.11 แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็นในแนวราบ

ขั้นตอนคำนวณ

ในขั้นตอนการคำนวณ ผู้วิจัยพัฒนาโปรแกรมจาก AutoLISP โดยตั้งชื่อไฟล์ที่พัฒนาว่า Hsd.lsp

ขั้นตอนที่ 1 สร้างฐานข้อมูลแนวสิ่งกีดขวาง (Obstruction Database)

จากข้อมูลแนวสิ่งกีดขวาง ที่เป็นเส้นตรงบน Obstruction Layer จะใช้ฟังก์ชัน tblsearch ใน AutoLISP ที่ทำหน้าที่ค้นหาสัญลักษณ์ที่ระบุเป็น Obstruction ว่ามีในตารางสัญลักษณ์ระบบ Layer หรือไม่ เพื่อเป็นการตรวจสอบว่ามี Obstruction Layer อยู่ใน File Drawing หรือไม่ หากตรวจสอบแล้วว่ามี จะใช้ฟังก์ชัน ssget ใน AutoLISP ที่ทำหน้าที่รวบรวมรายชื่อ entity ที่ต้องการเข้าเป็นกลุ่มเดียวกัน รวบรวมรายชื่อ entity ที่อยู่ใน Obstruction Layer ต่อจากนั้นจึงดึงคุณสมบัติของ entity ที่รวบรวมไว้ด้วยฟังก์ชัน enget ใน AutoLISP ซึ่งในการวิจัยนี้จะดึงเอาค่าพิกัดจุดต้น และจุดปลาย ของเส้นแนวสิ่งกีดขวางแต่ละเส้นไว้ เป็นฐานข้อมูลแนวสิ่งกีดขวาง

ขั้นตอนที่ 2 กำหนดจุดหาระยะมองเห็น (Find Point)

จากข้อมูลพิกัดบนแนวเส้นทาง จะกำหนดจุดหาระยะมองเห็นเริ่มจากจุดเริ่มต้นโครงการดำเนินการหาระยะมองเห็นของทุกจุด เมื่อคำนวณหาระยะมองเห็นจากจุดเริ่มต้นไปยังจุดสิ้นสุดแล้วเสร็จจะกำหนดจุดเริ่มหาระยะมองเห็นเป็นจุดสิ้นสุดโครงการ และคำนวณระยะมองเห็นกลับไปยังจุดเริ่มต้นโครงการ

ขั้นตอนที่ 3 กำหนดจุดตรวจสอบจากจุดเริ่มต้นไปยังจุดสิ้นสุดแนวเส้นทาง (Check Point From Start Point to End Point)

เมื่อกำหนดจุดหาระยะมองเห็นแล้ว ตรวจสอบว่าจุดดังกล่าวอยู่ในช่วงทางตรงหรือทางโค้ง โดยตรวจสอบจากค่ามุม Azimuth กับจุดข้างเคียง หากพบว่าจุดหาระยะมองเห็นอยู่ในทางตรง จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุด Tangent to Curve (TC) หรือ Tangent to Spiral (TS) แต่หากจุดหาระยะมองเห็นอยู่ในทางโค้ง จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุดหาระยะมองเห็น

ขั้นตอนที่ 4 สร้างเส้นแนวมองเห็น (Create Sight Line)

เมื่อกำหนดจุดหาและจุดตรวจสอบระยะมองเห็นได้แล้วสร้างเส้นแนวมองเห็น (Sight Line) ระหว่างจุดดังกล่าว

ขั้นตอนที่ 5 ตรวจสอบเส้นแนวมองเห็น (Check Sight Line)

จากเส้นแนวมองเห็นตรวจสอบว่าแนวดังกล่าวตัดกับแนวสิ่งกีดขวางหรือไม่ โดยการใช้ฟังก์ชัน inters ใน AutoLISP ที่ทำหน้าที่หาจุดตัด (Intersection Point) ระหว่างเส้นตรง 2 เส้น ซึ่งในที่นี้คือเส้นแนวมองเห็น กับ แนวเส้นซึ่งอยู่ในฐานข้อมูลแนวสิ่งกีดขวาง หากพบว่าเส้นแนวมองเห็นไม่ตัดกับแนวสิ่งกีดขวางจะกำหนดจุดตรวจสอบเป็นจุดถัดไปหลังจุดตรวจสอบแรก แล้วดำเนินการตรวจสอบเช่นนี้ไปเรื่อยๆจนกระทั่งเส้นแนวมองเห็นตัดกับแนวสิ่งกีดขวาง หรือ จนกระทั่งสิ้นสุดแนวเส้นทาง แต่หากพบว่าเส้นแนวมองเห็นตัดกับแนวสิ่งกีดขวางก็จะกำหนดระยะห่างตามแนวเส้นทางระหว่างจุดหาและจุดตรวจสอบก่อนหน้าเป็นระยะการมองเห็นในแนวราบของจุดหาระยะมองเห็นนั้น และจะเปลี่ยนจุดหาระยะมองเห็นเป็นจุดถัดไปและดำเนินการซ้ำไปเรื่อยๆจนกระทั่งสิ้นสุดแนวเส้นทาง

ขั้นตอนที่ 6 จัดเรียงข้อมูลพิกัดบนแนวเส้นทางจากจุดสิ้นสุดไปจุดเริ่มต้น (Sort Data Point From End Point to Start Point)

จากข้อมูลพิกัดบนแนวเส้นทาง จะจัดเรียงข้อมูลพิกัดบนแนวเส้นทางจากจุดสิ้นสุดไปยังจุดเริ่มต้นแนวเส้นทาง เพื่อเป็นข้อมูลในการคำนวณหาระยะมองเห็นในแนวราบจากจุดสิ้นสุดไปยังจุดเริ่มต้นแนวเส้นทาง จากนั้นจะทำการคำนวณในขั้นตอนที่ 2 ถึงขั้นตอนที่ 5 ซ้ำ

ข้อมูลส่งออก

- **ระยะมองเห็นในแนวราบของแนวเส้นทาง (Horizontal Sight Distance Profile)**

เป็นข้อมูลพิกัดระหว่างข้อมูลตำแหน่งจุดหาระยะมองเห็นบนแนวเส้นทาง และระยะมองเห็นในแนวราบของจุดนั้น 2 ชุดข้อมูลคือจากจุดเริ่มต้นไปยังจุดสิ้นสุด และจากจุดสิ้นสุดมายังจุดเริ่มต้นแนวเส้นทาง แล้วนำพิกัดดังกล่าวมาพล็อตเป็นผังเส้นระยะมองเห็นในแนวราบที่ตำแหน่งต่างๆบนแนวเส้นทาง

3.4.2 แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็นในแนวดิ่ง

ผู้วิจัยพัฒนาโปรแกรม โดยใช้แนวคิดและหลักการหาระยะมองเห็นในแนวดิ่ง ตามมาตรฐาน AASHTO Geometric Design Policy (1994) ซึ่งได้นำเสนอไว้ในบทที่ 2 แสดงดังแผนผังรูปที่ 3.12 โดยมีรายละเอียดดังนี้

ข้อมูลป้อนเข้าสู่โปรแกรม

- **แนวทางดิ่ง (Vertical Alignment)**

เป็นแนวทางดิ่งที่ต้องการหาระยะมองเห็นในแนวดิ่ง ซึ่งสามารถเตรียมแนวเส้นดังกล่าวจากโปรแกรม Softdesk เมื่อเตรียมแนวทางดิ่งแล้วเสร็จ ต้องแปลงแนวเส้นต่างๆ ให้โปรแกรม Softdesk ทราบว่าแนวเส้นต่างๆเหล่านั้นคือแนวทางดิ่งของถนน จากคำสั่ง Define FG Centerline จากเมนู Profile > Vertical Alignments ของโปรแกรม Softdesk ดังรายละเอียดที่นำเสนอมาแล้วในหัวข้อ 3.3.2

จากนั้นจึงสร้างแนวทางดิ่งมีระดับความสูงตามจริง กล่าวคือมีอัตราส่วนทางดิ่ง (Vertical Scale) เท่ากับ 1:1 โดยใช้คำสั่ง Profile > Create Profile... และให้จัดเตรียมแนวทางดิ่งดังกล่าวให้อยู่ใน Obstruction Layer เนื่องจากสิ่งบดบังแนวการมองเห็นในแนวทางดิ่ง ก็คือเส้นแนวทางดิ่งนั้นเอง

- **ข้อมูลพิกัดบนแนวทางดิ่ง (Data Point On Vertical Alignment)**

เนื่องจากโปรแกรม Softdesk ไม่มีฟังก์ชันในการหาค่าพิกัดต่างๆบนแนวทางดิ่ง จึงต้อง Define Alignment เส้นแนวทางดิ่งที่มีอัตราส่วนทางดิ่งเท่ากับ 1:1 ให้มีลักษณะเป็นแนวทางราบ โดยใช้คำสั่ง Align > Define from Entities ในเมนู Align จากนั้นจึงเตรียมข้อมูลค่าพิกัดต่างๆบนแนวเส้นทางจากคำสั่ง Align>Locate Point>Measure Alignment ในเมนู Align โดยระยะห่างระหว่างจุดจะเป็นความละเอียดในการคำนวณหาระยะการมองเห็น ผู้วิจัยได้กำหนดระยะตรวจสอบในเบื้องต้นทุกระยะ 10 เมตร ซึ่งสามารถเปลี่ยนแปลงค่าได้ตามความเหมาะสม

- **พารามิเตอร์ที่ใช้ในการตรวจสอบ (Vertical Sight Distance Parameters)**

เป็นการกำหนดคุณสมบัติในการตรวจสอบ ได้แก่ ประเภทของระยะมองเห็น, ความสูงของตาซึ่งขึ้นกับชนิดของรถยนต์ และความสูงของวัตถุ ซึ่งขึ้นกับประเภทของระยะการมองเห็น เป็นต้น เนื่องจากค่าดังกล่าวมีความแตกต่างกันตามแต่ละสถาบัน ดังที่เสนอไว้ในหัวข้อ 2.3.2 ผู้วิจัยจึงกำหนดค่าเบื้องต้นสำหรับค่าดังกล่าวไว้ ตามมาตรฐานของ AASHTO Geometric Design Policy (1994) ดังนี้

ระยะมองเห็นในแนวดิ่ง
Vertical Sight Distance

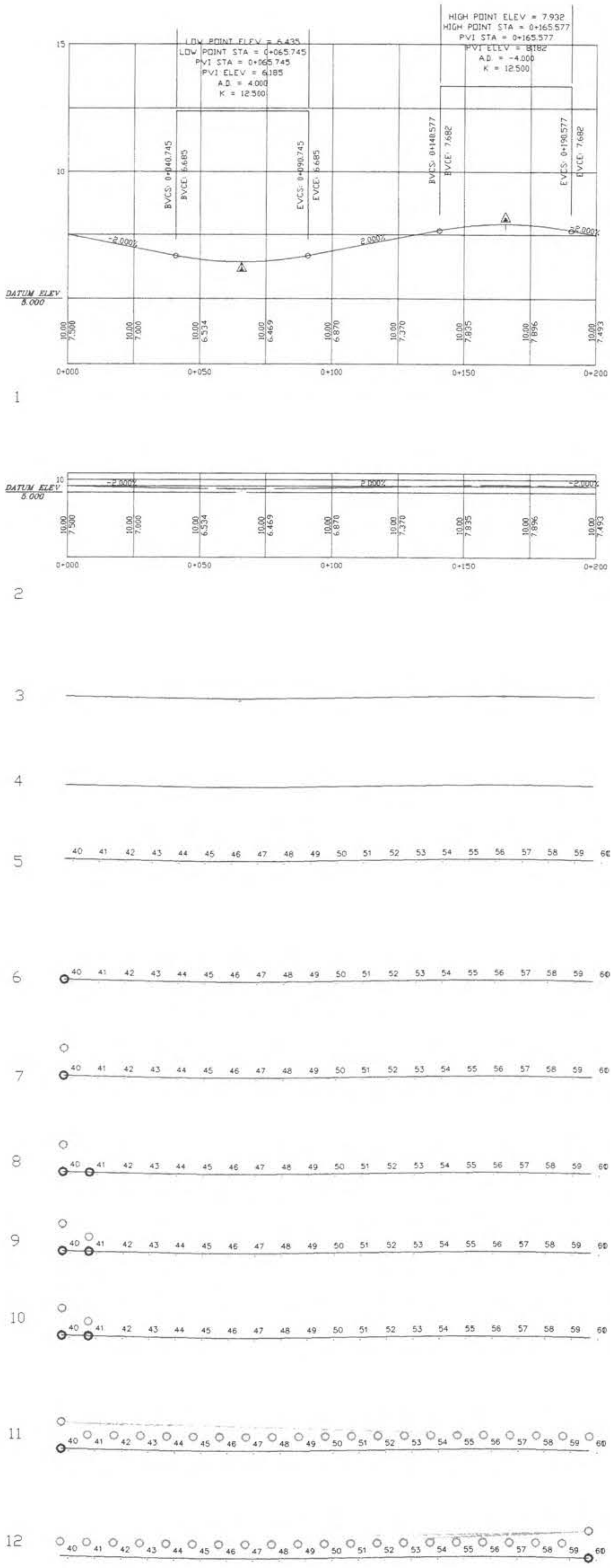
ข้อมูลนำเข้า
Input Data

- แนวเส้นทางดิ่ง
  Vertical Alignment
  — จาก Softdesk โดยใช้คำสั่ง *Profile > Create Profile* โดยแนวทางดิ่งต้องมีระดับความสูงตามจริง (ไม่ถูกขยาย) (1) (2) (3)
- เปลี่ยนแนวเส้นทางดิ่งเป็นแนวเส้นทางราบ
  Change Vertical to
  — จาก Softdesk เนื่องจากต้องการข้อมูลจุดพิกัดบนแนวเส้นทางดิ่ง จากคำสั่ง *Align > Define from Entities* (4)
- ข้อมูลพิกัดบนแนวเส้นทาง
  — จาก Softdesk โดยใช้คำสั่ง*Align>Locate Point>Measure Alignment* โดยระยะห่างระหว่างจุดจะเป็นความละเอียดในการคำนวณหาระยะการมองเห็น (5)
- พารามิเตอร์ที่ใช้ในการตรวจสอบ
  Vertical Sight Distance Parameters
  — ได้แก่ความสูงของตำแหน่งตา ความสูงของวัตถุสำหรับการมองเห็นเพื่อการหยุดที่ปลอดภัย หรือ การแซงที่ปลอดภัย (7) (9)

ขั้นตอนคำนวณ
Calculation Process

- จุดหาระยะมองเห็น
  Find Point
  — จากข้อมูลพิกัดบนแนวเส้นทาง หาระยะมองเห็นทุกพิกัด โดยบวกระยะความสูงของตำแหน่งตา (6) (7)
- ชนิดของการมองเห็น
  Type of Sight Distance
  — กำหนดชนิดของการมองเห็นเพื่อการหยุดหรือแซงที่ปลอดภัย เพื่อนำไปกำหนดความสูงของวัตถุ
- จุดตรวจสอบ
  Check Point
  — จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุด หาระยะมองเห็น โดยบวกระยะความสูงของวัตถุ ขึ้นกับชนิดของระยะมองเห็น (8) (9)
- สร้างเส้นแนวมองเห็น
  Create Sight Line
  — จากจุดหาและจุดตรวจสอบสร้างเส้นแนวมองเห็นระหว่างจุดดังกล่าว (10)
- ตรวจสอบเส้นแนวมองเห็น
  Check Sight Line
  — จากเส้นแนวมองเห็นตรวจสอบว่าแนวดังกล่าวตัดกับแนวเส้นทางดิ่งหรือไม่ (11)
  - ไม่ตัดกับแนวเส้นทางดิ่ง
    — จะหาจุดตรวจสอบเป็นจุดถัดไปหลังจุดตรวจสอบแรก แล้วดำเนินการตรวจสอบไปเรื่อยๆ จนกระทั่งแนวการมองเห็นตัดกับแนวเส้นทางดิ่ง หรือ สิ้นสุดแนวเส้นทาง
  - ตัดกับแนวเส้นทางดิ่ง
    Intersect
    — จะกำหนดระยะจากจุดหาและจุดตรวจสอบก่อนหน้าเป็นระยะการมองเห็นในแนวดิ่ง และจะเปลี่ยนจุดหาระยะมองเห็นเป็นจุดถัดไปและดำเนินการซ้ำไปเรื่อยๆจนกระทั่งสิ้นสุดแนวเส้นทาง
- ลำดับจุดย้อนกลับ
  inverse point
  — ลำดับจุดย้อนกลับ จากจุดสิ้นสุดโครงการไปยังจุดเริ่มต้นโครงการ และดำเนินการตรวจสอบระยะมองเห็นอีกครั้ง

ข้อมูลนำออก
Output Data
— เป็นข้อมูล Text File แสดงข้อมูลจุดหาระยะมองเห็นบนแนวเส้นทางและ ระยะมองเห็นทางดิ่งของจุดนั้น



รูปที่ 3.12 แนวคิดและหลักการทำงานของโปรแกรมหาระยะมองเห็นในแนวดิ่ง

1) ความสูงของตา กำหนดให้มีค่าเท่ากับ 1.07 เมตร (สำหรับรถยนต์นั่ง)

2) ความสูงของวัตถุ สำหรับระยะมองเห็นเพื่อการหยุด กำหนดให้มีค่าเท่ากับ 0.15 เมตร

3) ความสูงของวัตถุ สำหรับระยะมองเห็นเพื่อการแซง กำหนดให้มีค่าเท่ากับ 1.35 เมตร

ซึ่งสามารถเปลี่ยนแปลงค่าได้ตามความเหมาะสม

ขั้นตอนคำนวณ

ในขั้นตอนการคำนวณ ผู้วิจัยพัฒนาโปรแกรมจาก AutoLISP โดยตั้งชื่อไฟล์ที่พัฒนาว่า Vsd.lsp

ขั้นตอนที่ 1 สร้างฐานข้อมูลแนวทางดิ่ง (Obstruction Database)

จากข้อมูลแนวทางดิ่งบน Obstruction Layer จะใช้ฟังก์ชัน tblsearch, ssget และ enget ใน AutoLISP เพื่อสร้างข้อมูลพิกัดจุดต้น และจุดปลาย ของเส้นแนวทางดิ่งแต่ละเส้นไว้ เป็นฐานข้อมูลแนวสิ่งกีดขวาง ดังรายเอียดที่เสนอไว้ในหัวข้อ 3.4.1

ขั้นตอนที่ 2 ชนิดของการมองเห็น (Type of Sight Distance)

กำหนดชนิดของการมองเห็นว่าเป็นระยะมองเห็นเพื่อการหยุดหรือแซงที่ปลอดภัย เพื่อนำไปกำหนดความสูงของวัตถุ

ขั้นตอนที่ 3 กำหนดจุดหาระยะมองเห็นจากจุดเริ่มต้นแนวเส้นทาง (Find Point From Start Point)

จากข้อมูลพิกัดบนแนวทางดิ่ง จะดำเนินการหาระยะมองเห็นของทุกๆ พิกัด โดยนำจุดพิกัดที่เตรียมไว้บวกด้วยความสูงของตาผู้ขับขี่

ขั้นตอนที่ 4 กำหนดจุดตรวจสอบจากจุดเริ่มต้นไปยังจุดสิ้นสุดแนวเส้นทาง (Check Point From Start Point to End Point)

จะกำหนดจุดตรวจสอบระยะมองเห็นจุดแรกเป็นจุดหลังจุดหาระยะมองเห็น โดยบวกระยะความสูงของวัตถุ ขึ้นกับชนิดของระยะมองเห็น โดยเริ่มจากด้านจุดเริ่มต้นแนวเส้นทางไปยังจุดสิ้นสุดแนวเส้นทาง

ขั้นตอนที่ 5 สร้างเส้นแนวมองเห็น (Create Sight Line)

เมื่อกำหนดจุดหาและจุดตรวจสอบระยะมองเห็นได้แล้วสร้างเส้นแนวมองเห็น (Sight Line) ระหว่างจุดดังกล่าว

ขั้นตอนที่ 6 ตรวจสอบเส้นแนวมองเห็น (Check Sight Line)

จากเส้นแนวมองเห็นตรวจสอบว่าแนวดังกล่าวตัดกับแนวทางดิ่งหรือไม่ โดยการใช้ฟังก์ชัน inters ใน AutoLISP ดังรายละเอียดที่นำเสนอไว้ในหัวข้อ 3.4.1 หากพบว่าเส้นแนวมองเห็นไม่ตัดกับแนวทางดิ่งจะกำหนดจุดตรวจสอบเป็นจุดถัดไปหลังจุดตรวจสอบแรก แล้วดำเนินการตรวจสอบเช่นนี้ไปเรื่อยๆจนกระทั่งเส้นแนวมองเห็นตัดกับแนวทางดิ่ง หรือจนกระทั่งสิ้นสุดแนวเส้นทาง แต่หากพบว่าเส้นแนวมองเห็นตัดกับแนวทางดิ่งก็จะกำหนดระยะระหว่างจุดหาและจุดตรวจสอบก่อนหน้าเป็นระยะการมองเห็นในแนวดิ่งของจุดหาระยะมองเห็นนั้น และจะเปลี่ยนจุดหาระยะมองเห็นเป็นจุดถัดไปและดำเนินการซ้ำไปเรื่อยๆจนกระทั่งสิ้นสุดแนวเส้นทางดิ่ง

ขั้นตอนที่ 7 จัดเรียงข้อมูลพิกัดบนแนวเส้นทางจากจุดสิ้นสุดไปจุดเริ่มต้น (Sort Data Point From End Point to Start Point)

จากข้อมูลพิกัดบนแนวเส้นทาง จะจัดเรียงข้อมูลพิกัดบนแนวเส้นทางจากจุดสิ้นสุดไปยังจุดเริ่มต้นแนวเส้นทาง เพื่อเป็นข้อมูลคำนวณหาระยะมองเห็นในแนวดิ่งจากจุดสิ้นสุดไปยังจุดเริ่มต้นแนวเส้นทาง จากนั้นจะทำการคำนวณในขั้นตอนที่ 3 ถึงขั้นตอนที่ 6 ซ้ำ

ข้อมูลนำออก

- **ระยะมองเห็นในแนวดิ่งของแนวเส้นทาง (Vertical Sight Distance Profile)**

เป็นข้อมูลพิกัดระหว่างข้อมูลตำแหน่งจุดหาระยะมองเห็นบนแนวเส้นทางและ ระยะมองเห็นในแนวดิ่งของจุดนั้น 2 ชุดข้อมูลคือจากจุดเริ่มต้นไปยังจุดสิ้นสุด และจากจุดสิ้นสุดมายังจุดเริ่มต้นแนวเส้นทาง แล้วนำพิกัดดังกล่าวมาพล็อตเป็นผังเส้นระยะมองเห็นในแนวดิ่งที่ตำแหน่งต่างๆบนแนวเส้นทาง