บทที่ 4

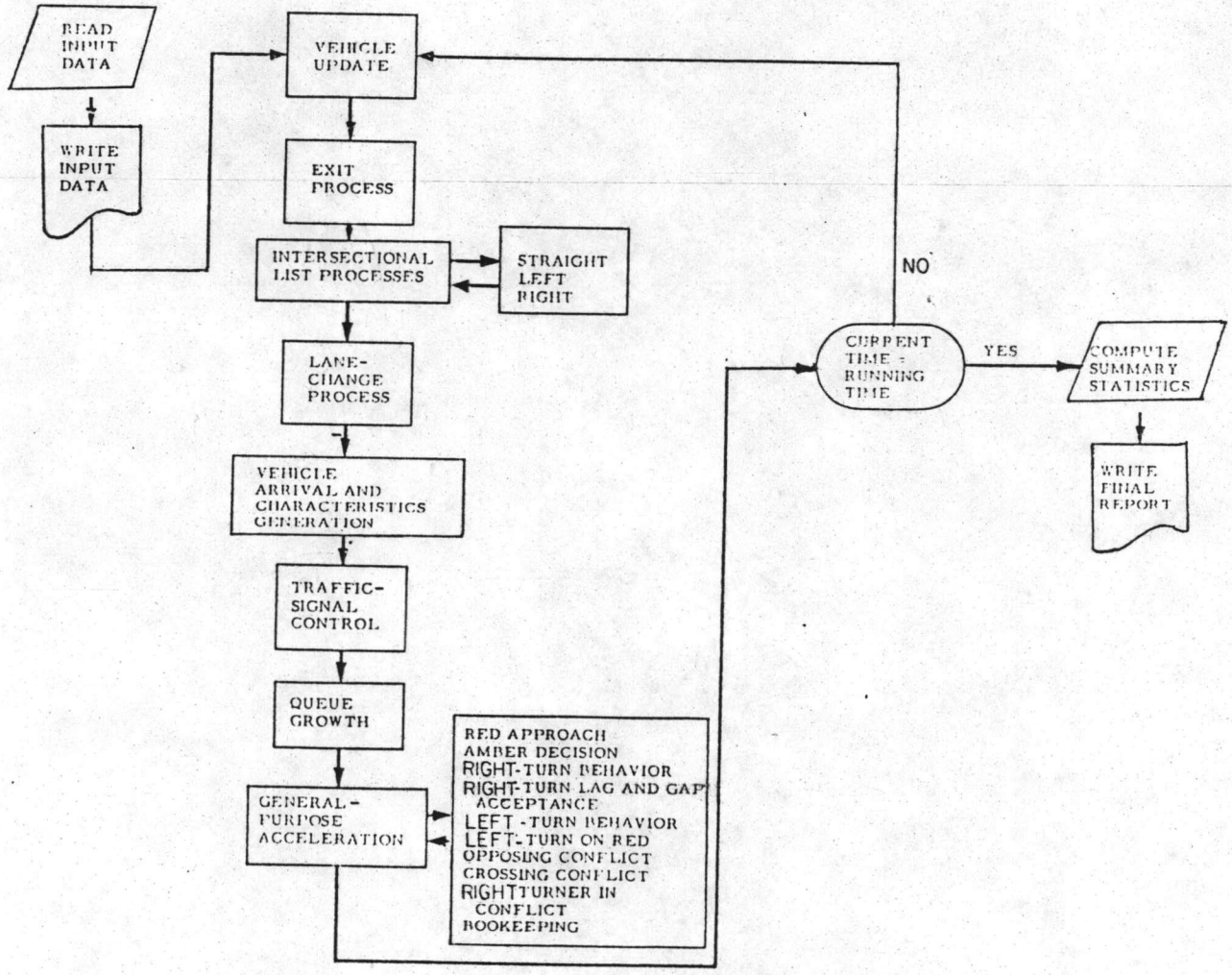
การปรับปรุง Flow Chart ในการสร้างโปรแกรมคอมพิวเตอร์

การปรับปรุงแบบจำลองนี้มีจุดมุ่งหมายเพื่อจำลองสภาพและพฤติกรรมของการจราจรบริเวณทางแยกสัญญาณไฟ เพื่อให้สามารถนำมาใช้ได้อย่างสะดวกทั้งเวลาและค่าใช้จ่าย พฤติกรรมการจราจรบริเวณทางแยกนั้นค่อนข้างซับซ้อนเกินกว่าจะแสดงได้ด้วยการคำนวณแบบธรรมดา Lewis (4) ได้กล่าวไว้ว่าในการปรับปรุง Simulation Model เพื่อให้สมบูรณ์ตามความเป็นจริงทุกอย่าง นอกจากจะยุ่งยากและเสียเวลาแล้ว ยังเป็นไปได้ว่าผลของการทำแบบจำลองดังกล่าวจะไม่เกิดประโยชน์มากนัก ดังนั้นในการทำ Simulation Model ควรคำนึงถึงลักษณะที่สำคัญ ๆ ก็เพียงพอแล้ว พฤติกรรมหรือเหตุการณ์ที่ไม่จำเป็นอาจตัดทิ้งได้บ้าง โดยพฤติกรรมนั้น ๆ น่าจะถูกนำไปร่วมในแบบจำลองอื่น ๆ ได้

จุดมุ่งหมายหลักในการทำวิจัยนี้ก็เพื่อปรับปรุงแบบจำลอง เพื่อศึกษาความล่าช้าของการจราจรบริเวณทางแยก ซึ่งมีสาเหตุมาจากความคับคั่งของการจราจรและความไม่เหมาะสมของจังหวะเวลาสัญญาณไฟ โดยจะไม่เน้นถึงสาเหตุอื่นที่ไม่ก่อให้เกิดความล่าช้าหรือก่อให้เกิดเพียงเล็กน้อย ผู้วิจัยได้เลือกภาษา FORTRAN IV ในการสร้างโปรแกรมคอมพิวเตอร์ ซึ่งสอดคล้องกับสภาพคอมพิวเตอร์ในประเทศปัจจุบัน โปรแกรมคอมพิวเตอร์ที่สร้างขึ้นนี้ผู้วิจัยได้พยายามให้มีรูปแบบที่ยืดหยุ่น ซึ่งสามารถนำมาใช้ทำ Simulation กับสภาพทางแยกหลาย ๆ แบบ เช่น สี่แยก, สามแยก, รถเดินทางเดียว, รถเดิน 2 ทาง เป็นต้น ซึ่งจะต้องแก้ไขตัวโปรแกรมคอมพิวเตอร์ในส่วน Input และ Output เท่านั้น สำหรับ Scanning-Cycle ผู้วิจัยขอแนะนำให้มีค่าอยู่ในช่วง 0.25-0.5 วินาที เนื่องจากว่า ถ้ามีค่าต่ำกว่า 0.25 วินาที โปรแกรมคอมพิวเตอร์จะใช้เวลาในการ Execution มากเกินความจำเป็น และถ้าให้มีค่ามากเกิน 0.5 วินาที จะทำให้ผลของการทำ Simulation ที่ได้ค่อนข้างหยาบเกินไป

4.1 <u>โปรแกรมหลัก</u> (Main Program Chain)

รูป 4.1 เป็นการแสดง Flow Chart ของ Main Program Chain ซึ่งแสดงความสัมพันธ์ของโปรแกรมส่วนต่าง ๆ (Routine) ภายในโปรแกรมหลัก และ



รูปที่ 4.1 ผังภูมิแสดงโปรแกรมหลัก (Main Program Chain)

Subroutine ที่ถูกเรียกใช้จาก Routine สัญญลักษณ์ต่าง ๆ ของผังภูมิ (Flow Chart) ดูภาคผนวก ค. ประกอบ

4.2 <u>การอ่านข้อมูลและตรวจสอบ</u> (Read and Write Input Data)

เป็น Routine แรกเริ่มทำการอ่านข้อมูล ซึ่งจะถูก Input สภาพการจราจรต่าง ๆ ที่จำเป็น จึงเป็นข้อมูลเบื้องต้นของการศึกษาที่ทางแยก รายละเอียดข้อมูลในการ Input ได้กล่าวไว้ในบทที่ 5 หลังจากคอมพิวเตอร์อ่านข้อมูล Input เสร็จ ก็จะพิมพ์ผลการอ่านออกมาในแบบ Form ที่เข้าใจง่าย เพื่อการตรวจสอบข้อมูลการ Input ดูภาคผนวก ก. และ ภาคผนวก จ. ประกอบ

4.3 <u>ลักษณะงานขั้นเริ่มต้น</u>

เป็น Routine ต่อจาก Read and Write Input Data โดยจะทำหน้าที่ในการ Set ค่าต่าง ๆ ก่อนที่จะทำการ Simulation ซึ่งพอจะแบ่งการทำงานออกได้ ดังนี้

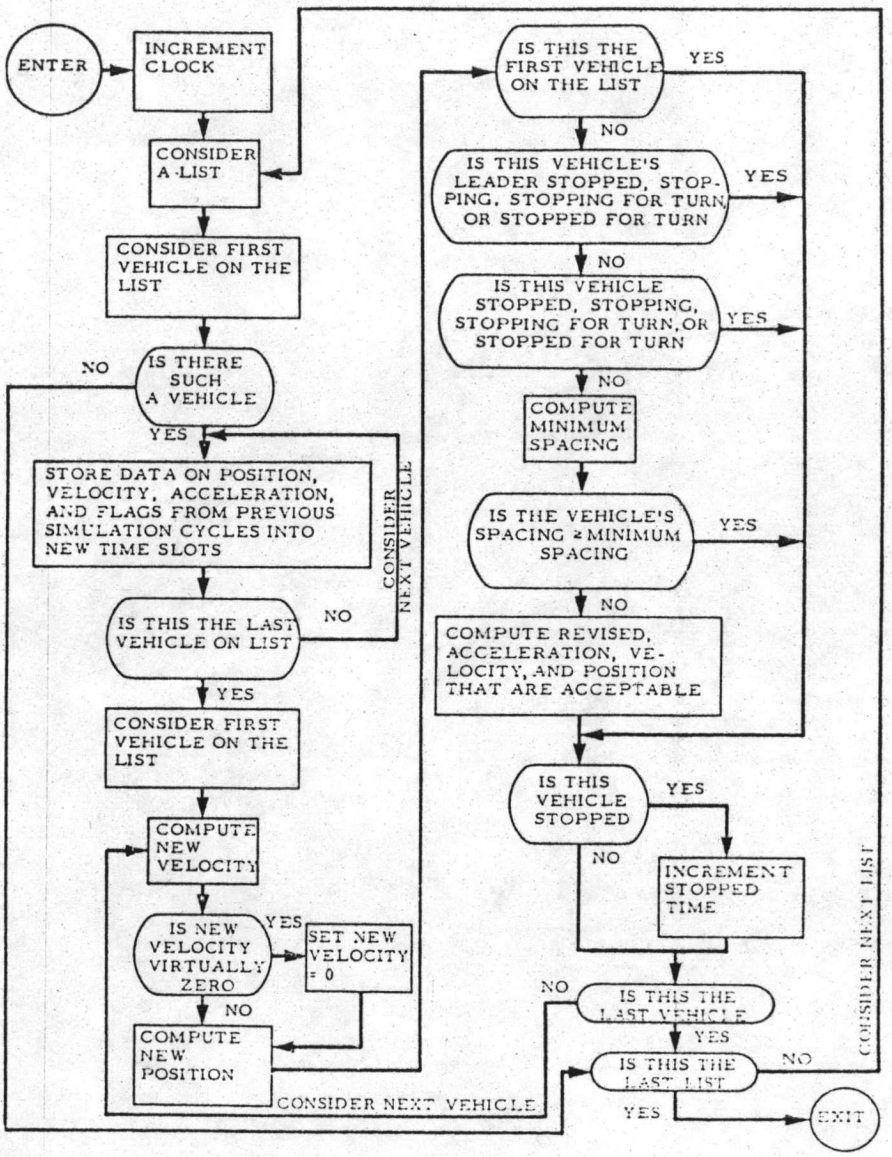
1. คำนวณเกี่ยวกับค่าทางเรขาคณิตต่าง ๆ ของทางแยก เช่น start-of-turn points, dischart boundary ในการ transform และความเร็วสูงสุดที่จะเลี้ยวได้ เป็นต้น

2. กำหนดค่าต่าง ๆ ในการเริ่มต้นการ Simulation รวมทั้งกำหนดค่า Flags words เป็นต้น

3. แปลงค่าจากหน่วยในการ Input มาเป็นหน่วยในการ Simulation เช่น Reaction time ฯลฯ

4.4 <u>ความคืบหน้าของยวดยาน</u> (Vehicle update)

Routine นี้มีหน้าที่ในการเปลี่ยนแปลงความเร็ว และตำแหน่งของยวดยานให้สอดคล้องกับผลของความเร่งที่ได้จาก General Purpose Acceleration โดยจะ update ทุก ๆ Scanning พร้อมทั้งรวบรวมข้อมูลของเวลาที่รถหยุด เพื่อส่งต่อให้ส่วน Output ในการคำนวณและแสดงผลทางสถิติ ดังแสดงในรูป 4.2



รูปที่ 4.2 ผังภูมิแสดงการทำงานในส่วนความคืบหน้าของยวดยาน (Vehicle Update)

4.5 <u>การออกจากระบบ</u> (Exit Process)

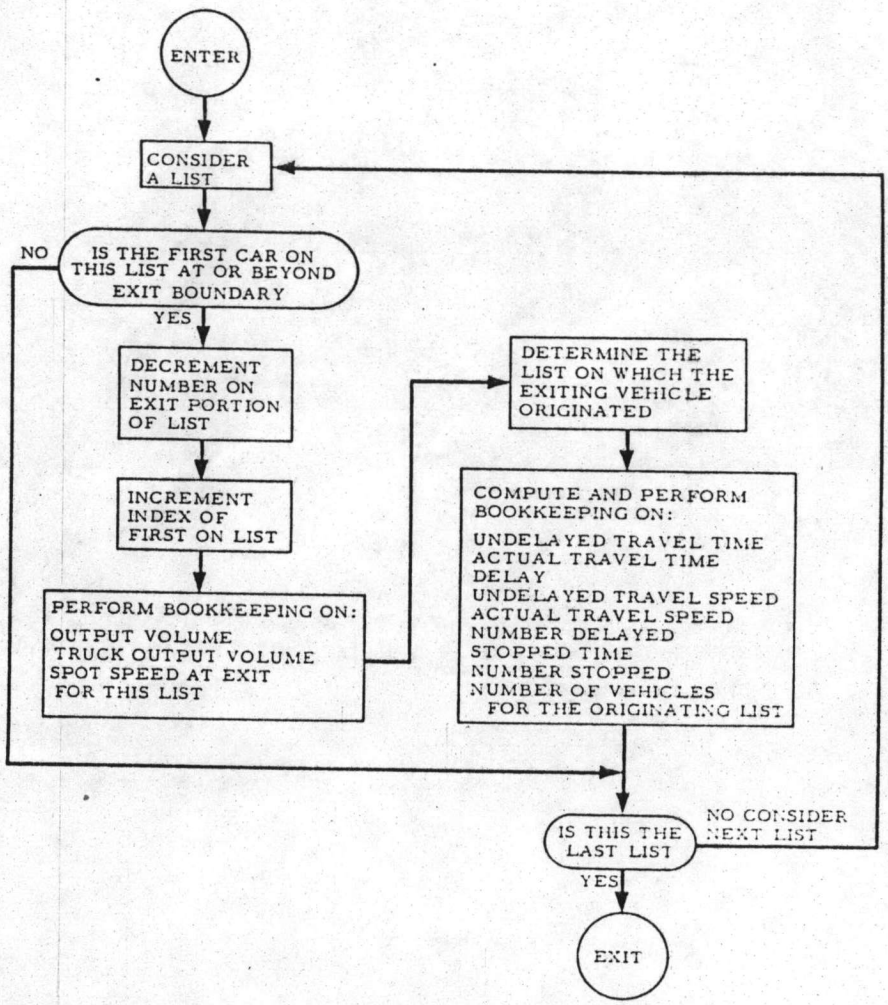
เป็น Routine ที่ใช้ตรวจสอบการออกจากระบบของยวดยาน โดยเปรียบเทียบตำแหน่งของรถกับจุด Exit Boundary ทุก ๆ Scanning โดยจะพิจารณาเฉพาะรถคันแรกของแต่ละช่องจราจร เมื่อรถออกจากระบบแล้วจะถือว่ารถดังกล่าวพ้นจากการพิจารณา และโปรแกรมส่วนนี้จะรวบรวมข้อมูลของรถที่ออกจากระบบ เพื่อส่งต่อให้ส่วน Output ในการคำนวณและแสดงผลทางสถิติเกี่ยวกับค่าของ Travel times, Travel speeds พร้อมทั้งจะเพิ่มค่า NFST(k) และลดค่า NE(k) ดังแสดงเป็นผังภูมิในรูป 4.3

4.6 <u>การเข้าสู่ทางแยกและเข้าช่องทางจราจร</u> (Entry to Intersectional List Process Routines)

หน้าที่ของ Routine นี้ คือ การพิจารณาเกี่ยวกับทิศทางของการเคลื่อนที่ของยวดยานและการเข้าสู่ทางแยก โดยจะพิจารณาเฉพาะรถคันแรกของ Approach portion ซึ่งสามารถรู้ได้ว่ารถเข้าสู่ทางแยกโดยการเปรียบเทียบตำแหน่งของรถกับ Enter Intersection Line ทุก ๆ Scanning เมื่อรถวิ่งผ่าน Enter Intersection Line จะถือว่ารถอยู่ในส่วนของ Exit Portion ในกรณีที่เป็นรถเลี้ยว โปรแกรมส่วนนี้จะติดต่อกับ Subroutine Move เพื่อย้ายข้อมูลของรถจาก Origin List ไปสู่ Receiver List ค่า Index และหมายเลขประจำรถ (J) จะถูกเปลี่ยนแปลง ดังแสดงเป็นผังภูมิในรูป 4.4, 4.5, 4.6, และ 4.7

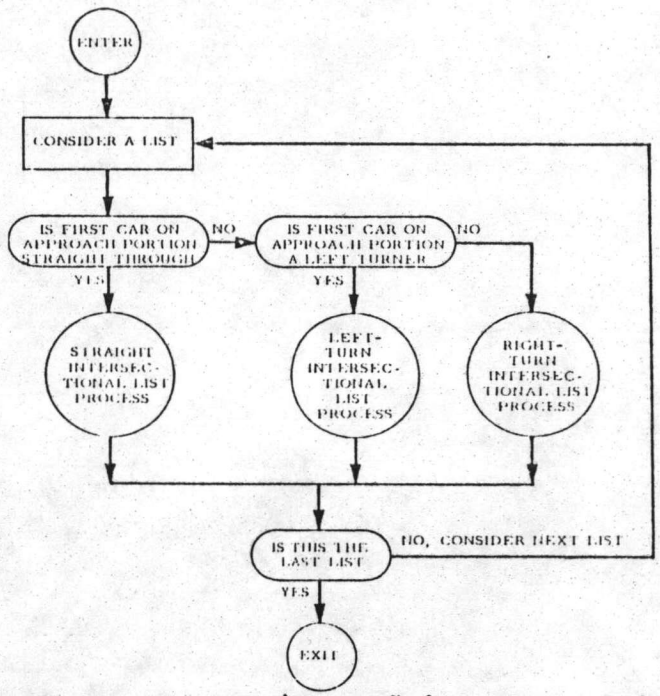
4.7 <u>การเปลี่ยนช่องจราจร</u> (Lane Change Process)

ในช่องจราจรที่อนุญาติให้มีรถตรงไปและเลี้ยวขวาได้ เมื่อรถคันแรกใน Approach portion จอดรอเพื่อเลี้ยวขวา ซึ่งจะทำให้รถคันหลังที่จะตรงไปจำเป็นต้องหยุดคอยไปด้วย โดยพฤติกรรมจริงแล้วรถคันหลังจะพยายามเปลี่ยนช่องจราจรเมื่อช่องจราจรด้านข้างมีช่องว่าง (Gap) เพียงพอ ในส่วน Routine นี้ จึงมีหน้าที่ในการตรวจสอบความเป็นไปได้ในการเปลี่ยนช่องจราจรของรถที่มีเงื่อนไขดังที่กล่าวมาแล้ว โดยการคำนวณเวลาจากตำแหน่งของรถคันหลังที่ใกล้ที่สุดในช่องจราจรข้างเคียง (Receiver List) ถึงตำแหน่งที่

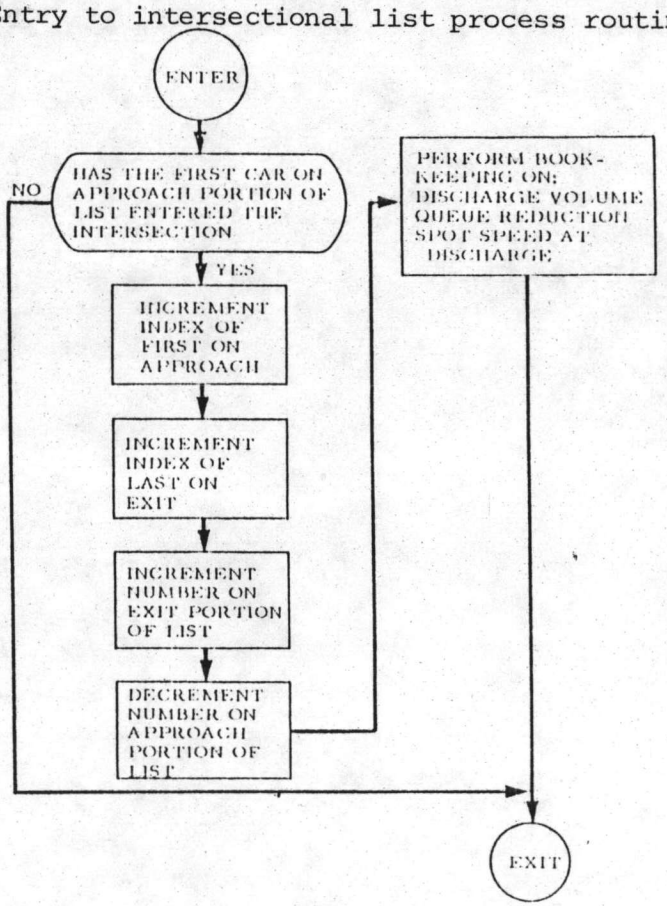


รูปที่ 4.3 ผังภูมิแสดงการทำงานในส่วนการออกจากระบบ

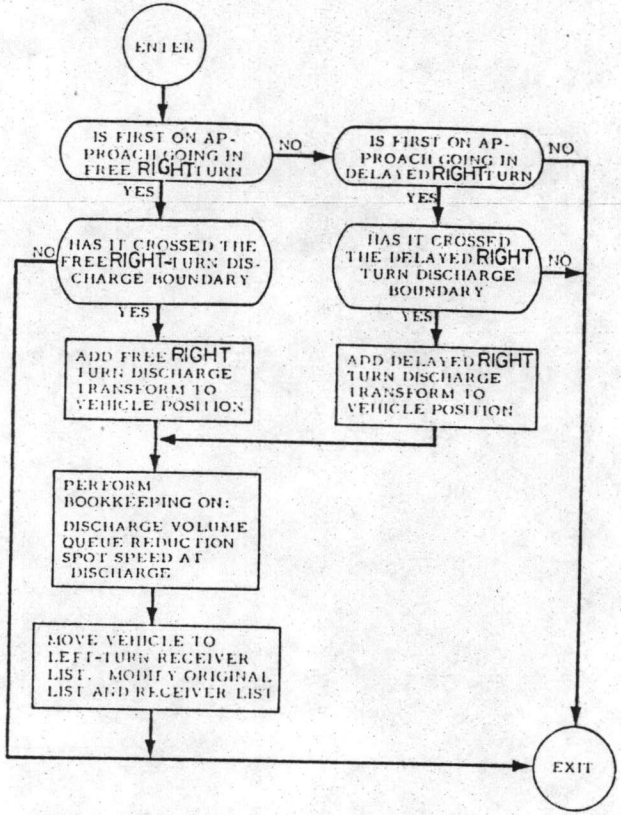
(Exit Process)



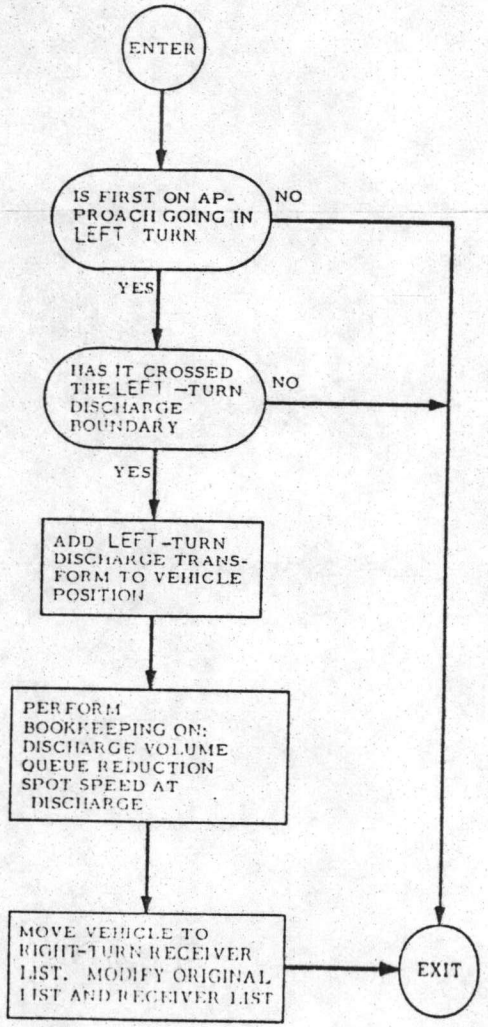
รูปที่ 4.4 ผังภูมิแสดงการทำงานในส่วนการเข้าสู่ทางแยกและเข้าช่องจราจร
(Entry to intersectional list process routines)



รูป 4.5 ผังภูมิแสดงการทำงานในส่วน Intersectional list process for straight-through vehicles



รูป 4.6 ผังภูมิแสดงการทำงานในส่วน Right-Turn Intersectional List Process



รูป 4.7 ผังภูมิแสดงการทำงานในส่วน Left-Turn Intersectional List Process

รถจะแทรกตัวเข้าไป พร้อมทั้ง Random เปรียบเทียบกับค่าใน Lane Change Probability Table ในตาราง 3.6 เมื่อมีการตัดสินใจที่จะเปลี่ยนช่องจราจรรถดังกล่าวจะแทรกตัวเข้ามาในช่องจราจรข้างเคียง และย้ายข้อมูลของรถที่เปลี่ยนช่องจราจรจาก Original list ไปสู่ Receiver list ดังแสดงเป็นผังภูมิในรูป 4.8

4.8 <u>การ Generate รถและคุณลักษณะบางประการขณะเข้ามาสู่ระบบ</u> (Vehicle Arrival and Characteristics Generation)

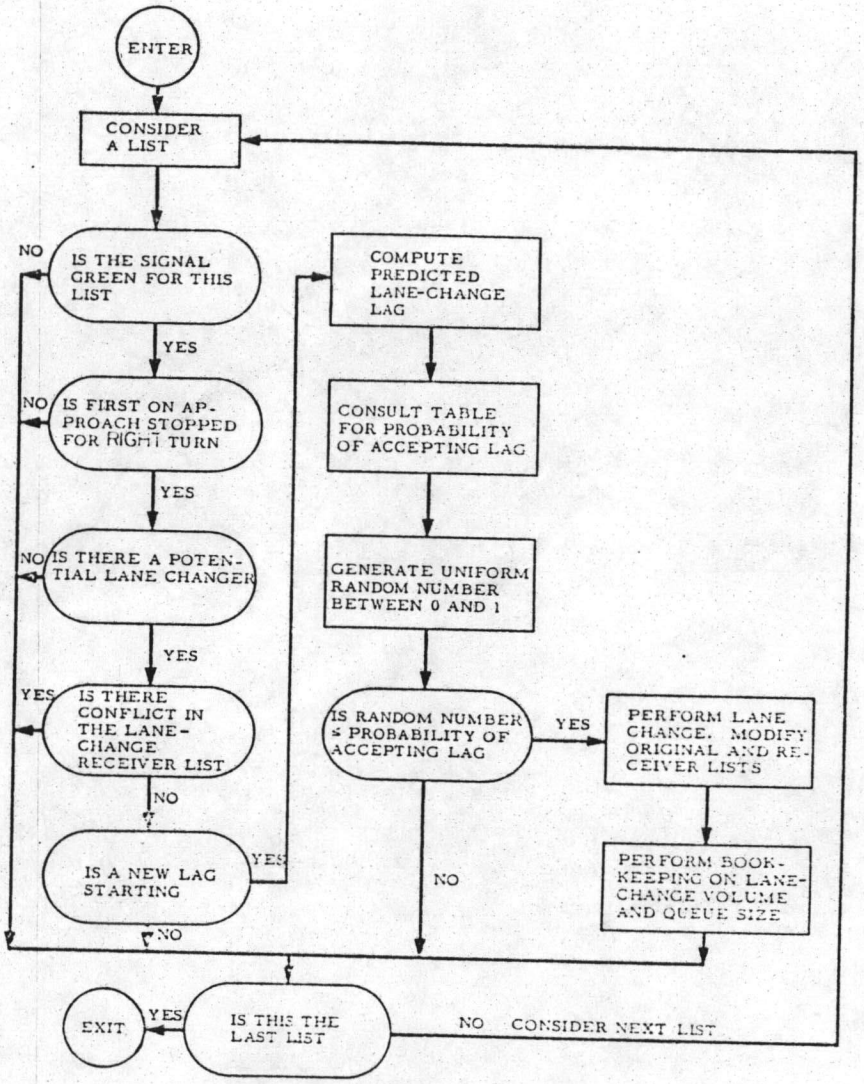
รูป 4.9 เป็นการแสดงผังภูมิของ Routine ส่วนนี้ ซึ่งจะทำหน้าที่ในการ generate รถเข้าสู่ระบบ โดยการคำนวณหา Headway จาก Probability Distribution ของการ Arrival ของแต่ละช่องจราจร ซึ่งสามารถเก็บข้อมูลได้จากสนาม นอกจากนี้ Routine ส่วนนี้ยังมีหน้าที่ในการ generate ค่าเริ่มต้นของรถที่เริ่มเข้าสู่ระบบ เช่น Direction of movement, Traget Velocity ฯลฯ

4.9 <u>การควบคุมสัญญาณไฟที่ทางแยก</u> (Traffic-Signal Control)

Subprogram ส่วนนี้เป็น Routine ในการจัดสัญญาณไฟแบบ Fix-time signal ซึ่งจะแบ่งสัญญาณไฟออกเป็น Phase ต่าง ๆ การเปลี่ยน Phase ของสัญญาณไฟสามารถทำได้ง่าย จึงเป็นการสะดวกในการทดสอบจังหวะเวลา รอบเวลา และแบบต่าง ๆ ของสัญญาณไฟ ซึ่งจะทำให้สามารถทราบถึงแบบของสัญญาณไฟที่ดีที่สุด ดูส่วน Input และผังภูมิในรูป 4.10 ประกอบ

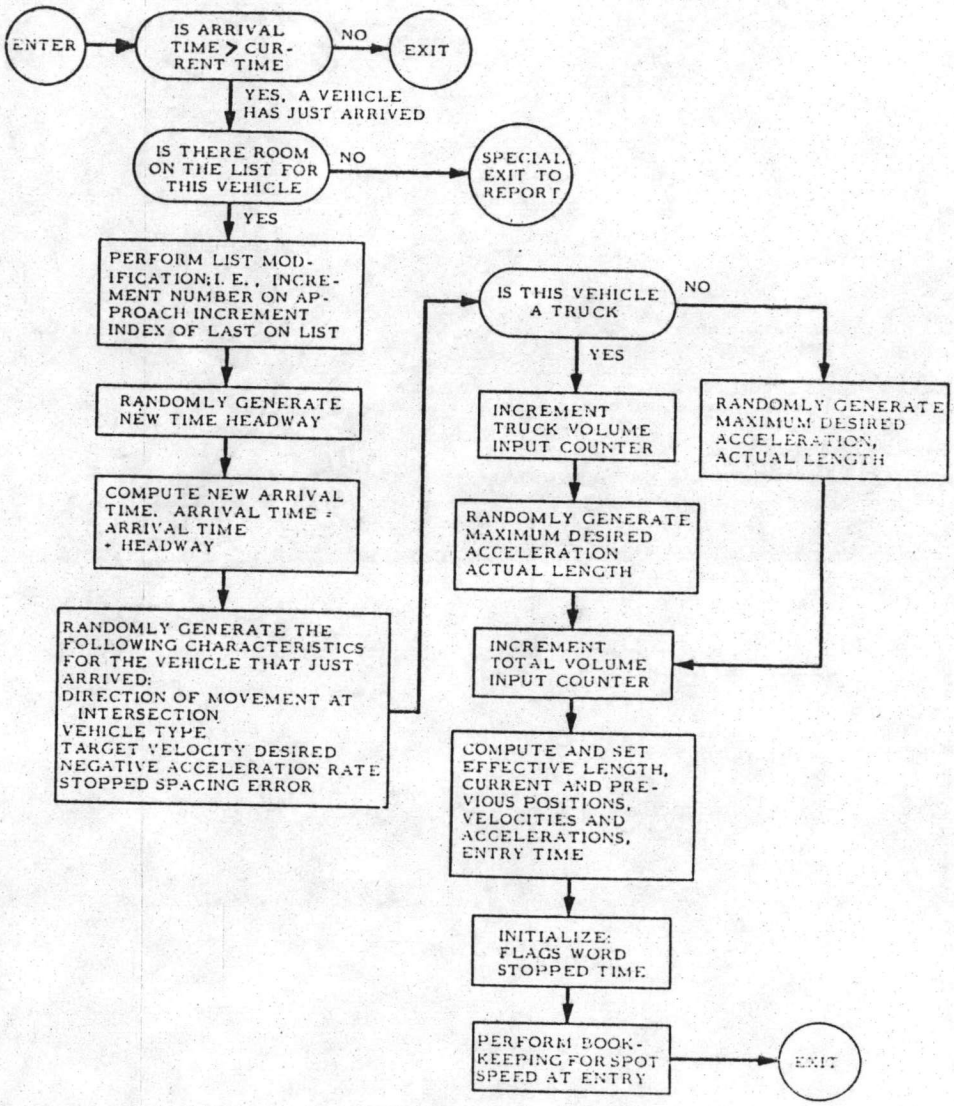
4.10 <u>การนับ Queue</u> (Queue Growth)

รูป 4.11 เป็นการแสดงผังภูมิ แสดงการทำงานของ Routine นี้ ซึ่งจะทำหน้าที่ตรวจเช็คจำนวน Queue ของรถที่จอดบริเวณทางแยก เนื่องจากสัญญาณไฟแดง เมื่อมีรถจอดเพิ่มขึ้นจำนวนฯ Queue size ก็จะเพิ่มขึ้น พร้อมกับรวบรวมข้อมูลของจำนวน Queue size เพื่อส่งต่อให้ส่วน Output ในการคำนวณและแสดงผลทางสถิติ

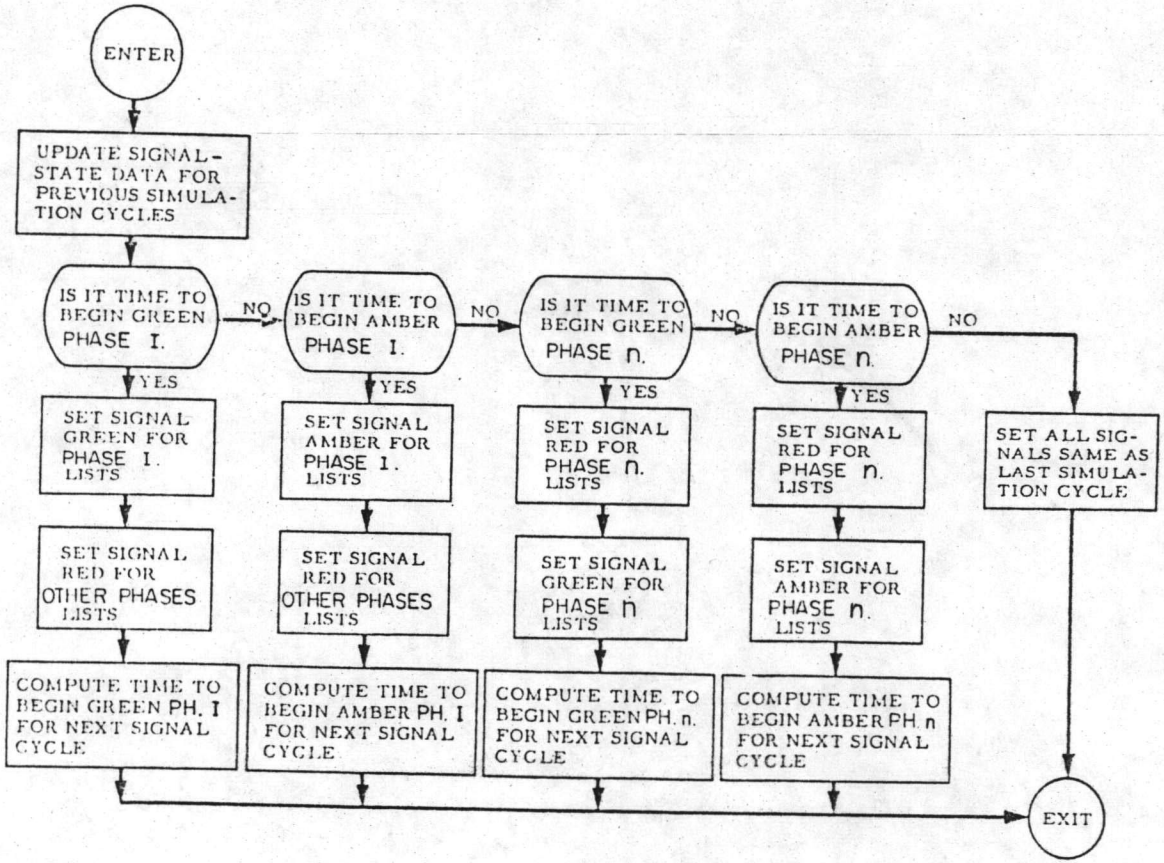


รูปที่ 4.8 ผังภูมิแสดงการทำงานในส่วนการเปลี่ยนช่องจราจร

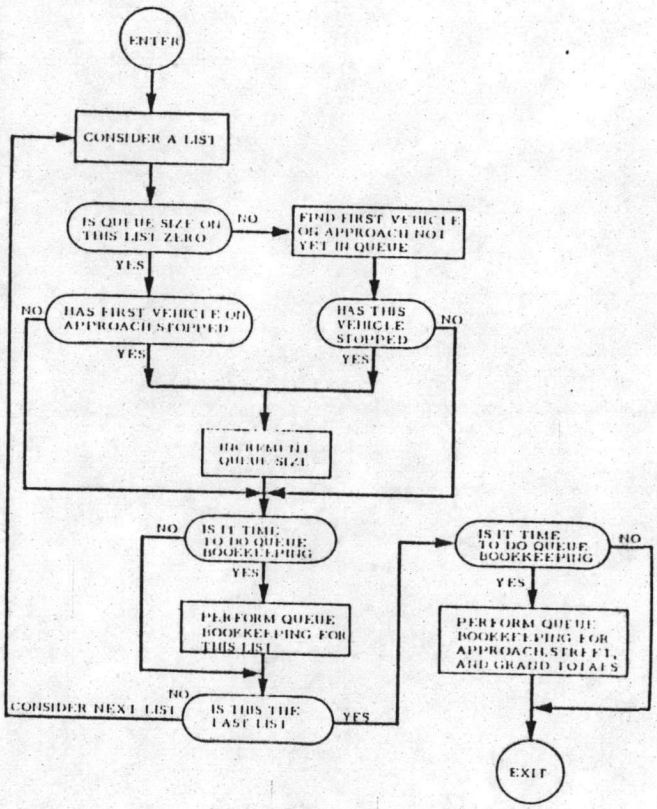
(Lane-Change Process)



รูปที่ 4.9 ผังภูมิแสดงการทำงานในส่วนการ Generate รถและคุณลักษณะบางประการขณะเข้ามาสู่ระบบ (Vehicle-Arrival and Characteristics Generation)



รูป 4.10 ผังภูมิแสดงการทำงานในส่วนการควบคุมสัญญาณไฟที่ทางแยก (Fixed-Time Traffic-Signal Control)



รูปที่ 4.11 ผังภูมิการทำงานในส่วนการนับ Queue

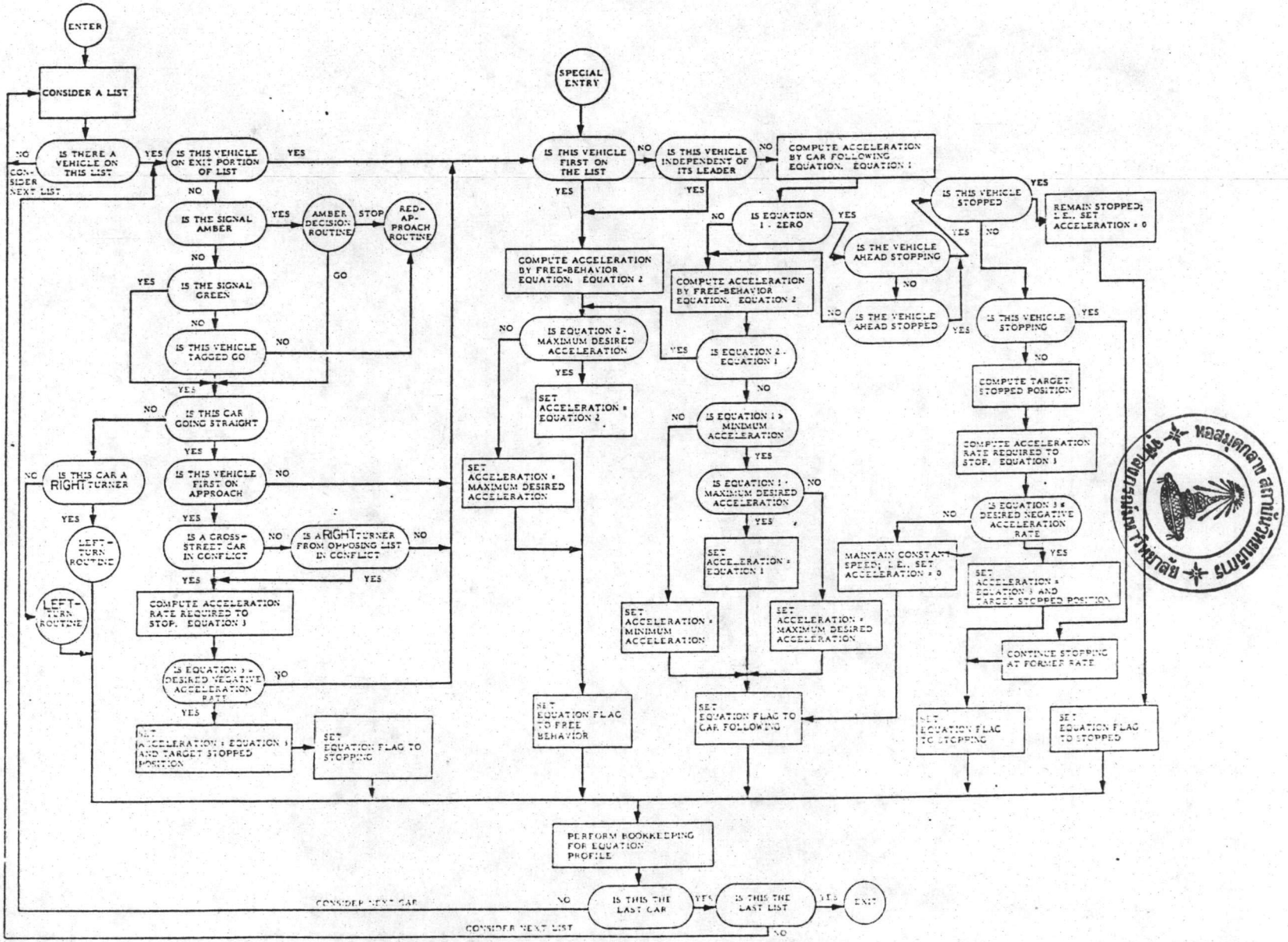
(Queue Growth)

4.11 ความเร่งและความเร็ว (General Purpose Acceleration)

Subprogram ส่วนนี้เป็น Routine ที่ใช้คำนวณหาความเร่งของยวดยานที่เกิดจากพฤติกรรมของผู้ขับขี่ที่ได้รับอิทธิพลจากสิ่งแวดล้อม เช่น สัญญาณไฟ เป็นต้น Routine ส่วนนี้จะทำงานสัมพันธ์ร่วมกับ Routine และ Sub-routine อื่น ๆ ดังแสดงไว้ในผังภูมิในรูป 4.1 และ 4.12 โปรแกรมส่วนนี้จะพิจารณาพฤติกรรมของรถเมื่อถึงเวลาของ Reaction time (T) โดยเริ่มจากรถคันแรก (NFST) ของช่องจราจรในส่วน Exit Portion และถ้าเป็นสัญญาณไฟเขียวการพิจารณาของโปรแกรมก็จะดำเนินต่อไปจนถึงคันสุดท้ายของช่องจราจร (NLST) แล้วจึงเปลี่ยนไปพิจารณาช่องจราจรต่อไป ในระหว่างการพิจารณาพฤติกรรมของรถที่อยู่ในช่อง Approach Portion ที่มีสัญญาณไฟเขียว ถ้าเป็นรถเลี้ยวการพิจารณาจะถูกส่งต่อไปพิจารณาที่ Routine พฤติกรรมการเลี้ยวขวา (Right Turn Behavior) หรือ Routine พฤติกรรมการเลี้ยวซ้าย (Turn Behavior Routine) เมื่อพิจารณาพฤติกรรมและคำนวณความเร่งของรถคันดังกล่าวเสร็จสิ้น การพิจารณาพฤติกรรมของรถก็จะถูกส่งกลับมาที่โปรแกรมความเร็วและความเร่ง (General Purpose Acceleration) ตามเดิมเพื่อพิจารณารถคันต่อไป ในกรณีที่เป็นสัญญาณไฟแดง รถทุกคันใน Approach Portion จะถูกส่งไปพิจารณาพฤติกรรมและคำนวณหาความเร่งโดย Routine พฤติกรรมที่สัญญาณไฟแดง (Red Approach Behavior Routine) สำหรับรถ NFSTA ที่จะตรงไปจะมีการตรวจ Conflict ของรถที่ตรงมาจาก Crossing-list และรถที่เลี้ยวขวามาจาก Opposing list โดยส่งไปพิจารณาที่ Crossing-Conflict และ Sub-routine จุดตัดของรถเลี้ยวขวา (Right turner in conflict)

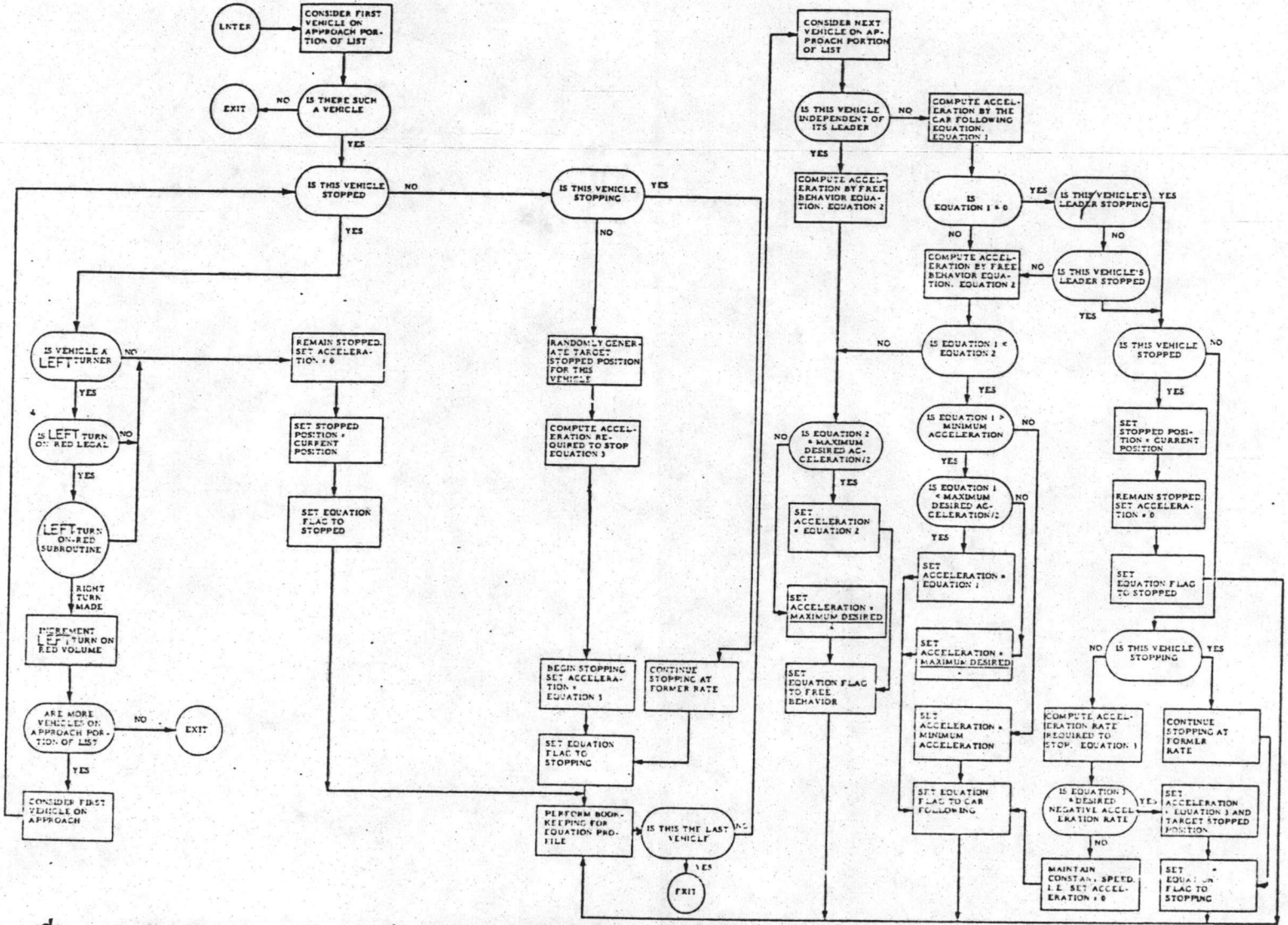
4.12 พฤติกรรมที่สัญญาณไฟแดง (Red-Approach Behavior)

รูป 4.13 เป็นผังภูมิแสดงการทำงานของ Routine นี้ ซึ่งจะพิจารณาพฤติกรรมและคำนวณความเร่งของรถทุกคันที่อยู่ในส่วน Approach Portion ของ Lists ที่มีสัญญาณไฟแดง หรือไฟเหลือง ซึ่งรถคันแรก (NFSTA) ได้ตัดสินใจหยุดจาก Routing การตัดสินใจที่สัญญาณไฟเหลือง (Amber Decision Routine) เมื่อพิจารณาพฤติกรรมของรถเสร็จสิ้นทุกคันแล้ว การพิจารณาพฤติกรรมของรถก็จะถูกส่งกลับมาที่โปรแกรมความเร็วและความเร่ง



รูปที่ 4.12 ผังภูมิการทำงานในส่วนความเร่งและความเร็ว (General-Purpose Acceleration)

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.13 ผังภูมิแสดงการทำงานในส่วนพฤติกรรมที่สัญญาณไฟแดง (Red-Approach Behavior)

(General Purpose Acceleration) เมื่อพิจารณาช่องจราจรต่อไป ในกรณีที่อนุญาติให้รถเลี้ยวซ้ายผ่านตลอด เมื่อรถที่ถูกพิจารณาเป็นรถเลี้ยวซ้ายการพิจารณาพฤติกรรมและคำนวณความเร่งจะถูกส่งไปที่ Routine เลี้ยวซ้ายผ่านตลอด (Left Turn on Red Routine)

4.13 <u>การตัดสินใจที่สัญญาณไฟเหลือง</u> (Amber Decision)

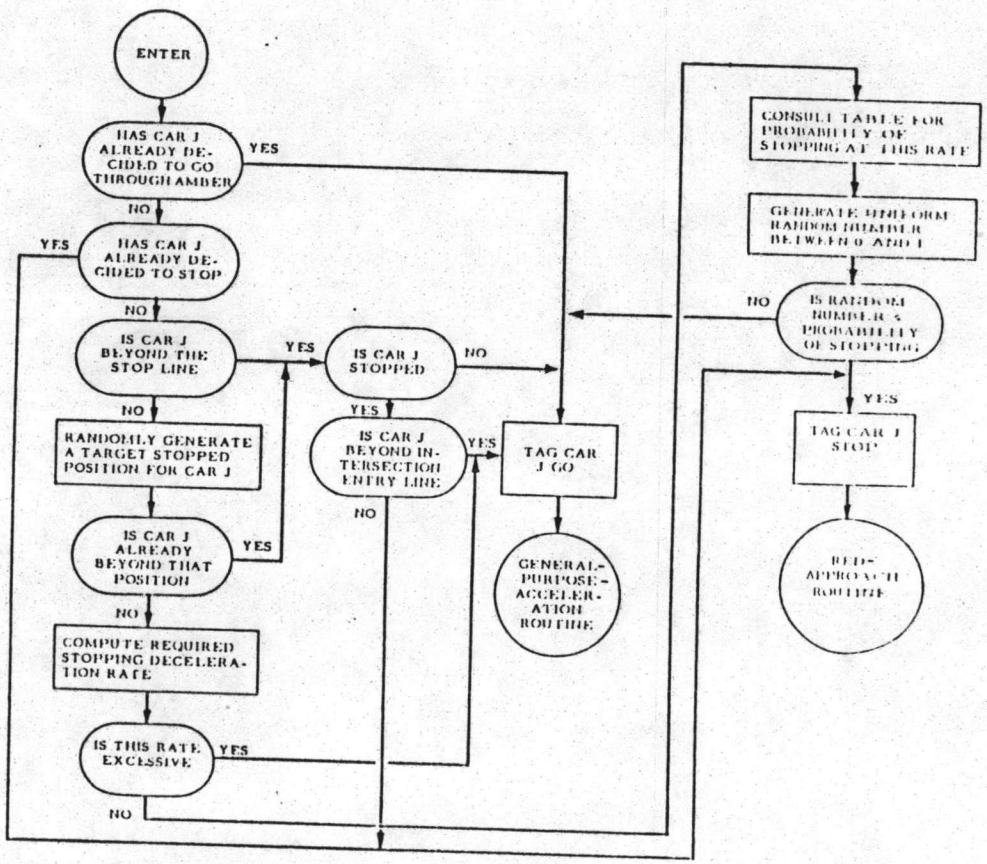
Subprogram ส่วนนี้เป็น Routine สำหรับการตัดสินใจของรถในส่วนของ Approach Portion ว่าจะหยุดหรือวิ่งต่อไปเมื่อเจอสัญญาณไฟเหลือง ถ้าผู้ขับขี่ตัดสินใจที่จะวิ่งต่อไป การพิจารณาต่าง ๆ จะถูกย้ายไปที่ Routine ความเร็วความเร่ง (General Purpose Acceleration) แต่ถ้าตัดสินใจที่จะหยุดการพิจารณาจะถูกส่งต่อไปที่พฤติกรรมที่สัญญาณไฟแดง (Red-Approach) หลักการในการตัดสินใจเมื่อเจอสัญญาณไฟเหลืองจะใช้วิธีคำนวณหาค่า Stopping Deceleration Rate Required จากตำแหน่งของรถที่เริ่มพิจารณาถึงเส้นหยุด (Stop line) พร้อมทั้ง Random เปรียบเทียบกับค่า Probability ของความเร่ง Acceleration ที่คำนวณได้ในตาราง 3.3 ดังแสดงเป็นผังภูมิในรูป 4.14

4.14 <u>พฤติกรรมการเลี้ยวซ้าย</u> (Left-Turn Behavior)

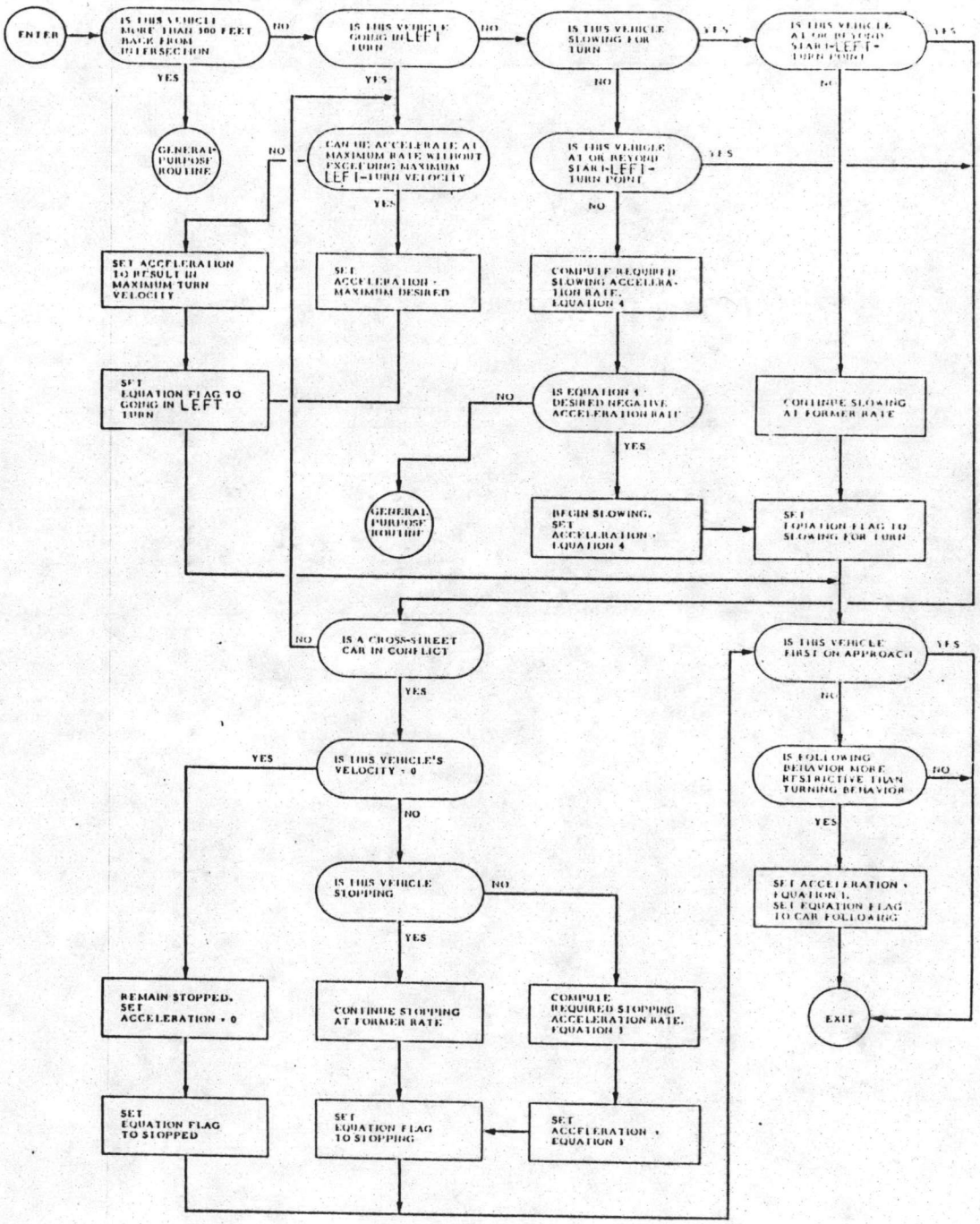
หลังจากพิจารณาแล้วว่าเป็นรถเลี้ยวซ้ายในส่วน Routine ความเร็ว,ความเร่ง (General Purpose Acceleration) (โดยตรวจดูจาก Flag word) การพิจารณาพฤติกรรมและการคำนวณความเร่งจะถูกส่งมาพิจารณาใน Routine ส่วนนี้ ซึ่งโปรแกรมส่วนนี้จะตรวจสอบ Conflict ของรถที่ตรงมาจาก Crossing List โดยส่งการพิจารณาตัดสินใจความเป็นไปได้ในการเลี้ยวซ้ายใน Subroutine จุดตัดของรถ (Crossing Conflict Subroutine) เมื่อรถตัดสินใจที่จะเลี้ยว การเลี้ยวจะดำเนินต่อไปตามพฤติกรรมที่กล่าวไว้ในบทที่ 3 และกำหนดค่า Flag word ตามพฤติกรรม ดูรูป 4.15 ประกอบ

4.15 <u>พฤติกรรมการเลี้ยวขวา</u> (Right-Turn Behavior)

Subprogram ส่วนนี้เป็น Routine ที่ทำหน้าที่คล้ายกับพฤติกรรมการเลี้ยวซ้าย (Left-Turn Behavior) โดยจะพิจารณาพฤติกรรมและคำนวณความเร่งของรถที่จะเลี้ยวขวา



รูปที่ 4.14 ผังภูมิแสดงการทำงานในส่วนการตัดสินใจที่สัญญาณไฟเหลือง (Amber Decision)



รูปที่ 4.15 ผังภูมิแสดงการทำงานในส่วนพฤติกรรมการเลี้ยวซ้าย

(Left-Turn Behavior)

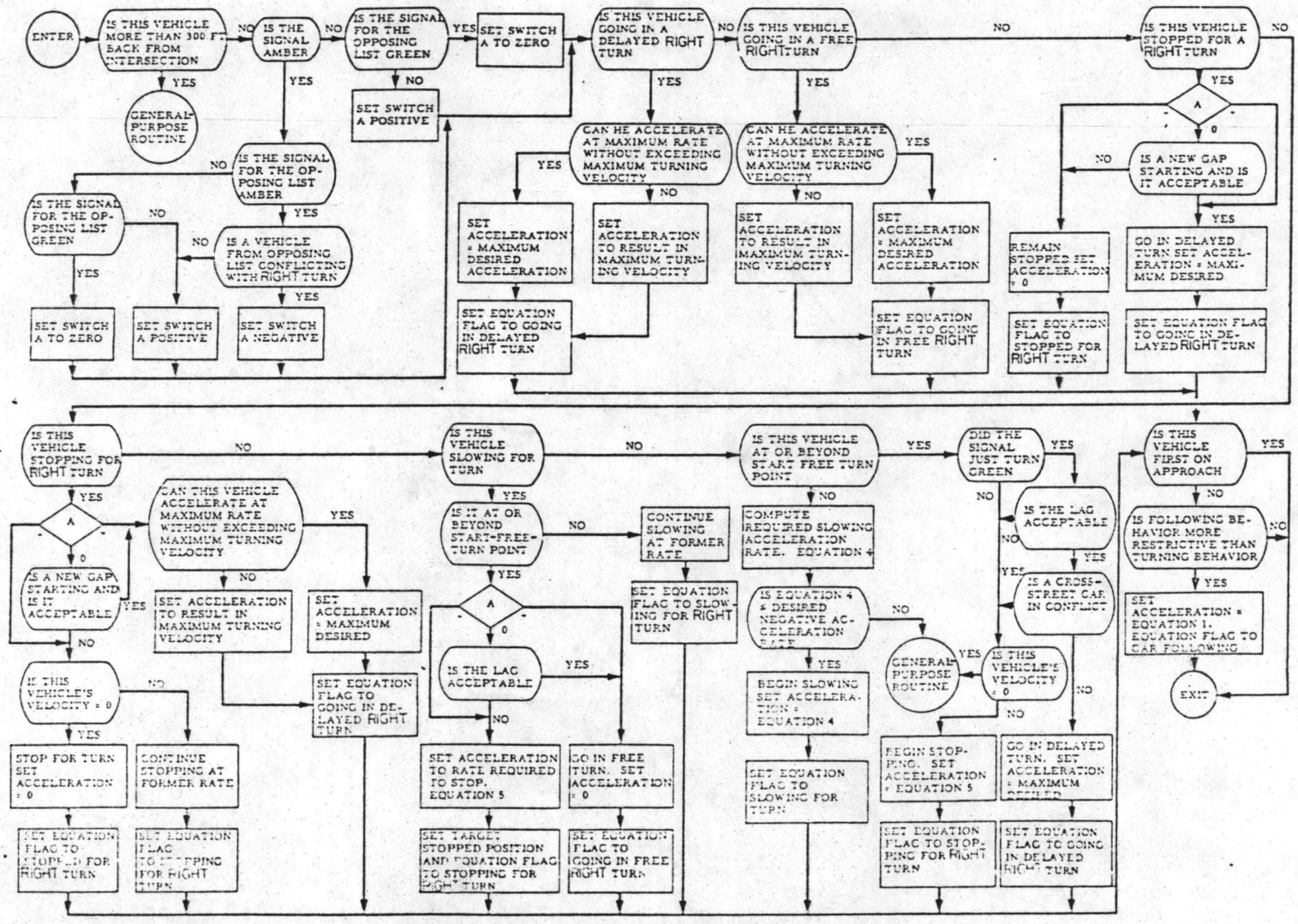
ตามพฤติกรรมที่ได้กล่าวมาแล้วในบทที่ 3 ซึ่งการเลี้ยวขวาของรถถูกกำหนดให้มีพฤติกรรมการเลี้ยวเป็น 2 ลักษณะ คือ Free-Turn และ Delay-Turn ในกรณี Opposing list มีสัญญาณไฟเขียวเช่นกัน รถที่เริ่มถูกพิจารณาการเลี้ยวจะถูกพิจารณาถึง Lag acceptable ใน Subroutine ช่วงเวลาที่ยอมรับของรถเลี้ยวขวาอิสระ (Right Turn Lag Acceptance Subroutine) และ Subroutine จุดตัดแบบ Crossing (Crossing Conflict Subroutine) ถึงความเป็นไปได้ในการเลี้ยวแบบ Free-Turn และถ้ามีปัญหาในการเลี้ยวแบบอิสระ (Free-Turn) การเลี้ยวแบบล่าช้า (Delay-Turn) จะถูกนำมาพิจารณาโดยพิจารณาถึง Gap acceptable ใน Subroutine ช่วงเวลาที่ยอมรับของรถเลี้ยวขวาแบบล่าช้า (Right-Turn Acceptance Subroutine) สำหรับกรณีที่ Opposing List มีสัญญาณไฟเหลือง จะมีการพิจารณา Conflict ของรถที่ตรงมาจาก Opposing List ใน Subroutine จุดตัดของรถตรงกันข้าม (Opposing Conflict Subroutine) ดังแสดงในรูป 4.16

4.16 พฤติกรรมการเลี้ยวซ้ายผ่านตลอด (Left-Turn on Red)

Subprogram ส่วนนี้เป็น Routine ที่มีหน้าที่ในการตัดสินใจถึงความเป็นไปได้ของการเลี้ยวซ้าย เมื่อมีสัญญาณไฟแดง โดยคำนวณเวลาเดินทางของรถที่ใกล้จุด Conflict ที่สุดถึงจุด Conflict ใน Receiver List พร้อมทั้ง Random เปรียบเทียบกับค่า Lag acceptable ในตารางที่ 3.5

4.17 ผลลัพธ์ (Out Put)

จะเป็นโปรแกรมส่วนที่ทำงานเกี่ยวกับการคำนวณและแสดงผลทางสถิติ โดยได้ข้อมูลจากการเก็บสะสมข้อมูลมาจาก Routine ต่าง ๆ นำมาคำนวณค่าทางสถิติ และแสดงผลในรูปที่เข้าใจง่าย ดูภาคผนวก ก. ประกอบ โดยจะเรียกใช้ Subroutine ต่าง ๆ ได้แก่ Subroutine OUTP1 ถึง Subroutine OUTP7, Subroutine MEAN, Subroutine STDV, Subroutine MAXIM และ Subroutine MINIM ซึ่งจะกล่าวรายละเอียดในหัวข้อต่อไป



รูปที่ 4.16 ผังภูมิแสดงการทำงานในส่วนพฤติกรรมการเลี้ยวขวา (Right-Turn Behavior)

4.18 โปรแกรมย่อย (Subroutine)

Subroutine เป็นโปรแกรมย่อยที่ถูกเรียกใช้จากหลายส่วนของโปรแกรมหลัก ซึ่งจะทำให้ประหยัดเวลาและเนื้อที่ในหน่วยความจำ ส่วนที่จะกล่าวต่อไปนี้เป็นการอธิบายถึงการทำงานของ Subroutine ต่าง ๆ ดังนี้

4.18.1 โปรแกรมย่อยสำหรับการย้ายข้อมูลของรถ (Subroutine Modify Lane "Move" and Change Car's Characteristics "Change")

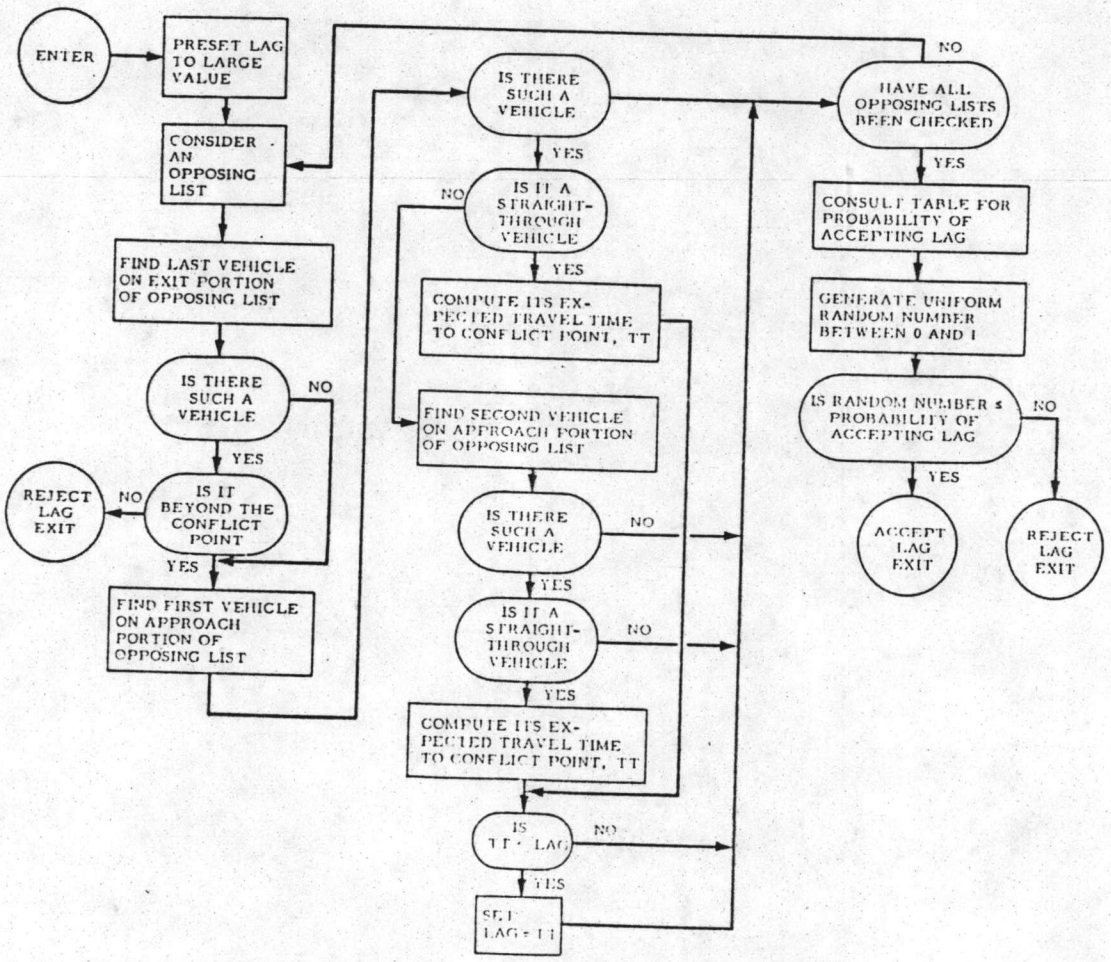
เป็น Subroutine ที่ทำหน้าที่ในการวางตำแหน่งของรถที่จะเลี้ยวเข้าไปใน Receiver List โดยเทียบตำแหน่ง (Position) ของรถที่เลี้ยวและตำแหน่งของรถใน Receiver List เมื่อหาตำแหน่งได้แล้วจะทำการปรับลำดับของรถ (J) ทั้งใน Origin List และ Receiver List พร้อมทั้งย้ายข้อมูลของรถโดยใช้ Subroutine Move Car's Characteristic (Change)

4.18.2 โปรแกรมย่อยสำหรับหาช่วงเวลาที่ยอมรับของรถเลี้ยวขวาอิสระ (Subroutine Right Turn Lag Acceptance "RLAG")

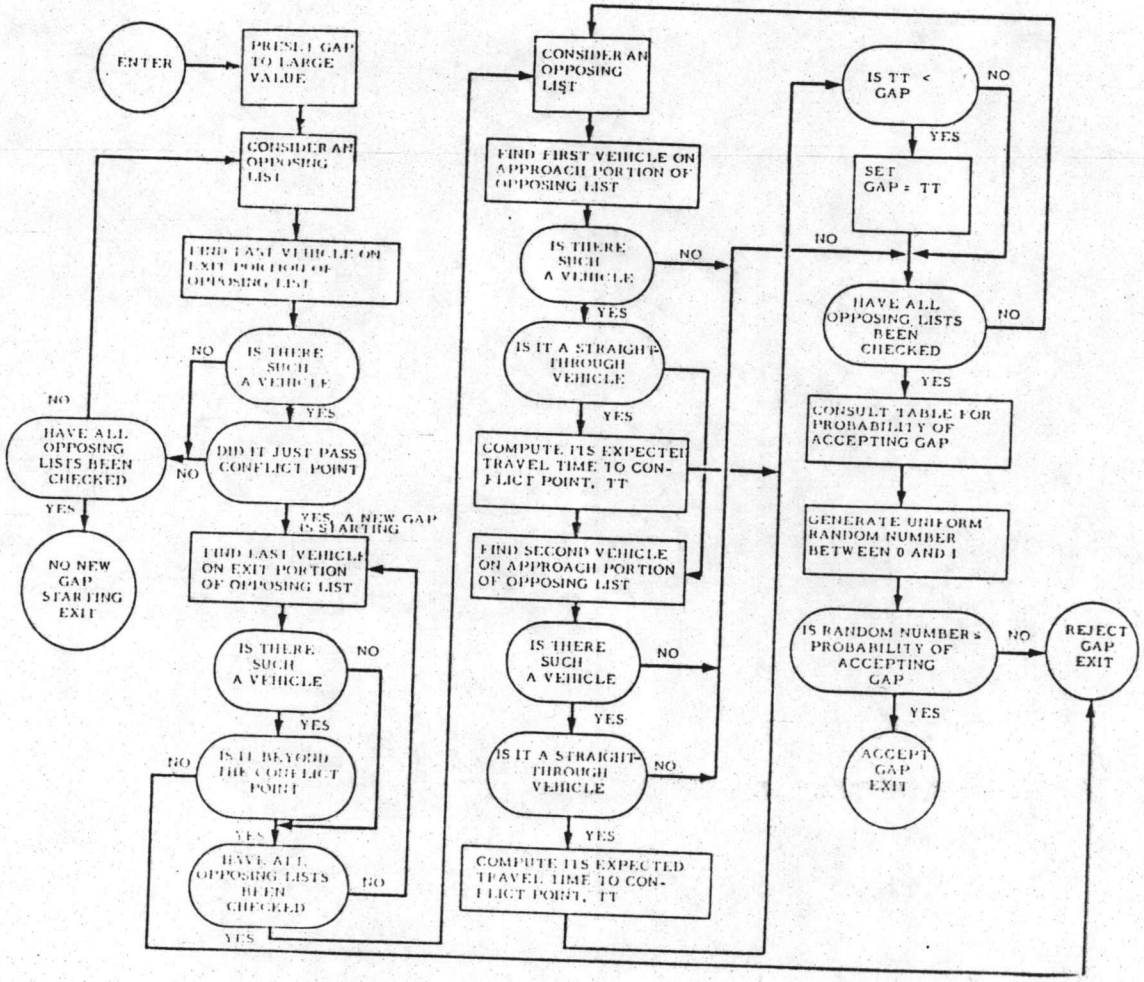
เป็น Subroutine ที่ใช้เป็นตัวตัดสินใจของรถที่จะเลี้ยวขวา โดยคำนวณเวลาของรถที่ตรงมาจาก Opposing List ถึงจุด Conflict ของการเลี้ยวขวา และทำการ Random เปรียบเทียบกับค่า Probability ในตาราง 3.5 ถ้าค่า Random มากกว่าค่าของ Probability ในตาราง 3.5 ถือว่าผ่านได้และ set ค่า R = 1 เมื่อค่า Random น้อยกว่าจะ set ค่า R = -1 พร้อมทั้งส่งค่า R กลับสู่โปรแกรมหลักที่เรียกใช้เพื่อให้รู้ถึงการตัดสินใจ ดูรูป 4.17

4.18.3 โปรแกรมย่อยสำหรับหาช่วงเวลาที่ยอมรับของรถเลี้ยวขวาแบบล่าช้า (Subroutine Right Turn Car Acceptance "RGAP")

เป็น Subroutine ที่ทำหน้าที่เหมือน Subroutine Right Turn Lag แต่ค่าที่ใช้ในการตัดสินใจจะใช้ค่าจาก GAP ในตาราง 3.5 ดูผังงานในรูป 4.18



รูปที่ 4.17 ผังภูมิแสดงการทำงานของโปรแกรมย่อยสำหรับหาช่วงเวลาที่ยอมรับของรถเลี้ยวขวาแบบอิสระ (Right-Turn Lag Acceptance)



รูปที่ 4.18 ผังภูมิแสดงการทำงานของโปรแกรมย่อยสำหรับหาช่วงเวลาที่ยอมรับของรถเลี้ยวขวาแบบล่าช้า (Right-Turn Gap Acceptance)

4.18.4 <u>โปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถจากฝั่งตรงข้าม</u> (Subroutine Opposing Conflict "OPPOS")

เป็น Subroutine ที่ทำหน้าที่ในการตรวจสอบว่ามีรถที่ตรงมาจาก Opposing List จะอยู่ในบริเวณจุด Conflict ของแนวการเลี้ยวขวาหรือไม่ การตรวจสอบจะทำทุก Opposing List ถ้ามีรถอยู่ในบริเวณจุด Conflict จะ set ค่า R = -1 และถ้าไม่มีรถอยู่ในบริเวณ Conflict ในทุก Opposing List ก็จะ set ค่า R = 1 และส่งค่า R กลับไปที่โปรแกรมหลักส่วนที่เรียกใช้ ซึ่งทำให้รู้ว่ามีรถอยู่ในบริเวณ Conflict หรือไม่ ดูรูป 4.19

4.18.5 <u>โปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถจากด้านขวาง</u> (Subroutine Crossing Conflict "CROSS")

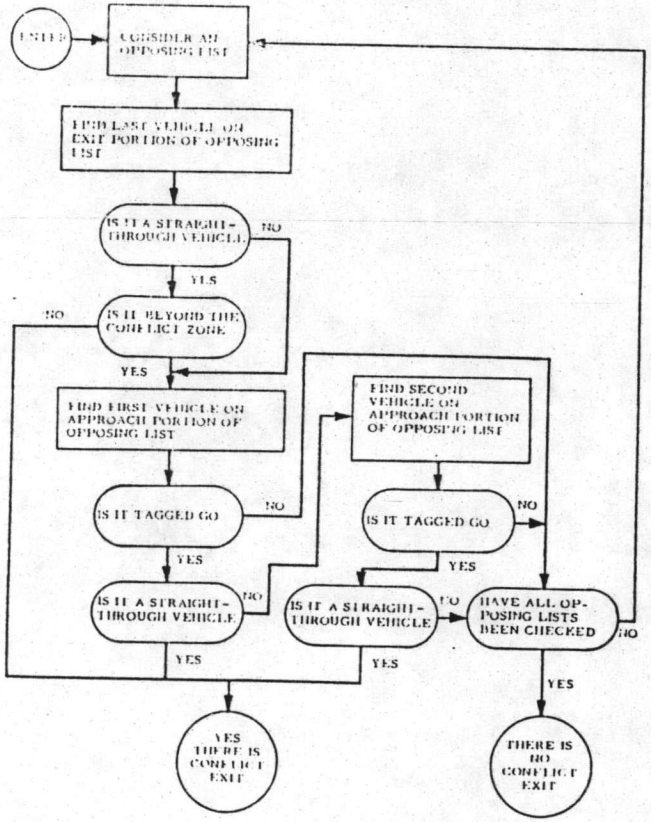
เป็น Subroutine ที่ทำหน้าที่คล้าย Subroutine Opposing Conflict โดยจะทำการตรวจสอบว่ามีรถที่ตรงมาจาก Crossing List อยู่ในบริเวณจุด Conflict หรือไม่ ดูผังภูมิในรูป 4.20

4.18.6 <u>โปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถที่เลี้ยวขวา</u> (Subroutine Right-Turn in Conflict "RTURN")

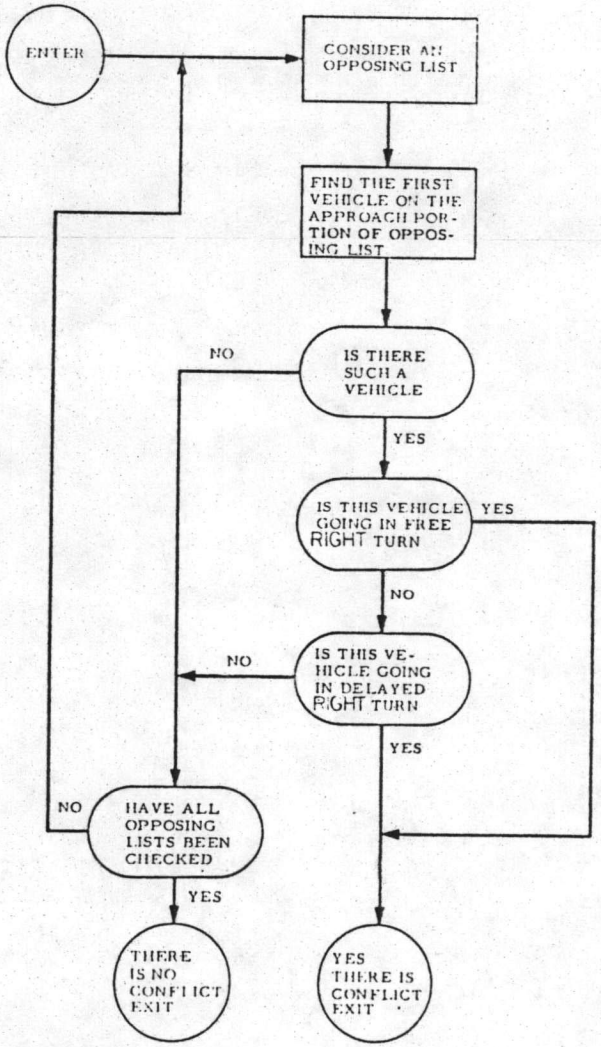
เป็น Subroutine ที่ทำหน้าที่คล้ายกับ Subroutine Opposing Conflict โดยจะพิจารณาในกรณีที่รถเลี้ยวขวาใน Opposing List กำลังมีสัญญาณไฟเหลือง ซึ่งผ่านการตัดสินใจที่จะเลี้ยวมาแล้วใน Amber Decision ดูผังภูมิในรูป 4.21

4.18.7 <u>โปรแกรมย่อยสำหรับการสุ่มหาคุณลักษณะ</u> (Subroutine Generate Characteristics "GEN")

เป็น Subroutine ที่ทำหน้าที่ในการ Integrate หา Normal Distribution Probability ซึ่งเป็น f(Z) ($Z = \frac{x-u}{\sigma}$) เมื่อ Generate คุณลักษณะของรถเมื่อเริ่มเข้าสู่ระบบ ในการทำงานส่วนนี้จะ Integrate ฟังค์ชั่นของ Z {f(Z)} โดยจะให้

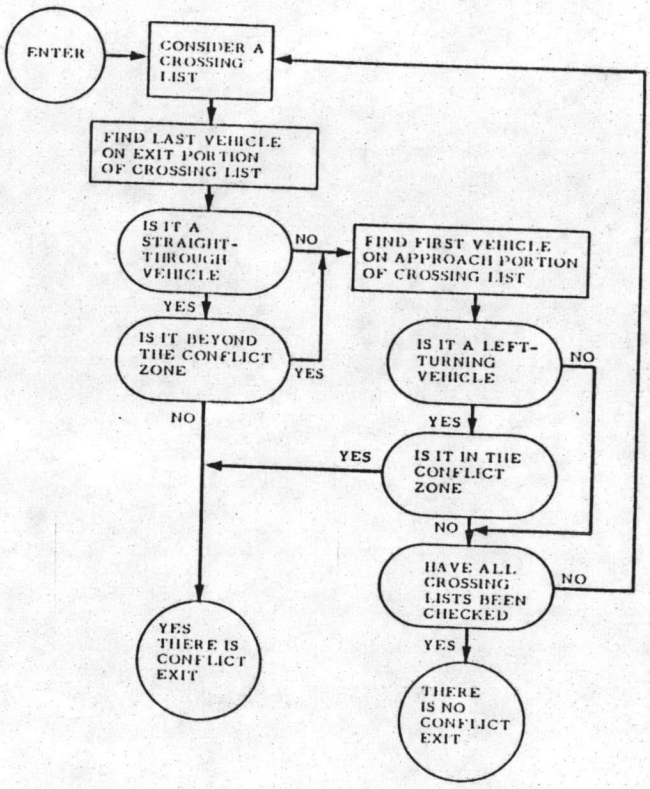


รูปที่ 4.19 ผังภูมิการทำงานของโปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถจากฝั่งตรงข้าม (Opposing, Conflict)



รูปที่ 4.20 ผังภูมิการทำงานของโปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถเลี้ยวขวา (Right-Turn in Conflict)

หอสมุดกลาง สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 4.21 ผังภูมิแสดงการทำงานในส่วนโปรแกรมย่อยสำหรับพิจารณาจุดตัดของรถจากด้านขวาง (Crossing Conflict)

มีค่า n = 10 (ดูภาคผนวก ค. ประกอบ) ซึ่งจะให้ค่า Probability ได้ใกล้เคียงพอสมควร แล้วนำมาเปรียบเทียบกับค่า Random ที่ได้มาจาก Vehicle Arrival and Characteristics Generation เมื่อค่า Probability มากกว่าค่า Random คุณลักษณะของรถจะเป็นค่า X

4.18.8 <u>ผลลัพธ์</u> (Subroutine Output)

หัวข้อนี้จะเป็นการกล่าวถึง Subroutine ทั้งหมดที่ใช้ในการประมวลผลทางสถิติในส่วน Output ดูภาคผนวก ก. ประกอบ

1. OUTP1 เป็น Subroutine ที่แยกข้อมูลของแต่ละ list และรวมข้อมูลของทุก list ที่อยู่ในทิศ (Direction) เดียวกัน รวมทั้งรวมข้อมูลทั้งหมดในช่อง Grand Total โดยคุณลักษณะข้อมูลที่จะใช้ใน Subroutine นี้จะเป็น Real Value

2. OUTP2 เป็น Subroutine ที่ทำหน้าที่แสดงผลทางสถิติของ Travel Time และยังเปลี่ยนหน่วยของเวลาจากวินาทีเป็นนาที

3. OUTP3 เป็น Subroutine ที่ใช้ประมวลผลทางสถิติ โดยแบ่งข้อมูลเป็นช่อง (inteval) แสดง Frequency, percent และ Cumulative percent ข้อมูลที่ใช้ใน Subroutine นี้จะมีลักษณะเป็น Integer value

4. OUTP4 เป็น Subroutine ที่ทำหน้าที่เหมือน OUTP1 แต่ละลักษณะข้อมูลที่ใช้กับ Subroutine นี้จะเป็น Integer value

5. OUTP5 เป็น Subroutine ที่ทำหน้าที่เหมือน OUTP3 แต่ลักษณะข้อมูลที่ใช้กับ Subroutine นี้จะเป็น Real value

6. OUTP6 เป็น Subroutine ที่ทำหน้าที่คล้าย OUTP2 แต่จะเป็นการแสดงผลเกี่ยวกับเรื่อง Travel Speed และแปลงหน่วยความเร็วจาก เมตร/วินาที เป็น กิโลเมตร/ช.ม.

7. OUTP7 เป็น Subroutine ในการหาค่า Percent ของบริเวณรถในแต่ละแบบ ซึ่งถูกเรียกใช้จาก Output Routine

8. MEANV เป็น Subroutine ที่ใช้ในการคำนวณหาค่าเฉลี่ย (Mean) ของข้อมูลในแต่ละ list และในแต่ละทิศรวมทั้งค่าเฉลี่ยรวมทั้งระบบ

9. STDV เป็น Subroutine ที่ใช้ในการคำนวณค่าความเบี่ยงเบน (Standard Deviation) ของข้อมูลในแต่ละ list และในแต่ละทิศทางรวมทั้งค่าความเบี่ยงเบนรวมของทั้งระบบ

10. MAXIM เป็น Subroutine ที่ใช้หาค่ามากสุด (Maximum) ของแต่ละ list และในแต่ละทิศทางรวมทั้งค่ามากสุดของระบบ

11. MANIM เป็น Subroutine ที่ช้หาค่าต่ำสุด (Minimum) ของแต่ละ list และในแต่ละทิศทางรวมทั้งค่าต่ำสุดของระบบ

4.19 <u>ฟังค์ชั่นสำหรับการสุ่มค่า</u> (Real Function Randu)

เป็น function ที่ถูกเรียกใช้จากโปรแกรมหลักเพื่อสร้างค่า Random function นี้ผู้วิจัยได้สร้างขึ้นเนื่องจากคอมพิวเตอร์บางรุ่นไม่มี function ของ Random การสร้างค่า Random ในแบบจำลองนี้จะทำการสร้างค่าขึ้น 100 ค่า และเก็บไว้ในหน่วยความจำ เมื่อจะใช้ค่า Random ก็จะเรียกมาใช้ทีละค่า