

บทที่ 3

โปรแกรมวิเคราะห์กระแสเตเตอร์

ในบทที่ 2 ได้กล่าวถึงวิธีการตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส โดยแบ่งการตรวจจับความผิดปกติเป็น 2 ส่วนคือ ส่วนตรวจวัดกระแสเตเตอร์ ซึ่งได้กล่าวรายละเอียดในบทที่ 2 แล้ว ส่วนโปรแกรมวิเคราะห์กระแสเตเตอร์จะขอกกล่าวถึงรายละเอียดในบทนี้

จากการรวบรวมวิธีการสำหรับใช้ในการตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส สามารถแบ่งได้เป็น 3 ชนิดคือ ความผิดปกติที่เกิดขึ้นเนื่องจากการแตกหักของแท่งตัวในโรเตอร์ ความผิดปกติที่เกิดขึ้นเนื่องจากความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และความผิดปกติที่เกิดขึ้นเนื่องจากการลัดวงจรระหว่างรอบของขดลวดสเตเตอร์ ซึ่งได้กล่าวรายละเอียดแล้วในบทที่ 2 และในการเขียนโปรแกรมวิเคราะห์กระแสเตเตอร์ได้ทำการเขียนโปรแกรมโดยใช้ความรู้ที่ได้ทำการศึกษารวบรวมจากบทที่ 2 และใช้ LabVIEW (Laboratory Virtual Instrument Engineering Workbench) เป็นเครื่องมือในการเขียนโปรแกรม

3.1 LabVIEW (Laboratory Virtual Instrument Engineering Workbench)

LabVIEW เป็นโปรแกรมที่ใช้ในการพัฒนาโปรแกรมสำหรับนำไปใช้งานต่าง ๆ ได้อย่างกว้างขวาง โดยใช้ภาษาในการเขียนโปรแกรมที่เป็นรูปภาพ (Graphical programming language) หรือเรียกว่า ภาษาจี (G-Language) ซึ่งแตกต่างกับภาษาซี ปาสคาล เบสิก หรือภาษาอื่น ๆ ที่รูปแบบคำสั่งต่าง ๆ ของภาษาเหล่านี้จะเป็นชุดคำสั่งที่เป็นข้อความ (Text-based language) LabVIEW ได้รวมการติดต่อกับฮาร์ดแวร์ด้วยวิธีต่าง ๆ ไว้ด้วยกัน เช่น การติดต่อผ่านทาง GPIB (General Purpose Interface Bus), VXI (VME eXtensions for Instrumentation bus), RS-232 (Recommended Standard 232), RS-485 (Recommended Standard 485 และ DAQ board (Plug-in data acquisition board) และยังสามารถรวบรวมและสร้างฟังก์ชันต่าง ๆ ที่เกี่ยวกับการใช้งานโปรแกรมมาตรฐานต่าง ๆ เช่น การติดต่อระบบเครือข่ายโดยใช้ TCP/IP และระบบติดต่ออัตโนมัติ (OLE-Automation)

LabVIEW สามารถนำไปใช้ในการสร้างโปรแกรมต่าง ๆ ได้หลายรูปแบบ เริ่มจากกระบวนการง่าย ๆ ตลอดจนกระบวนการที่สลับซับซ้อน ดังแสดงในรูปที่ 3.1 สามารถนำไปใช้เป็นตัววัดอุณหภูมิไปจนถึงสร้างเป็นดิจิทัลออสซิลโลสโคปได้ LabVIEW ได้ทำการรวบรวม Library ต่าง ๆ ซึ่งสามารถนำไปประยุกต์ใช้งานได้อย่างกว้างขวางเช่น ใช้ในการจัดเก็บข้อมูล การวิเคราะห์ข้อมูล การนำเสนอข้อมูล และการเก็บข้อมูล และยังได้รวบรวมเครื่องมือที่ใช้ในการพัฒนาโปรแกรมต่าง ๆ สามารถตรวจสอบการทำงานของโปรแกรมในแต่ละขั้นตอนได้โดยง่าย สะดวกและง่ายต่อการแก้ไขข้อผิดพลาดของโปรแกรม ทำให้การพัฒนาโปรแกรมง่ายขึ้น อีกทั้งยังนำไปใช้ประโยชน์ได้ดีในการแก้ปัญหาทางวิทยาศาสตร์และปัญหาทางวิศวกรรม เนื่องจากภาษาที่ใช้ในการเขียนโปรแกรมมีความยืดหยุ่นและมีสมรรถนะสูง ไม่ยุ่งยากสลับซับซ้อน

การเรียกชื่อคำศัพท์ต่าง ๆ บน LabVIEW โดยเทียบเคียงกับคำศัพท์ที่ใช้ในการเขียนโปรแกรมทั่วไป สามารถสรุปได้ดังแสดงในตารางที่ 3.1 ดังนี้

ตารางที่ 3.1 ตารางเทียบเคียงคำศัพท์ที่ใช้เรียกใน LabVIEW และโปรแกรมทั่วไป

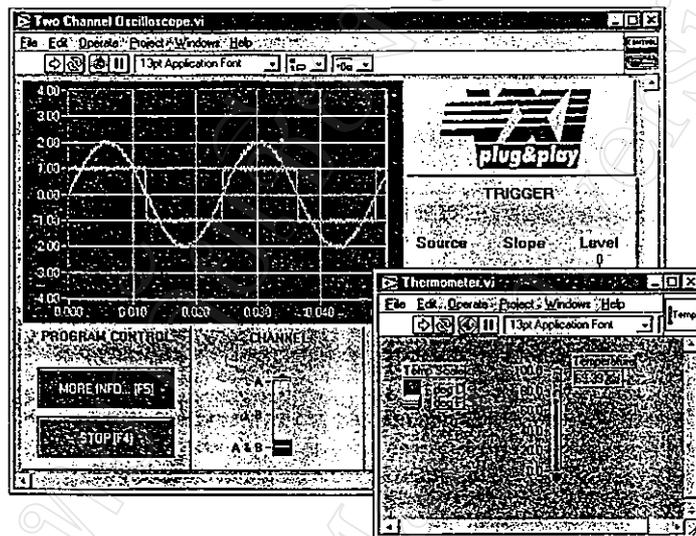
LabVIEW	Conventional Program
VI	Program
Function	Function
SubVI	Subroutine, subprogram
Front panel	User interface
Block diagram	Program code
G	C, Pascal, Basic, etc.

3.1.1 การทำงานของ LabVIEW

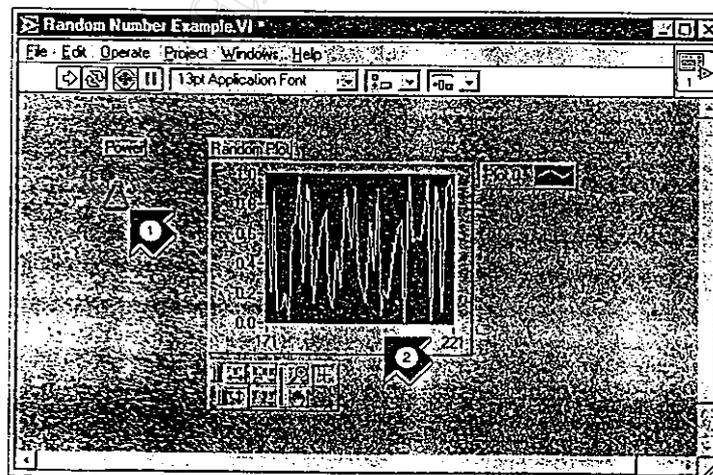
โปรแกรมของ LabVIEW หรือที่เรียกว่า เครื่องมือวัดเสมือน (Virtual Instruments, VIs) ประกอบด้วย Front panel และ แผนผัง (Block diagram) เครื่องมือต่าง ๆ ที่ใช้ในการสร้างโปรแกรม เรียกว่า Palettes และวิธีการส่งผ่านข้อมูลของโปรแกรม (Data flow)

(1) Front panel เป็นส่วนโปรแกรมที่ติดต่อกับผู้ใช้งาน (Graphic user interface) ส่วนนี้จะเป็นส่วนรับข้อมูลเข้าและแสดงส่วนผลลัพธ์ของโปรแกรม ซึ่งสามารถแสดงผลได้หลายรูปแบบ เช่น สวิตช์ลูกบิด สวิตช์ปุ่มกด รูปกราฟและอื่น ๆ ดังแสดงในรูปที่ 3.2 ลูกศรหมายเลข 1 คือ Control input ลูกศรหมายเลข 2 คือ Indicator output

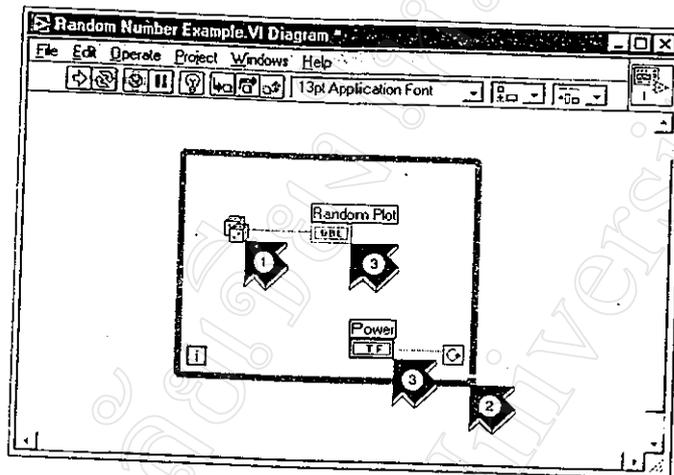
(2) แผนผัง (Block diagram) ประกอบด้วยคำสั่งต่าง ๆ ของโปรแกรมที่เป็นรูปภาพ ฟังก์ชัน คำสั่งโครงสร้างต่าง ๆ และ Terminal ที่สัมพันธ์กับ Control และ Indicator บน Front panel ดังแสดงในรูปที่ 3.3 ลูกศรหมายเลข 1 คือ ฟังก์ชัน ลูกศรหมายเลข 2 คือ คำสั่งโครงสร้าง และ ลูกศรหมายเลข 3 แสดง Terminal จาก Front panel



รูปที่ 3.1 การนำ LabVIEW ไปใช้งานในรูปแบบต่าง ๆ



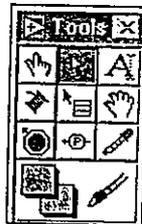
รูปที่ 3.2 Front panel ของโปรแกรม



รูปที่ 3.3 แผนผังของโปรแกรม

(3) Palettes ประกอบด้วย Tools palette, Controls palette และ Functions palette

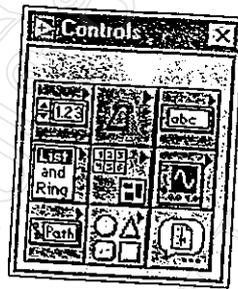
Tools palette ประกอบด้วยเครื่องมือที่ใช้สำหรับแก้ไข objects บน front panel และแผนผัง ดังแสดงในรูปที่ 3.4



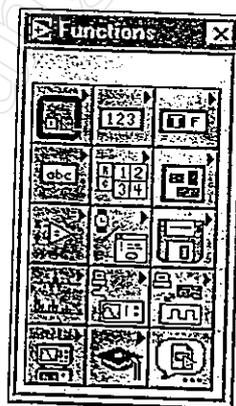
รูปที่ 3.4 Tools palette ของโปรแกรม

Controls palette ประกอบด้วย Controls และ Indicators ของ Front panel ที่จะต้องใช้ในการสร้างส่วนติดต่อกับผู้ใช้งาน (User interface) ดังแสดงในรูปที่ 3.5

Functions palette ประกอบด้วย Objects ต่าง ๆ ที่ใช้ในการสร้างโปรแกรม (VI) เช่น ฟังก์ชันทางคณิตศาสตร์ Instrument I/O, File I/O, ฟังก์ชันที่ใช้ในการวิเคราะห์ เป็นต้น ดังแสดงในรูปที่ 3.6



รูปที่ 3.5 Controls palette ของโปรแกรม



รูปที่ 3.6 Functions palette ของโปรแกรม

(4) **Data flow** เป็นลักษณะการทำงานส่งผ่านข้อมูลของ LabVIEW ในขณะที่โปรแกรมทำงาน Block diagram ในรูปที่ 3.7 ประกอบด้วยโหนดต่างๆ เช่น VIs คำสั่งโครงสร้างต่างๆ terminals จาก Front panel โหนดต่างๆ เหล่านี้ถูกต่อโดยการลากเส้นต่อกัน ซึ่งจะเป็นการกำหนดการส่งผ่านข้อมูลไปยังโปรแกรม และแต่ละโหนดจะทำงานก็ต่อเมื่อมีข้อมูลอินพุตครบเท่านั้น เมื่อโหนดนั้นจบการทำงานก็จะส่งข้อมูลออกไปยังโหนดต่อไปตามทางเดินของข้อมูลที่ได้ทำการเขียนโปรแกรมไว้

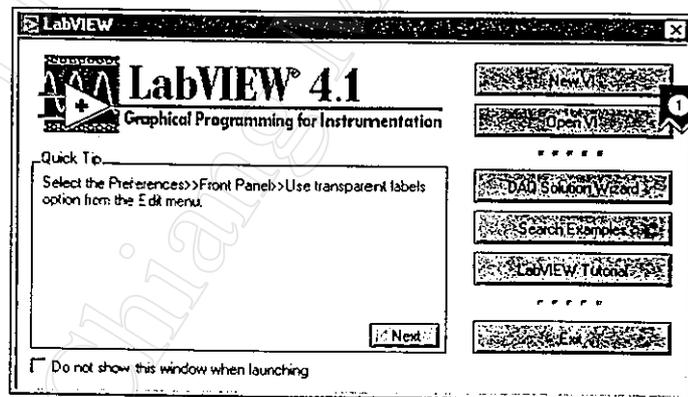
3.1.2 การสร้างโปรแกรมบน LabVIEW

การสร้างโปรแกรมบน LabVIEW หรือ Virtual Instruments (VIs) มีขั้นตอนดังนี้

3.1.2.1 การสร้างส่วนติดต่อกับผู้ใช้งาน (Front panel)

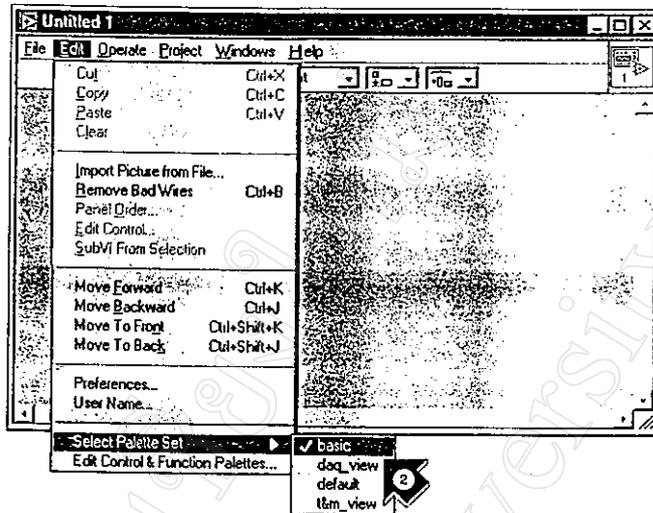
(1) สร้าง VI ใหม่โดยเลือก New VI ใน LabVIEW dialog box ดังแสดง

ในรูปที่ 3.7



รูปที่ 3.7 การสร้างโปรแกรมใหม่

(2) บน Front panel ที่สร้างขึ้นใหม่ ให้เลือก Edit>>Select>>Set>>basic ดังแสดงในรูปที่ 3.8



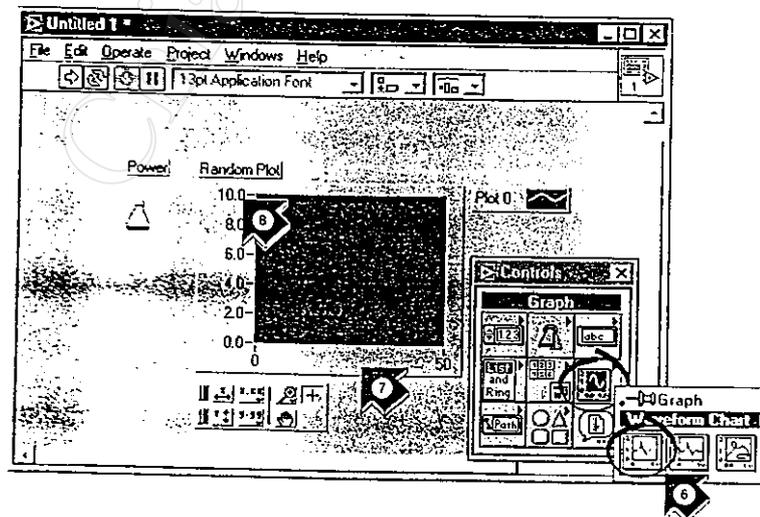
รูปที่ 3.8 การเลือก Palette

(3) เลือก Vertical Toggle Switch จาก sub palette **Control**>>**Boolean** แล้วเลือกไอคอนของ Toggle switch

(4) เลื่อนตัวชี้มายัง Front panel แล้วคลิกวาง Toggle switch บน front panel

(5) พิมพ์ Power ลงบนชื่อของ Toggle switch

(6) สร้าง Waveform chart โดยเลือก **Controls**>>**Graph**>>**Waveform chart** ดังแสดงในรูปที่ 3.9

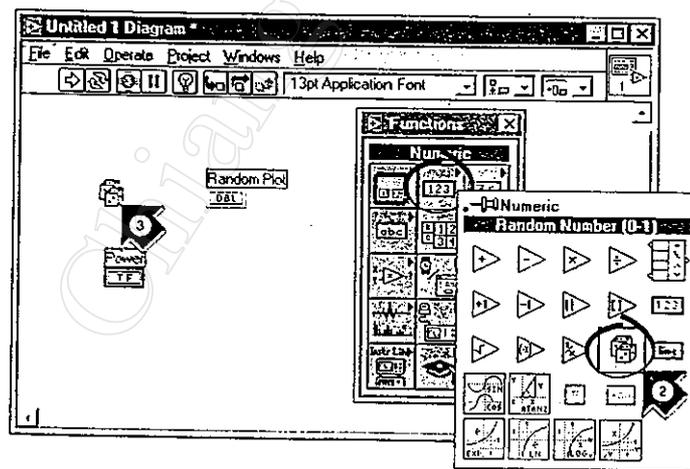


รูปที่ 3.9 การสร้าง Waveform chart

- (7) คลิกวาง Chart ลงบน Front panel แล้วใส่ชื่อ Random plot
- (8) เปลี่ยนสเกลของ Waveform chart โดยเลือก **Windows>>Show tools Palette** ดับเบิลคลิกที่ 10.0 บนแกน Y แล้วป้อนค่า 1.0

3.1.2.2 การสร้างแผนผังของโปรแกรม

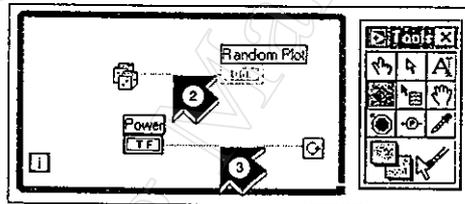
- (1) เลือก Show diagram จาก Windows จะพบ terminals ของทั้ง 2 อย่างที่ได้สร้างขึ้นบน Front panel มาแสดงที่แผนผัง ดังแสดงในรูปที่ 3.10
- (2) เลือกฟังก์ชันในการสุ่มตัวเลข Random number (0-1) จาก **Function>>Numeric** ถ้า Function palette ไม่มี ให้เลือก **Windows>>Show Function palette**
- (3) คลิกวางฟังก์ชันการสุ่มตัวเลขลงบนแผนผัง
- (4) เลือก While loop จาก **Functions >> Structures**
- (5) คลิกวางตัวชี้ลงบนแผนผัง แล้วทำการจัดตำแหน่งให้เหมาะสม



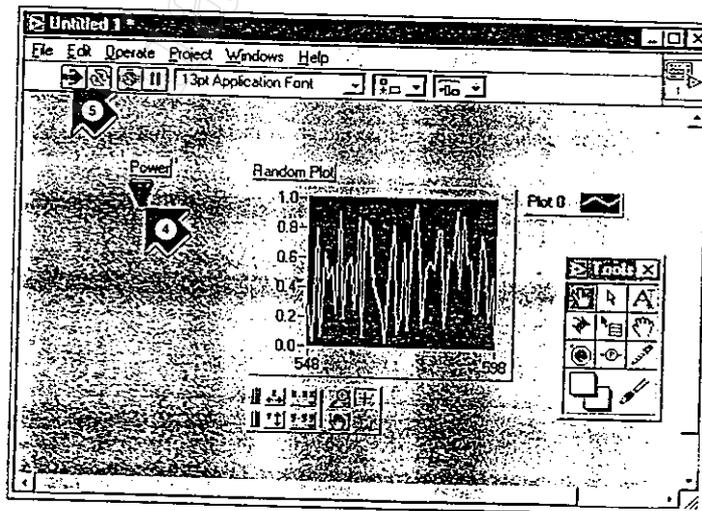
รูปที่ 3.10 การสร้างแผนผังของโปรแกรม

3.1.2.3 การต่อสายเชื่อมระหว่าง Objects และการ Run โปรแกรม

- (1) เลือก Wiring tool จาก tools palette
- (2) ต่อสายเชื่อมเลือกฟังก์ชันในการสุ่มตัวเลขไปยัง Random plot chart terminal โดยในการสร้างสายต่อเชื่อมให้คลิกที่ฟังก์ชันในการสุ่มตัวเลขแล้วลากไปยัง Random Plot Chart แล้วคลิกอีกครั้งหนึ่งเพื่อเป็นการปลดสายออกจาก Random Plot Chart ดังแสดงในรูปที่ 3.11
- (3) ต่อสาย Toggle switch terminal ไปยังเงื่อนไขของ While Loop
- (4) กลับไปที่ Front panel และเลือก Operating tool คลิก Power toggle switch ให้อยู่ตำแหน่ง True ดังแสดงในรูปที่ 3.12
- (5) คลิกปุ่ม Run เพื่อทำการ Run VI
- (6) หยุดการทำงานของ โปรแกรมโดยคลิกที่ Power toggle switch อีกครั้ง เพื่อให้กลับสู่ตำแหน่ง False



รูปที่ 3.11 การต่อสายเชื่อมระหว่าง Objects



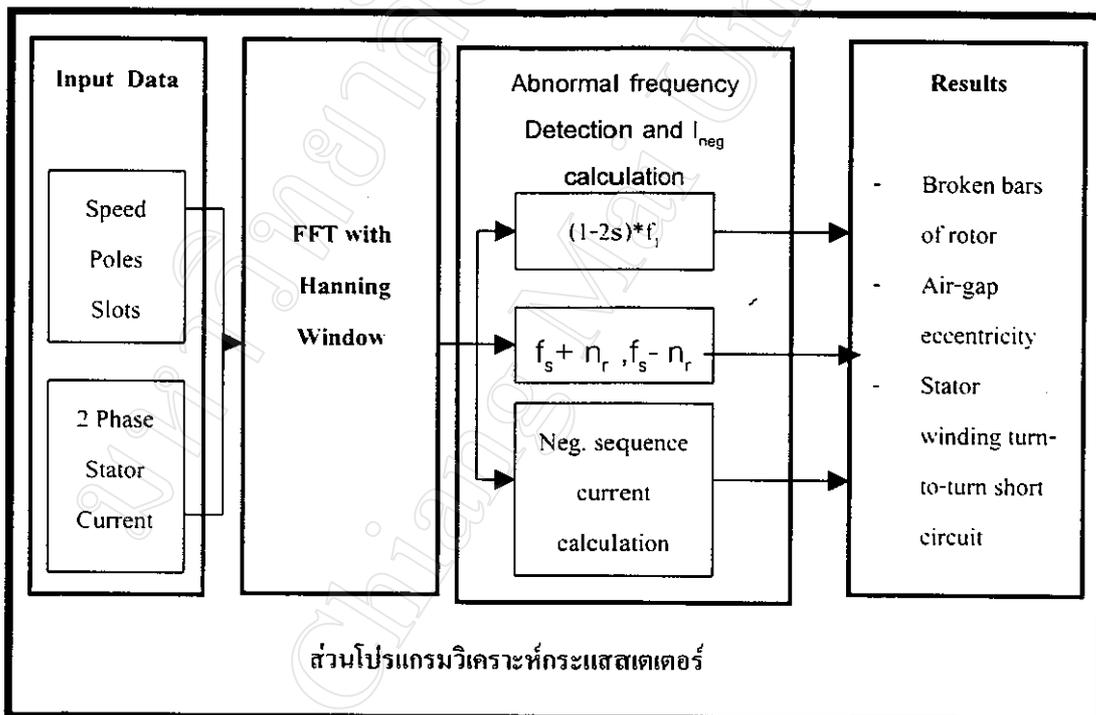
รูปที่ 3.12 การ Run โปรแกรม

3.2 อัลกอริทึมในการเขียนโปรแกรมตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส

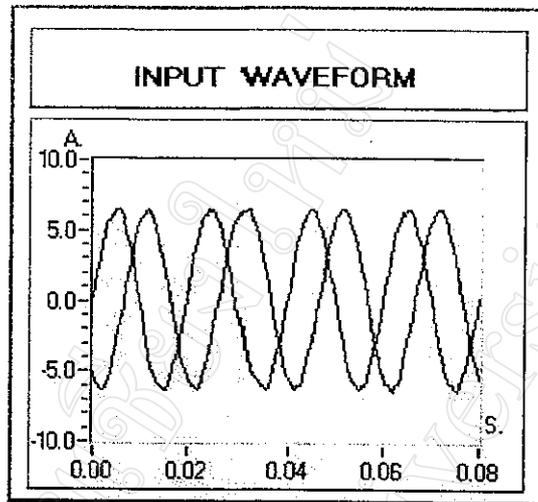
โปรแกรมตรวจจับความผิดปกติเขียนขึ้นโดยใช้ Labview เป็นเครื่องมือในการเขียนโปรแกรม ข้อดีของ Labview คือเป็นเครื่องมือที่เขียนโปรแกรมได้ง่าย โดยจะมี Library ของฟังก์ชันที่สำคัญต่าง ๆ มาก สำหรับอัลกอริทึมในการเขียนโปรแกรมได้แสดงดังรูปที่ 3.13

โปรแกรมประกอบด้วย 4 ส่วนหลัก คือ

3.2.1 ส่วนรับข้อมูล เป็นข้อมูลกระแสสแตเตอร์ 2 เฟส ข้อมูลที่ใช้ในการคำนวณและข้อมูลจำเพาะของมอเตอร์ซึ่งประกอบด้วย จำนวนขั้ว จำนวนสล็อต ความเร็วรอบในการหมุน ดังแสดงในรูปที่ 3.14 และรูปที่ 3.15



รูปที่ 3.13 อัลกอริทึมในการเขียนโปรแกรม



รูปที่ 3.14 รูปคลื่นสัญญาณกระแสเตเตอร์

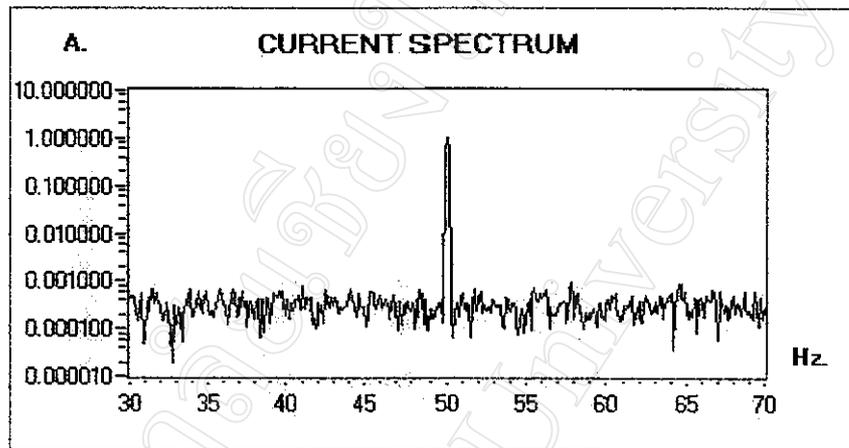
The figure shows a form titled "INPUT DATA" with four input fields:

No load speed (rpm)	Full load speed (rpm)
1495	1450
No. of slots	No. of poles
28	4

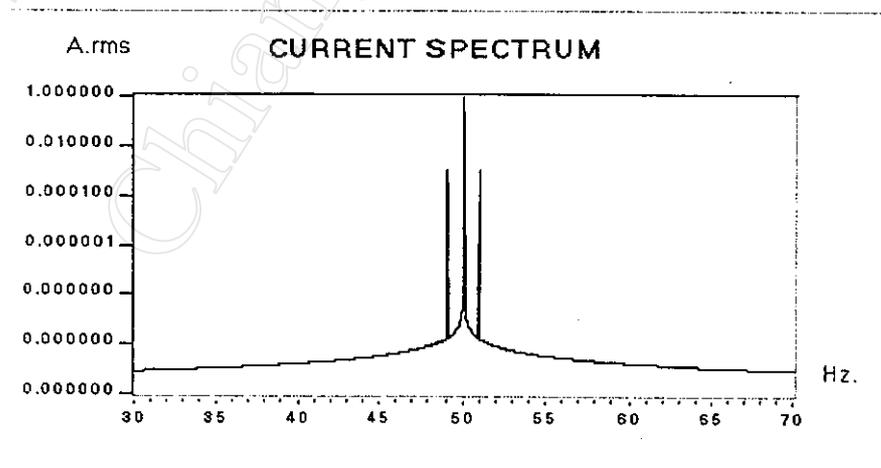
รูปที่ 3.15 ข้อมูลที่ใช้ในการคำนวณ

3.2.2 การแปลงฟูรีเยร์ (Fast Fourier Transform, FFT) เป็นส่วนที่รับข้อมูลกระแสเตเตอร์จากส่วนรับข้อมูลแล้วนำมาทำการเปลี่ยนเป็นโดเมนความถี่ (Frequency domain) เพื่อใช้ในการวิเคราะห์ต่อไป ดังแสดงในรูปที่ 3.16 โดยในส่วนของ Fast Fourier Transform จะใช้ Window Function เป็นเครื่องมือในการปรับปรุงสัญญาณเพื่อให้มีความถูกต้อง

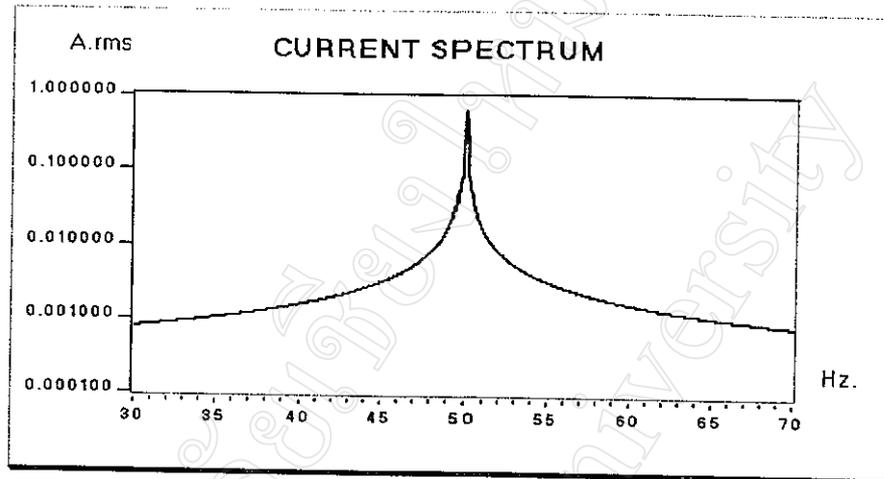
โดยจะเป็นการลดความผิดเพี้ยนที่เกิดขึ้นจากการสุ่มตัวอย่างข้อมูลไม่ครบคาบ ซึ่งจะทำให้เกิด Spectral leakage ดังแสดงในรูปที่ 3.17 และรูปที่ 3.18



รูปที่ 3.16 สเปกตรัมของกระแสที่ได้จากการเปลี่ยนโดเมนเวลาเป็นโดเมนความถี่



รูปที่ 3.17 สเปกตรัมของกระแสเมื่อทำการสุ่มครบคาบ

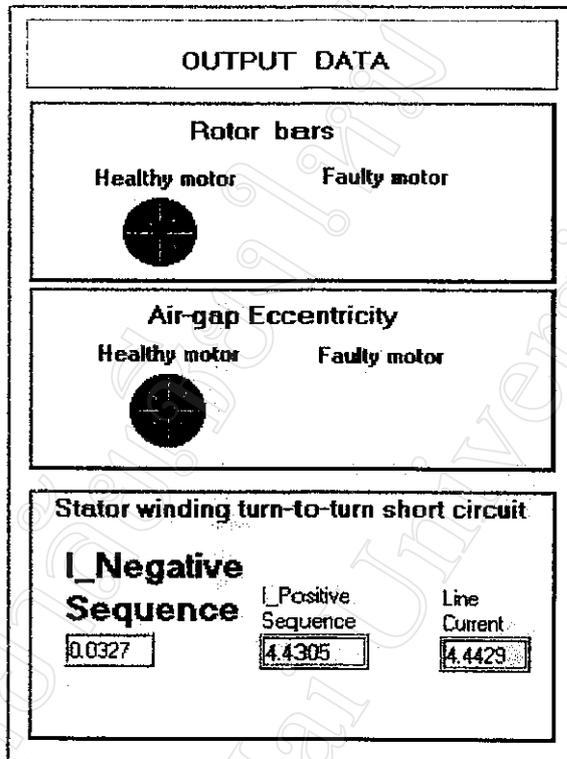


รูปที่ 3.18 สเปกตรัมของกระแสเมื่อทำการซูมไม่ครบคาบ

3.2.3 ส่วนตรวจจับความผิดปกติ แบ่งเป็น 3 ชนิด ดังแสดงในรูปที่ 3.3 คือ

- (1) ส่วนตรวจจับความผิดปกติเนื่องจากการแตกหักของแท่งตัวนำในโรเตอร์ ส่วนนี้จะตรวจจับสเปกตรัมที่ความถี่ $(1 \pm 2s) f_1$
- (2) ส่วนตรวจจับความผิดปกติเนื่องจากความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ ส่วนนี้จะตรวจจับสเปกตรัมที่ความถี่ $f_r \pm f_1$
- (3) ส่วนการคำนวณกระแสลำดับลบ โดยในการคำนวณจะใช้ข้อมูลของกระแสสเตเตอร์เพียง 2 เฟส แล้วทำการคำนวณหากระแสเฟสที่ 3 ได้ เนื่องจากผลรวมของกระแสเฟส A และเฟส B มีค่าเท่ากับกระแสเฟส C

3.2.4 ส่วนแสดงผล ในส่วนนี้การแสดงผลจะแบ่งตามชนิดของความผิดปกติดังกล่าวข้างต้น เพื่อให้ดูเข้าใจง่ายโดยแยกเป็นส่วน ๆ คือ ส่วนของการแตกหักของโรเตอร์ ส่วนของความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และสุดท้ายคือส่วนของกระแสลำดับลบ กระแสลำดับบวก และกระแสที่มอเตอร์ใช้จริง (Line current) ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 ส่วนแสดงผลการวินิจฉัยความผิดปกติของมอเตอร์

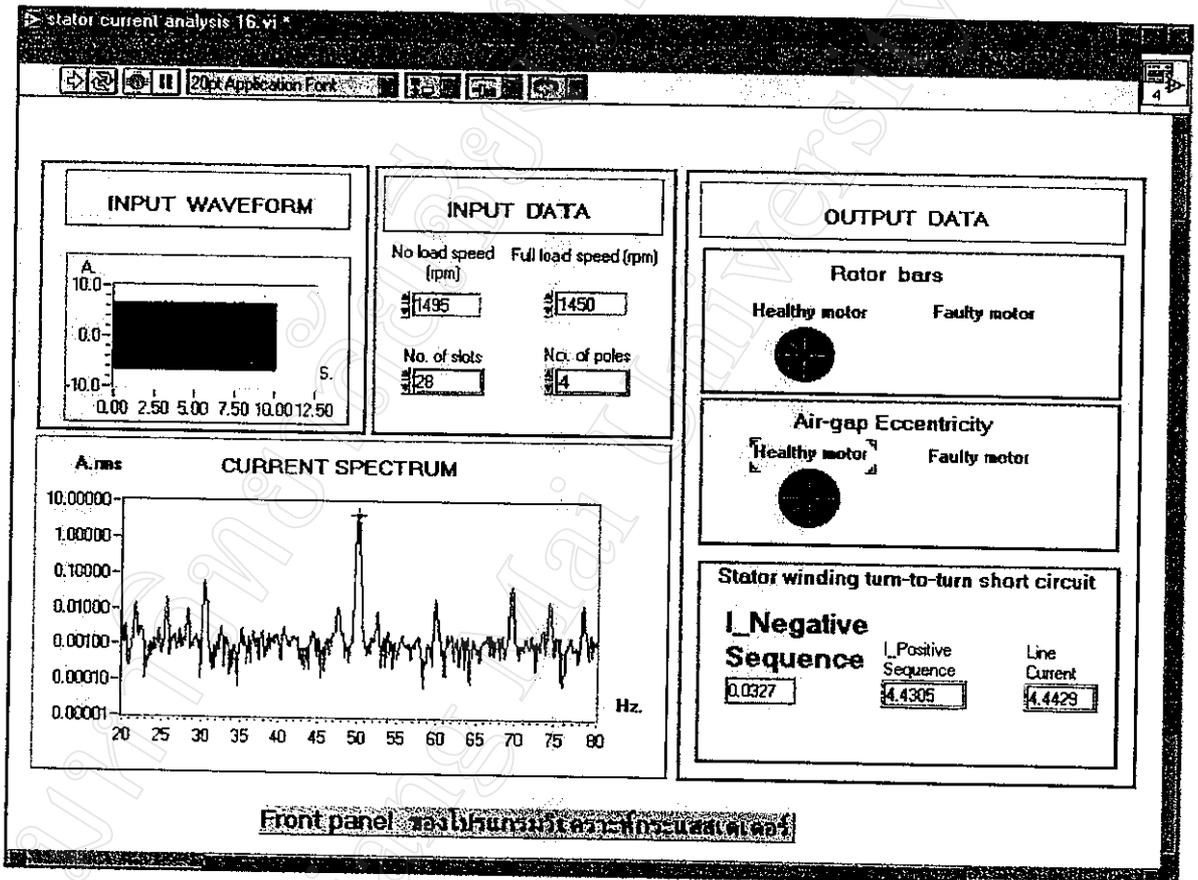
3.3 Front panel และแผนผังของโปรแกรมวิเคราะห์กระแสเตเตอร์ เพื่อตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส

3.3.1 Front panel ของโปรแกรมประกอบด้วย ส่วนของรูปคลื่นกระแสเตเตอร์ ส่วนของข้อมูลที่ใช้ในการคำนวณ ได้แก่ ความเร็วรอบขณะมีโหลดกับไม่มีโหลด จำนวนแท่งตัวนำทั้งหมด จำนวนขั้วแม่เหล็ก ส่วนของสเปกตรัมของกระแสที่ได้จากการแปลงฟาสต์ฟูเรียร์ และส่วนแสดงผลการวินิจฉัยทั้ง 3 ชนิด ดังแสดงในรูปที่ 3.20

การใช้งานโปรแกรมสามารถทำได้ง่าย โดยมีขั้นตอนดังนี้

- (1) ป้อนข้อมูลลงในส่วนของ Input data ประกอบด้วย ความเร็วรอบขณะไม่มีโหลด ความเร็วรอบขณะมีโหลด จำนวนสล็อต และจำนวนขั้ว
- (2) แล้วทำการกดปุ่ม Run
- (3) โปรแกรมจะให้ทำการป้อนข้อมูลกระแสขณะมีโหลด แล้วตามด้วยข้อมูลกระแสขณะไม่มีโหลด

(4) โปรแกรมจะแสดงผลการวินิจฉัยโดยแยกแยะความผิดปกติให้ทั้ง 3 สาเหตุคือ การแตกหักของแท่งตัวนำในโรเตอร์ ความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และการลัดวงจรระหว่างรอบของขดลวด โดยแสดงผลในรูปของกระแสลำดับลบ



รูปที่ 3.20 Front panel ของโปรแกรมวิเคราะห์กระแสสเตเตอร์ (Stator current analysis 16.vi)

3.3.2 แผนผังของโปรแกรม (Block diagram) ประกอบด้วยโปรแกรมย่อย (SubVIs) ต่าง ๆ และในบทนี้จะขอเสนอในส่วนหลัก ๆ ส่วนรายละเอียดขอให้ดูในภาคผนวก ก ซึ่งได้แสดงทุก ๆ ส่วนของ VI และในรูปที่ 3.21 ถึงรูปที่ 3.24 ได้แสดงผังงาน (Flow chart) ขั้นตอนการใช้งานโปรแกรมตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส ผังงานขั้นตอนการตรวจจับการแตกหักของแท่งตัวนำในโรเตอร์ ผังงานขั้นตอนการตรวจจับความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และผังงานขั้นตอนการตรวจจับการลัดวงจรระหว่างรอบของขดลวดสเตเตอร์ สามารถอธิบายผังงานข้างต้นดังนี้

3.3.2.1 ฟังก์ชันของการทำงานของโปรแกรมตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส

จากรูป 3.21 สามารถสรุปขั้นตอนได้ดังนี้

- (1) เปิดโปรแกรม "Stator Current Analysis 16.vi" จาก LabVIEW
- (2) ป้อนข้อมูลที่ต้องใช้ในการคำนวณได้แก่ ความเร็วของมอเตอร์ขณะไม่มีโหลด และขณะมีโหลด จำนวนสล้อต และจำนวนขั้วของมอเตอร์
- (3) กดปุ่ม "Run" จะมีกล่องข้อความให้ทำการป้อนข้อมูลสัญญาณของกระแสขณะมีโหลด ให้ทำการเลือกไฟล์สัญญาณกระแสขณะมีโหลด แล้วกด "ok"
- (4) ทำการป้อนข้อมูลสัญญาณของกระแสขณะไม่มีโหลด ทำการเลือกไฟล์สัญญาณกระแสขณะไม่มีโหลด แล้วกด "ok"
- (5) รอโปรแกรมทำการประมวลผล
- (6) จบขั้นตอนการทำงานของโปรแกรม

3.3.2.2 ฟังก์ชันขั้นตอนการตรวจจับความผิดปกติเนื่องจากการแตกหักของแท่งตัวนำในโรเตอร์

จากรูป 3.22 สามารถสรุปขั้นตอนได้ดังนี้

- (1) รับค่าข้อมูลที่ต้องใช้ในการคำนวณได้แก่ ความเร็วของมอเตอร์ขณะไม่มีโหลด และขณะมีโหลด จำนวนสล้อต และจำนวนขั้วของมอเตอร์
- (2) รับค่ากระแสขณะมีโหลด 1 เฟส
- (3) ปรับปรุงสัญญาณโดยใช้ 7 Term B-Harris window
- (4) การแปลงฟาสต์ฟูเรียร์ (FFT)
- (5) หาค่ากระแสที่ความถี่มูลฐาน (I, f_1)
- (6) หาค่าอัตราส่วนของแอมพลิจูดของความถี่ที่ผิดปกติกับแอมพลิจูดของความถี่มูลฐานที่ $N=4$ แห่ง
- (7) หาค่าช่วงที่ใช้ในการ Scan หาแอมพลิจูดที่ผิดปกติ $N_r \pm 10$
- (8) หาค่าความถี่ในการหมุนของโรเตอร์
- (9) หาค่าแอมพลิจูดที่ผิดปกติที่ความถี่ $(1-2s)f_1$ (I, f_{broken}) ที่ความเร็วในการหมุนของมอเตอร์ขณะมีโหลด ± 10 รอบต่อนาที
- (10) หาค่าอัตราส่วนแอมพลิจูดของความถี่ $(1-2s)f_1$ ต่อแอมพลิจูดที่ความถี่มูลฐาน f_1 ,
($R_{s, measure}$)

- (11) เปรียบเทียบอัตราส่วนของแอมพลิจูดว่ามากกว่า กรณีที่ $N=4$ ใช่หรือไม่
 - 11.1 ใช่ แสดงผล "Faulty"
 - 11.2 ไม่ใช่ แสดงผล "Healthy"
- (12) จบการทำงาน

3.3.2.3 ฟังก์ชันขั้นตอนการตรวจจับความผิดปกติเนื่องจากความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์

จากรูป 3.23 เป็น ซึ่งสามารถสรุปขั้นตอนได้ดังนี้

- (1) รับค่าข้อมูลที่ต้องใช้ในการคำนวณได้แก่ ความเร็วของมอเตอร์ขณะไม่มีโหลด และขณะมีโหลด จำนวนสล็อต และจำนวนขั้วของมอเตอร์
 - (2) รับค่ากระแสขณะมีโหลด 1 เฟส
 - (3) รับค่ากระแสขณะไม่มีโหลด 1 เฟส
 - (4) ปรับปรุงสัญญาณ โดยใช้ 7 Term B-Harris window
 - (5) การแปลงฟาสต์ฟูเรียร์ (FFT)
 - (6) ตรวจสอบว่าเป็นกระแสขณะไม่มีโหลดใช่หรือไม่
 - 6.1 ใช่ ให้หาช่วงของความถี่รอบที่ใช้ในการ Scan หาค่า Lower side-band
 - 6.2 ไม่ใช่ ให้ไปขั้นตอนที่ 7
 - (7) ตรวจสอบว่าเป็นกระแสขณะมีโหลดใช่หรือไม่
 - 7.1 ใช่ ให้หาช่วงของความถี่รอบที่ใช้ในการ Scan หาค่า Upper side-band
 - 7.2 ไม่ใช่ กลับไปขั้นตอนที่ 5
 - (8) หาค่าความเร็วในการหมุนของโรเตอร์เป็นรอบต่อวินาที
 - (9) หาช่วงความถี่ที่ต้องใช้ในการ Scan หา Lower side-band
 - (10) หาช่วงความถี่ที่ต้องใช้ในการ Scan หา Upper side-band
 - (11) หาค่าแอมพลิจูดของ Lower side-band โดยหาค่า Maximum ในช่วงความถี่ข้อ 9
 - (12) หาค่าแอมพลิจูดของ Upper side-band โดยหาค่า Maximum ในช่วงความถี่ข้อ 10
 - (13) ตรวจสอบว่าทำครบทั้งขณะไม่มีโหลดและขณะมีโหลดใช่หรือไม่
 - 13.1 ใช่ ไปทำขั้นตอนที่ 14
 - 13.2 ไม่ใช่ กลับไปขั้นตอนที่ 5
 - (14) ตรวจสอบสเปกตรัมที่ได้ว่าเป็น Upper side-band ใช่หรือไม่

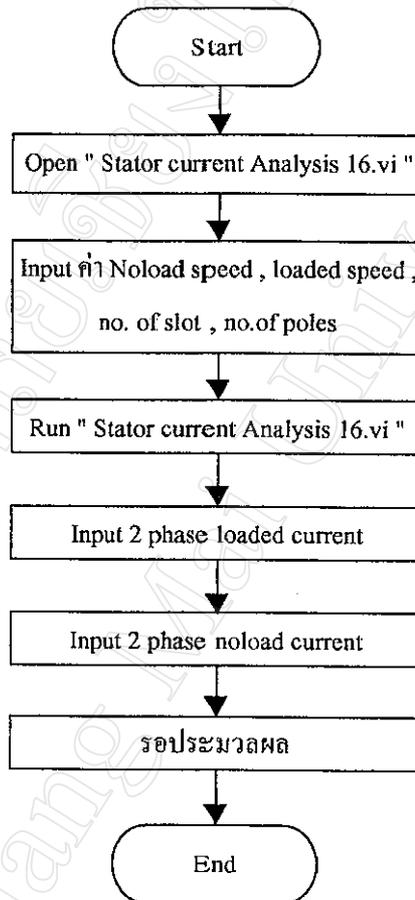
- 14.1 ใช่ ไปทำขั้นตอน 15
- 14.2 ไม่ใช่ ตรวจสอบสเปกตรัมที่ได้ว่าเป็น Lower side-band ใช่หรือไม่
 - 14.2.1 ใช่ ไปทำข้อ 16
 - 14.2.2 ไม่ใช่ กลับไปขั้นตอนที่ 5
- (15) เปรียบเทียบค่าแอมพลิจูดของ Upper side-band ขณะมีโหลดอย่างน้อยกว่า 0.9 เท่าของขณะไม่มีโหลดใช่หรือไม่
 - 15.1 ใช่ แสดงผล "Faulty"
 - 15.2 ไม่ใช่ แสดงผล "Healthy"
- (16) เปรียบเทียบค่าแอมพลิจูดของ Lower side-band ขณะมีโหลดอย่างน้อยกว่า 0.9 เท่าของขณะไม่มีโหลดหรือใช่หรือไม่
 - 16.1 ใช่ แสดงผล "Faulty"
 - 16.2 ไม่ใช่ แสดงผล "Healthy"
- (17) ตรวจสอบว่าทำครบทั้ง Lower และ Upper side-band ใช่หรือไม่
 - 17.1 ใช่ จบการทำงาน
 - 17.2 ไม่ใช่ ไปทำขั้นตอนที่ 5

3.3.2.4 ผังงานขั้นตอนการตรวจจัดการสัณฐานระหว่างรอบของขดลวดสเตเตอร์

จากรูป 3.24 เป็น สามารถสรุปขั้นตอนได้ดังนี้

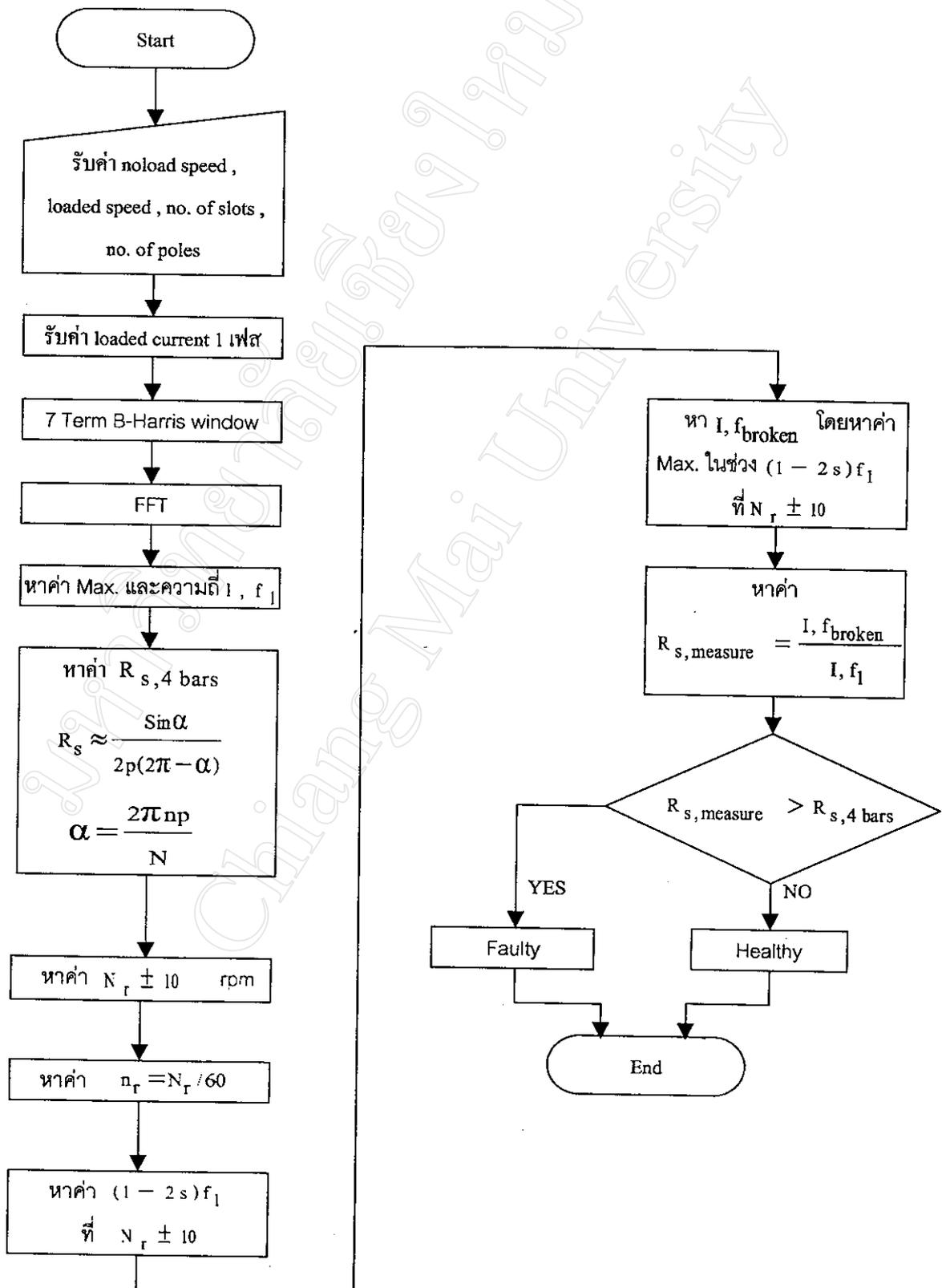
- (1) รับค่ากระแสขณะมีโหลด 2 เฟส
- (2) ปรับปรุงสัญญาณโดยใช้ 7 Term B-Harris window
- (3) การแปลงฟาสต์ฟูเรียร์ (FFT)
- (4) หาค่าแอมพลิจูดและมุมของกระแสที่ความถี่มูลฐาน (I, f) ทั้ง 2 เฟส
- (5) หาค่ากระแส $I_c = I_a + I_b$
- (6) หาค่ากระแสลำดับลบจาก $I_{-2} = 1/3(I_a + a^2 I_b + a I_c)$
- (7) หาค่ากระแสลำดับบวกจาก $I_{+1} = 1/3(I_a + a I_b + a^2 I_c)$
- (8) แสดงผลการคำนวณ
- (9) จบการทำงาน

ผังงานขั้นตอนการใช้งานโปรแกรม



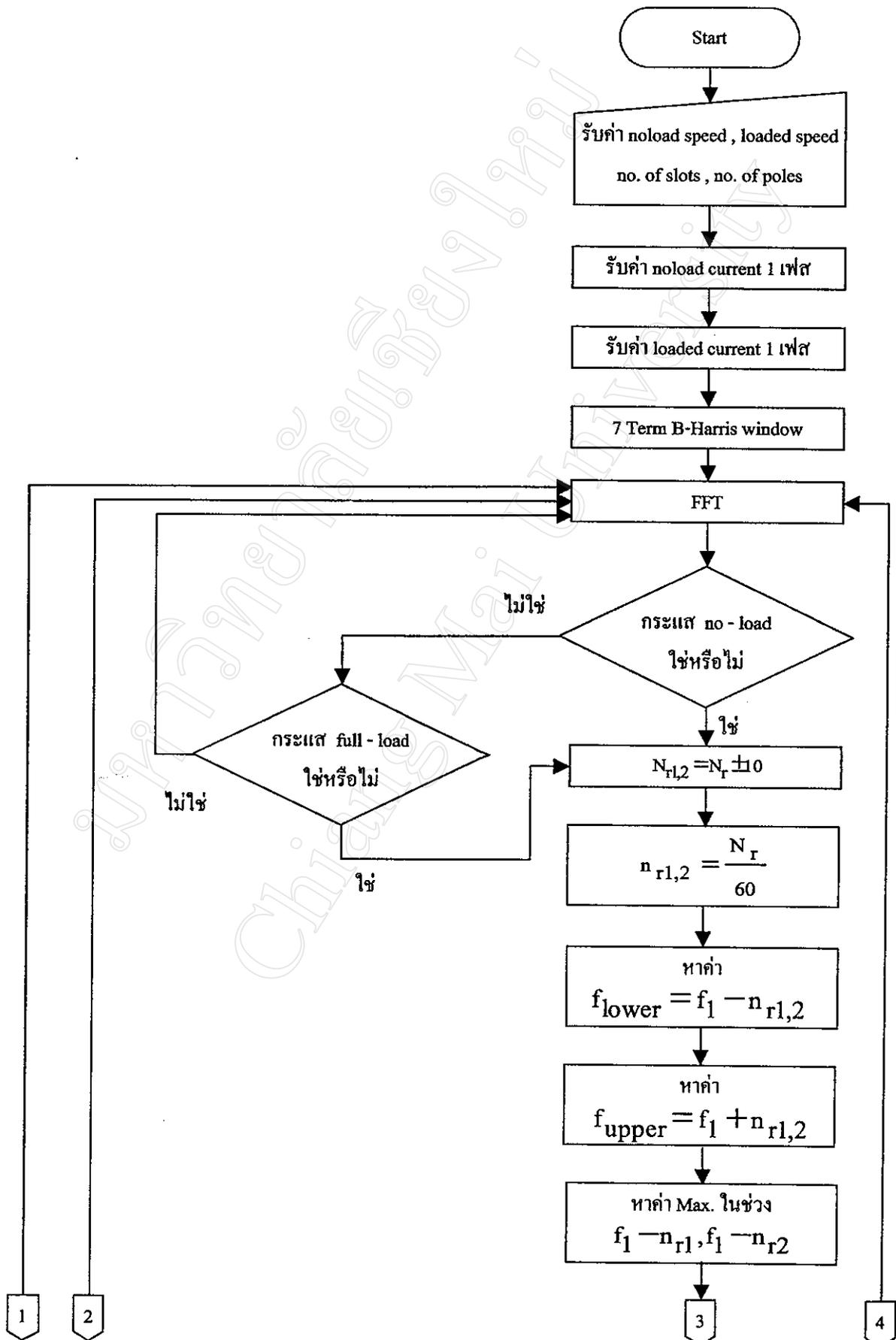
รูปที่ 3.21 ผังงานขั้นตอนการใช้งานโปรแกรม

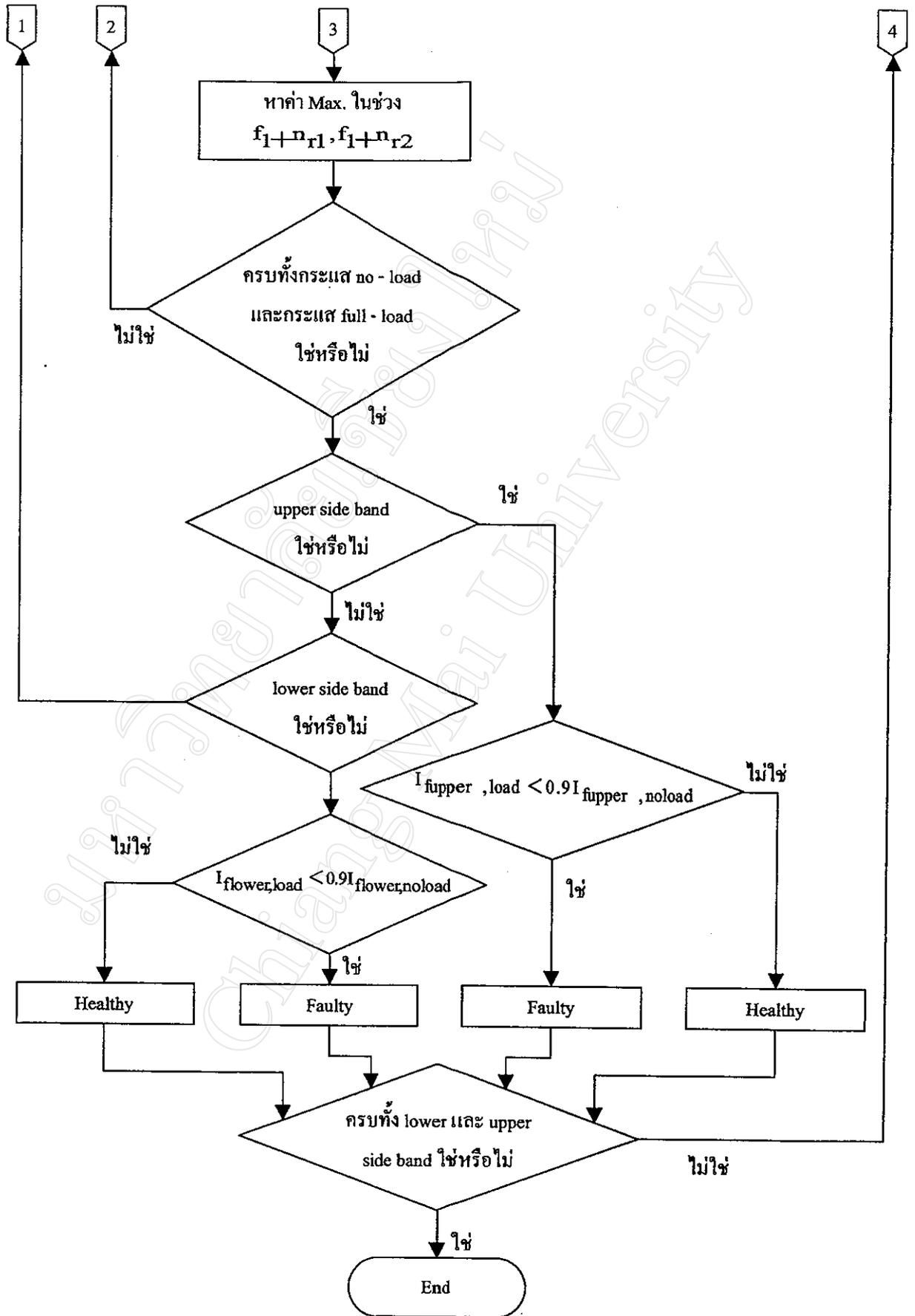
ผังงานขั้นตอนการตรวจจัดการแตกหักของแท่งตัวนำในโรเตอร์



รูปที่ 3.22 ผังงานขั้นตอนการตรวจจัดการแตกหักของแท่งตัวนำในโรเตอร์

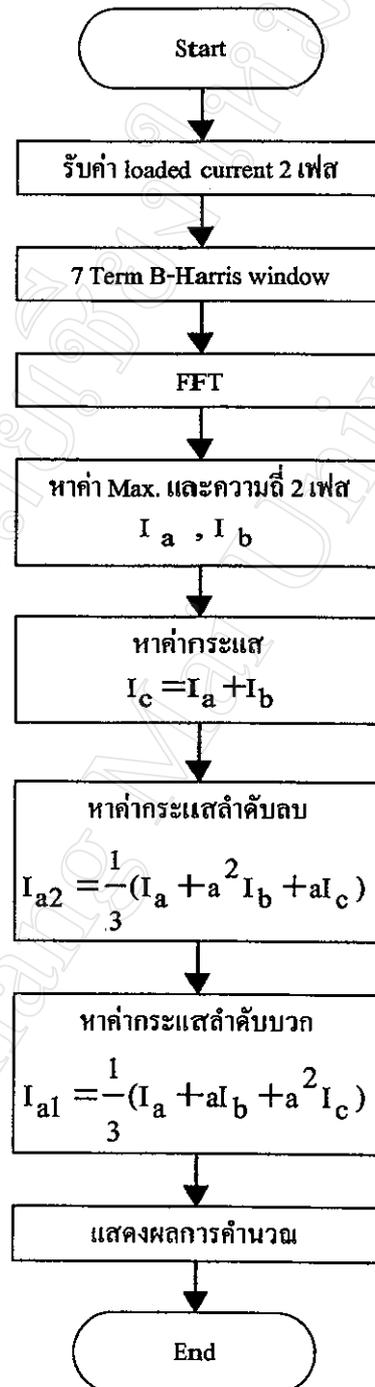
ผังงานขั้นตอนการตรวจจับความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์



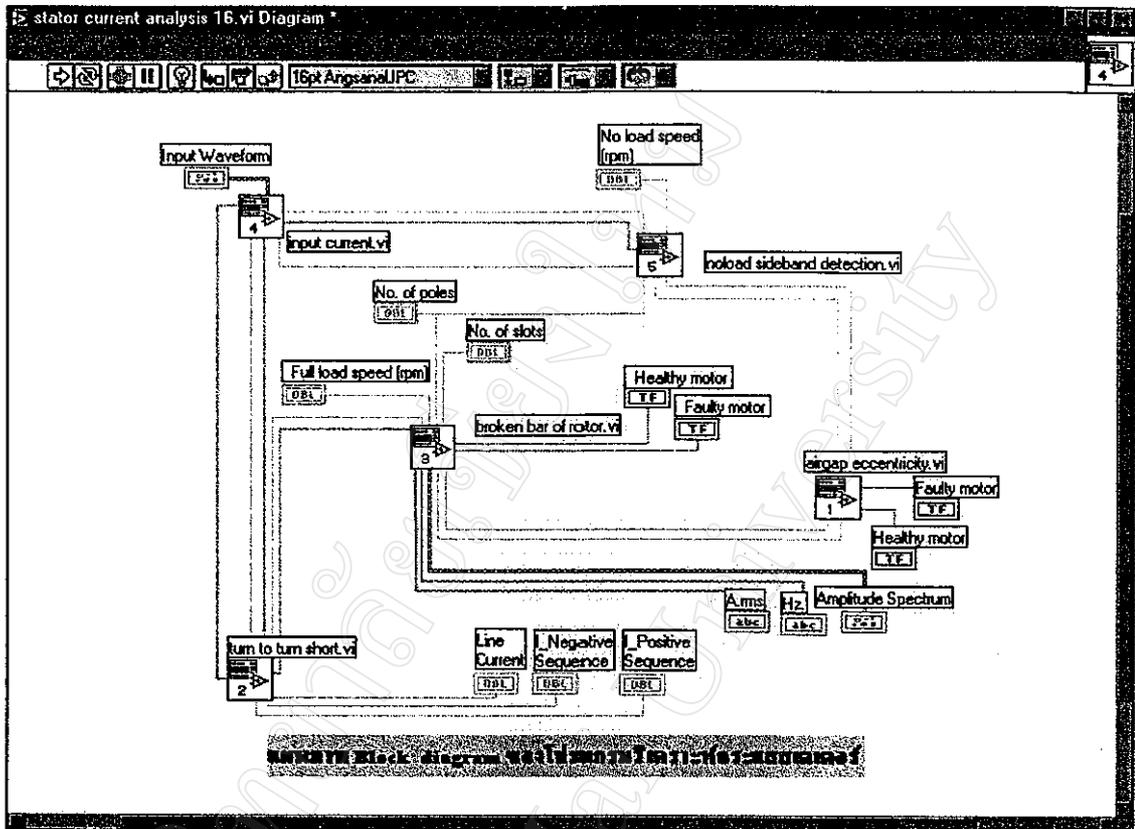


รูปที่ 3.23 ฟังก์ชันขั้นตอนการตรวจจับความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับ โรเตอร์

ผังงานขั้นตอนการตรวจจับการลัดวงจรระหว่างรอบของขดลวดสเตเตอร์



รูป 3.24 ผังงานขั้นตอนการตรวจจับการลัดวงจรระหว่างรอบของขดลวดสเตเตอร์



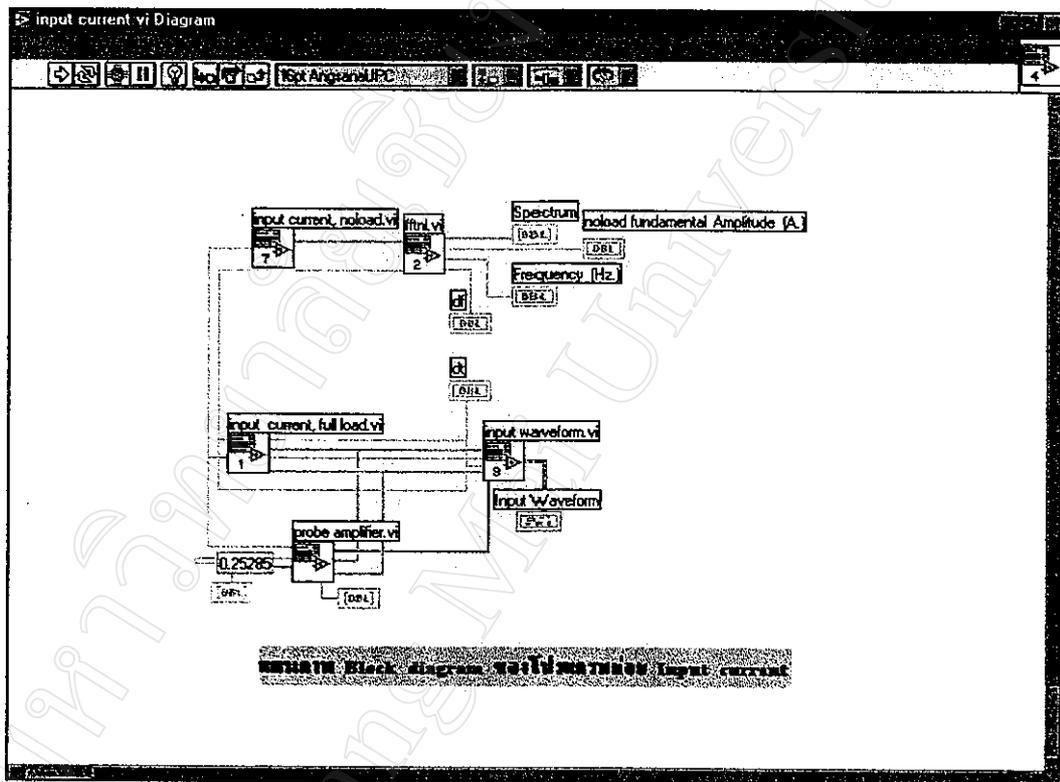
รูปที่ 3.25 แผนผังของโปรแกรมวิเคราะห์กระแสเตเตอร์ (Stator current analysis 16.vi)

ในรูปที่ 3.25 แสดงแผนผังของโปรแกรมวิเคราะห์กระแสเตเตอร์เพื่อตรวจจับความผิดปกติของมอเตอร์เหนี่ยวนำแบบ 3 เฟส โดยในแผนผังของ Stator current analysis 16.vi ประกอบด้วย 5 โปรแกรมย่อย คือ input current.vi, noload sideband detection.vi, broken bar of rotor.vi, airgap eccentricity.vi, turn to turn short.vi และในแต่ละโปรแกรมย่อยก็จะประกอบด้วยโปรแกรมย่อยต่อไปอีก ในที่นี้จะขอพูดถึงเฉพาะการทำงานของ stator current analysis 16.vi และโปรแกรมย่อยทั้ง 5 โปรแกรมข้างต้นเป็นหลัก ส่วนโปรแกรมย่อยอื่น ๆ จะขอกล่าวถึงเฉพาะส่วนที่สำคัญเท่านั้น

3.3.2.1 การทำงานของ Stator current analysis 16.vi

Stator current analysis 16.vi ประกอบด้วย 5 โปรแกรมย่อยดังกล่าวข้างต้น และมีการทำงานดังนี้

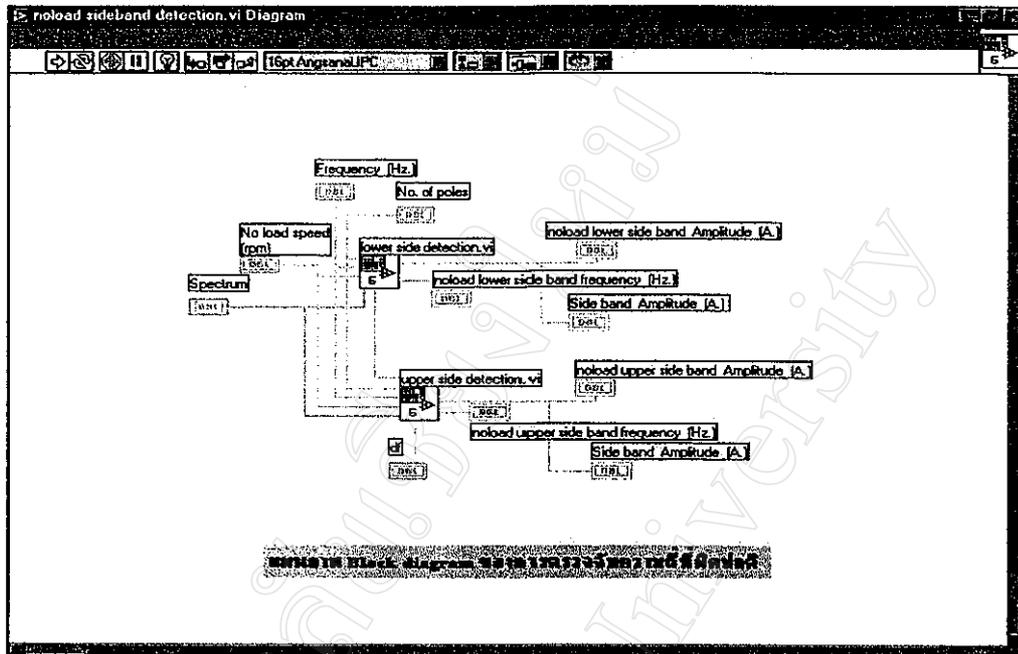
(1) input current.vi เป็นโปรแกรมรับข้อมูลกระแสสแตเตอร์ที่ได้จากคิจิตลอสซิลโลสโคป แล้วนำไปเปลี่ยนเป็นสเปกตรัมของกระแสโดยใช้การแปลงฟาสท์ฟูเรียร์ และปรับปรุงสัญญาณโดยใช้วินโดว์แบบ 7 Term B-Harris และเป็นการลด Spectral leakage ทำให้ได้รับสเปกตรัมของกระแสที่มีความถูกต้องมากที่สุด แล้วส่งสเปกตรัมของกระแสไปให้ noload sideband detection .vi และ turn to turn short .vi และได้แสดงแผนผังดังรูปที่ 3.22



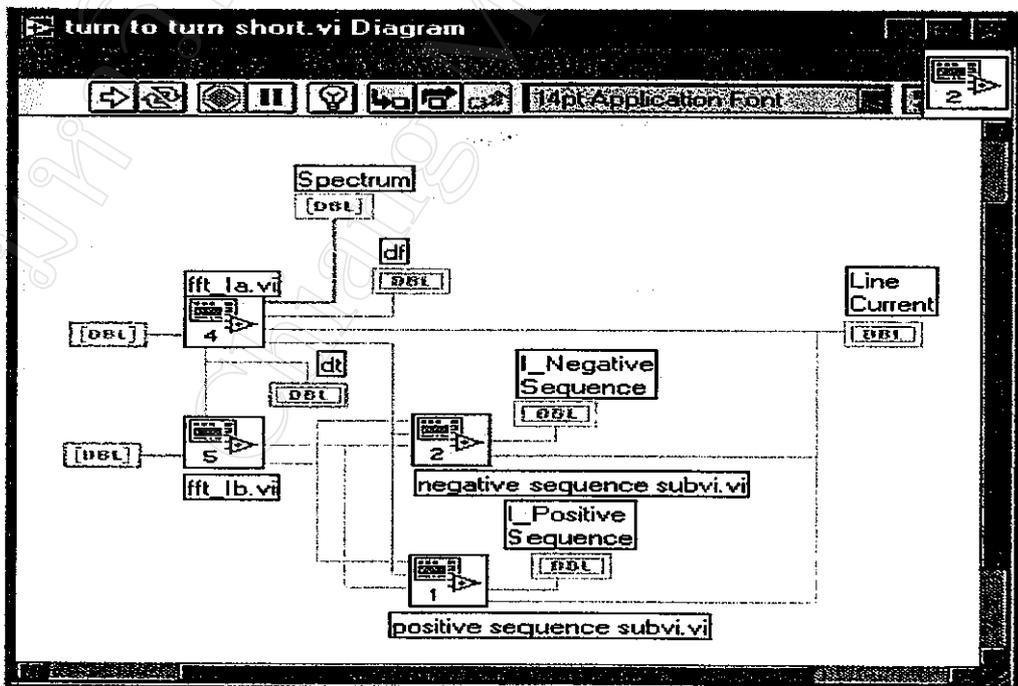
รูปที่ 3.26 แผนผังของ input current .vi

(2) noload sideband detection.vi ทำหน้าที่ตรวจจับ lower และ upper side-band แล้วส่งไปให้กับ airgap eccentricity.vi เพื่อทำการวินิจฉัยความผิดปกติที่เกิดเนื่องจากความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับ โรเตอร์ และได้แสดงแผนผัง ดังรูปที่ 3.27

(3) turn to turn short.vi ทำหน้าที่รับข้อมูลกระแสสแตเตอร์ที่ได้จากคิจิตลอสซิลโลสโคป แล้วนำไปเปลี่ยนเป็นสเปกตรัมของกระแสโดยใช้การแปลงฟาสท์ฟูเรียร์ และปรับปรุงสัญญาณโดยใช้วินโดว์แบบ 7 Term B-Harris แล้วนำไปทำการคำนวณหากระแสลำดับลบ และกระแสลำดับบวก เพื่อวินิจฉัยความผิดปกติที่เกิดขึ้นเนื่องจากการลัดวงจรระหว่างรอบของขดลวดสเตเตอร์ และได้แสดงแผนผังดังรูปที่ 3.28



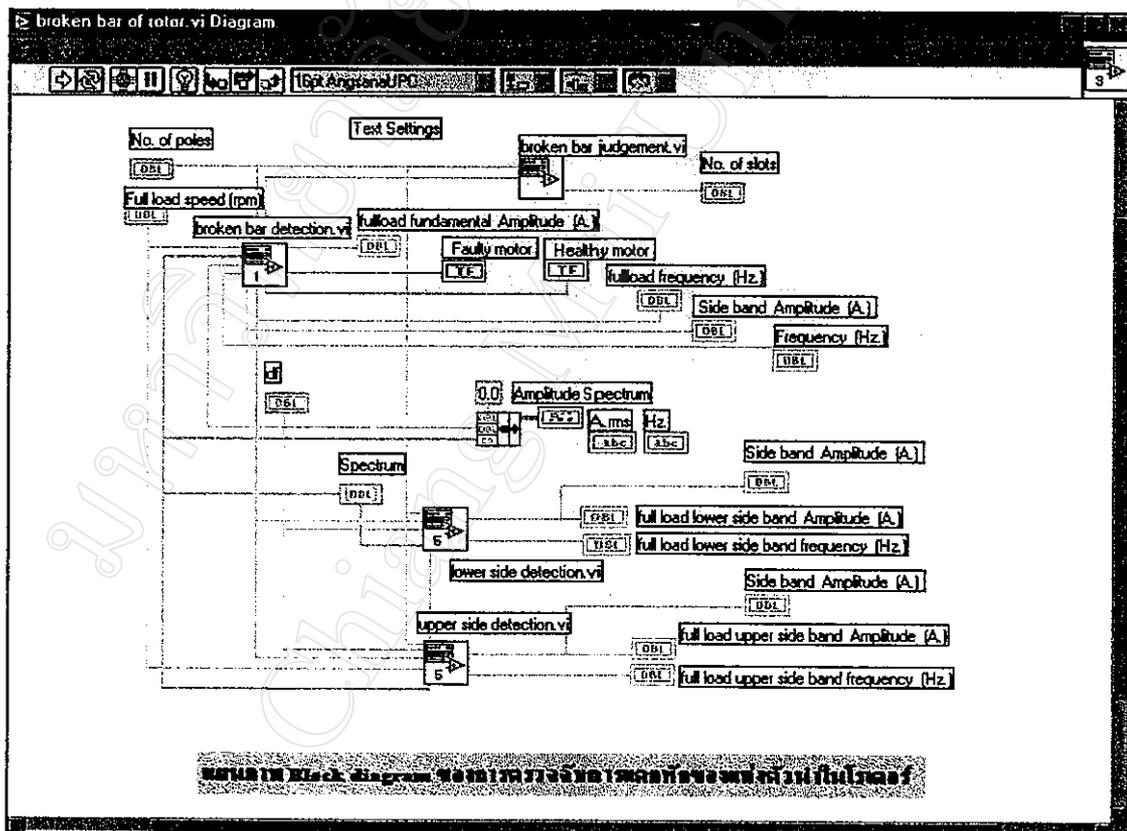
รูปที่ 3.27 แผนผังของ noload sideband detection .vi



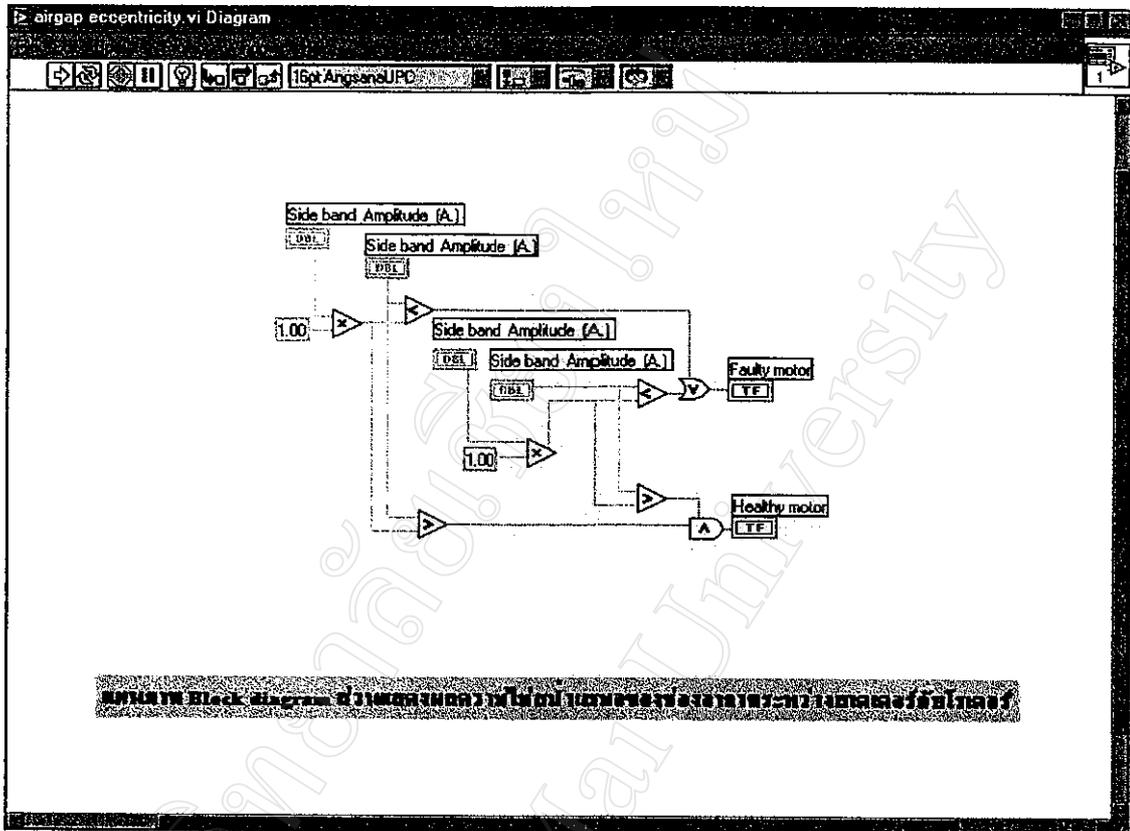
รูปที่ 3.28 แผนผังของ turn to turn short .vi

(6) broken bar of rotor.vi ทำหน้าที่ตรวจจับ lower และ upper side-band ทั้งในกรณีของการแตกหักของแท่งตัวนำโรเตอร์และความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และทำการวินิจฉัยความผิดปกติที่เกิดขึ้นเนื่องจากการแตกหักของแท่งตัวนำในโรเตอร์ และได้แสดงแผนผัง ดังรูปที่ 3.29

(5) airgap eccentricity.vi ทำหน้าที่รับ upper และ lower side-band จาก broken bar of rotor.vi และ no load side band detection และทำการวินิจฉัยความผิดปกติที่เกิดจากความไม่สม่ำเสมอของช่องอากาศระหว่างสเตเตอร์กับโรเตอร์ และได้แสดงแผนผัง ดังรูปที่ 3.30



รูปที่ 3.29 แผนผังของ broken bar of rotor.vi



รูปที่ 3.30 แผนผังของ airgap eccentricity .vi

3.4 การเลือกความถี่ในการสุ่มที่เหมาะสม

จากหัวข้อ 2.4.2 เกี่ยวกับข้อกำหนดเกี่ยวกับการแปลงสัญญาณด้วยการแปลงฟูรีเยร์ กล่าวไว้ว่า ช่วงห่างของความถี่ (Interval frequency, f_0) เท่ากับ $1/NT$

นั่นคือ

$$f_0 = \frac{1}{NT}$$

โดย

N = จำนวนตัวอย่าง

T = ช่วงห่างของเวลาแต่ละจุดตัวอย่าง

$$T = \frac{1}{\text{Sampling rate}}$$

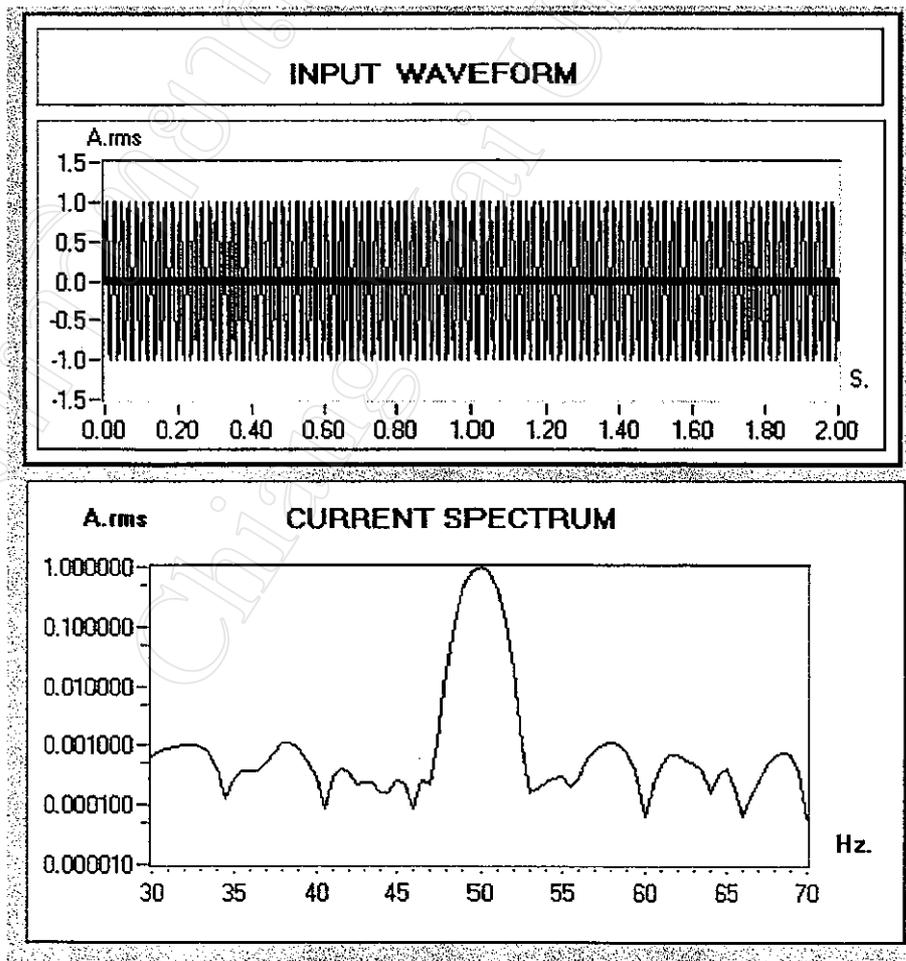
$$f_0 = \frac{\text{Sampling rate}}{N}$$

จะเห็นว่าช่วงห่างของความถี่หรือ Resolution จะขึ้นอยู่กับความถี่ในการสุ่ม (Sampling rate) และจำนวนตัวอย่าง (N) เมื่อจำนวนตัวอย่างมีค่ามากช่วงห่างของความถี่ก็จะมีค่าน้อย นั่นคือความละเอียดของสัญญาณจะมีค่าสูงขึ้น ในกรณีที่พิจารณาที่ความถี่ในการสุ่มเท่ากัน หรือในกรณีที่พิจารณาที่จำนวนตัวอย่างเท่ากันความละเอียดของสัญญาณจะมากขึ้นเมื่อความถี่ในการสุ่มมีค่าต่ำ แต่ในกรณีนี้ถ้าเลือกความถี่ในการสุ่มต่ำเกินไปจะทำให้เกิดความผิดเพี้ยนของสัญญาณที่เป็น โดเมนเวลามากเช่นกัน และในวิทยานิพนธ์นี้ใช้ดิจิตอลออสซิลโลสโคปในการตรวจวัดสัญญาณกระแส เนื่องจากมีความสะดวกในการวัด สามารถวัดกระแสได้ง่าย จัดเก็บข้อมูลได้ง่ายโดยทำการบันทึกข้อมูลลงแผ่นดิสก์แล้วสามารถนำมาประมวลผลภายหลังได้ ซึ่งจำนวนตัวอย่างที่ได้จากการวัดโดยใช้ดิจิตอลออสซิลโลสโคปจะคงที่ที่ 20,000 ตัวอย่าง เนื่องจากหน่วยความจำภายในมีค่าคงที่ และสามารถสรุปความสัมพันธ์ของความถี่ในการสุ่ม จำนวนตัวอย่าง ความละเอียดของโดเมนความถี่ และจำนวนคาบข้อมูลที่วัดได้ ที่ความถี่ในการสุ่มต่าง ๆ กัน

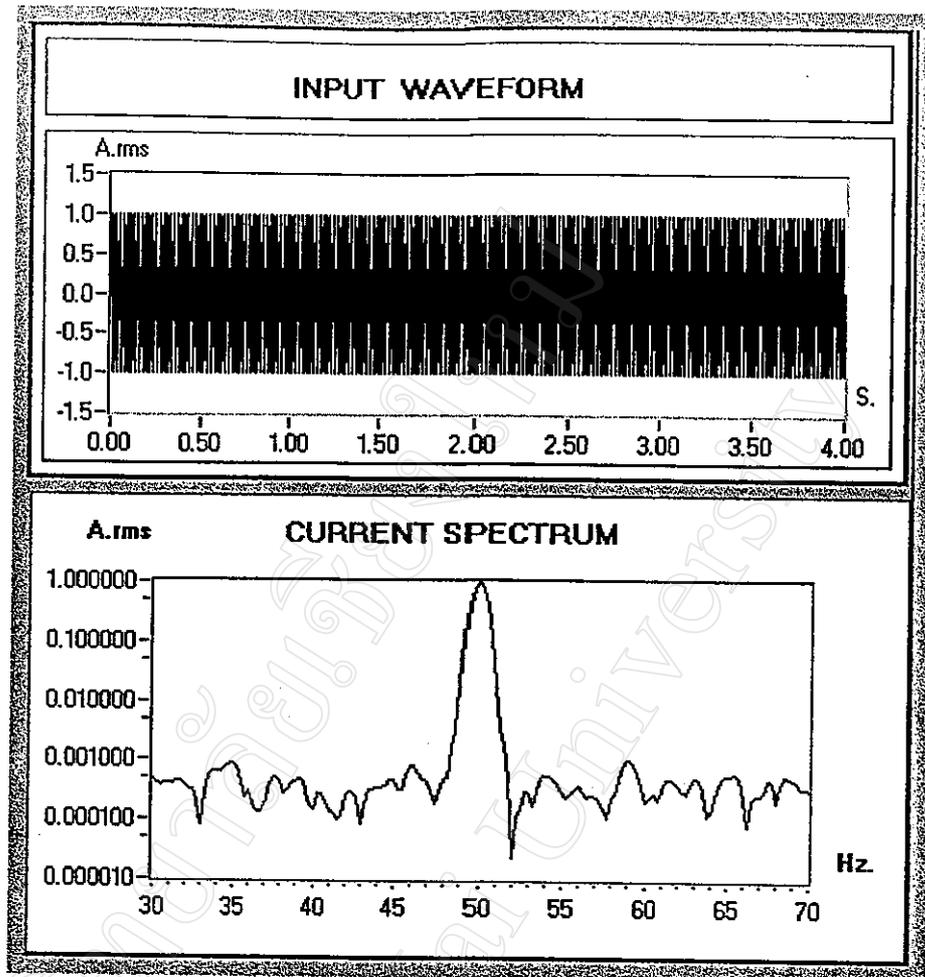
ตารางที่ 3.2 ความสัมพันธ์ของความถี่ในการสุ่ม จำนวนตัวอย่าง ความห่างของความถี่ และจำนวนคาบของข้อมูล

ความถี่ในการสุ่ม (ตัวอย่างต่อวินาที)	จำนวนตัวอย่าง (ตัวอย่าง)	ความห่างของความถี่ f_0 (Resolution) (เฮิรตซ์)	จำนวนเวลาที่วัดได้ (วินาที)
200	20.000	0.010	100
500	20.000	0.025	40
1.000	20.000	0.050	20
2.000	20.000	0.100	10
5.000	20.000	0.250	4

จากตารางที่ 3.2 จะเห็นว่าที่ความถี่ในการสุม 200 ตัวอย่างต่อวินาทีความละเอียดของสเปกตรัมจะมีค่าสูงสุดคือ 0.010 เฮิรต์ แต่ความถูกต้องของสัญญาณที่เป็นโคเมนเวลาจะมีค่าต่ำเนื่องจากที่ความถี่ในการสุม 200 ตัวอย่างต่อวินาที มีค่าเท่ากับ 4 ตัวอย่างต่อไซเคิล และในกรณีที่ความถี่ในการสุม 20,000 ตัวอย่างต่อวินาที ถึงแม้กรณีนี้ความละเอียดของสัญญาณโคเมนเวลาจะมีมากแต่ความห่างของความถี่จะมากทำให้สเปกตรัมที่ได้มีความละเอียดต่ำ ดังนั้นในการเลือกความถี่ในการสุมที่เหมาะสม ในวิทยานิพนธ์นี้ได้ทำการจำลองโดย จำลองกระแสที่ 1 แอมป์ 50 เฮิรต์ 0.005 แอมป์ 49 เฮิรต์และ 51 เฮิรต์ แล้วทำการวัดค่าความผิดพลาดของเฟสและแอมพลิจูด พบว่าที่ความถี่ในการสุม 2,000 ตัวอย่างต่อวินาที ให้ค่าเฟสและแอมพลิจูดที่ถูกต้องมากที่สุดคือ มีความผิดพลาดของแอมพลิจูด 1.33 เปอร์เซ็นต์ ดังแสดงในตารางที่ 3.3 และได้แสดงรูปการจำลองที่ความถี่ในการสุมต่าง ๆ กัน ดังรูปที่ 3.31 - 3.35



รูปที่ 3.31 สเปกตรัมของกระแสเมื่อความถี่ในการสุม 10,000 ตัวอย่าง/วินาที

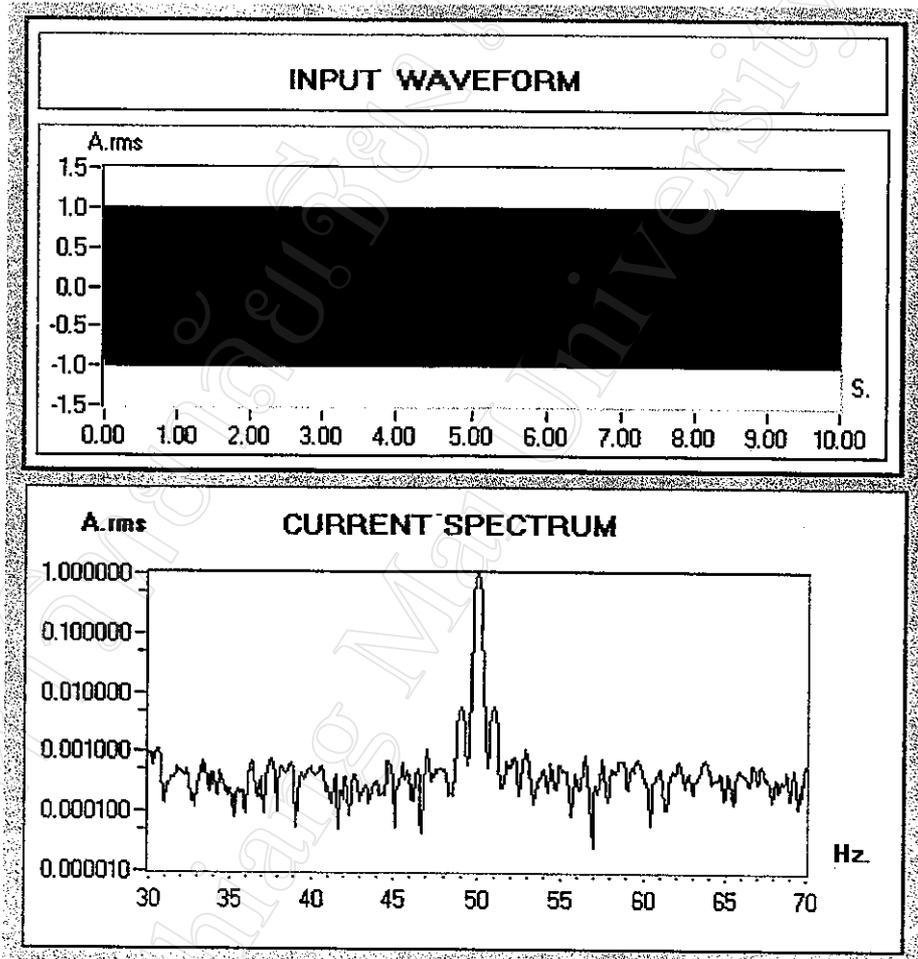


รูปที่ 3.32 สเปกตรัมของกระแสเมื่อความถี่ในการสุ่ม 5,000 ตัวอย่าง/วินาที

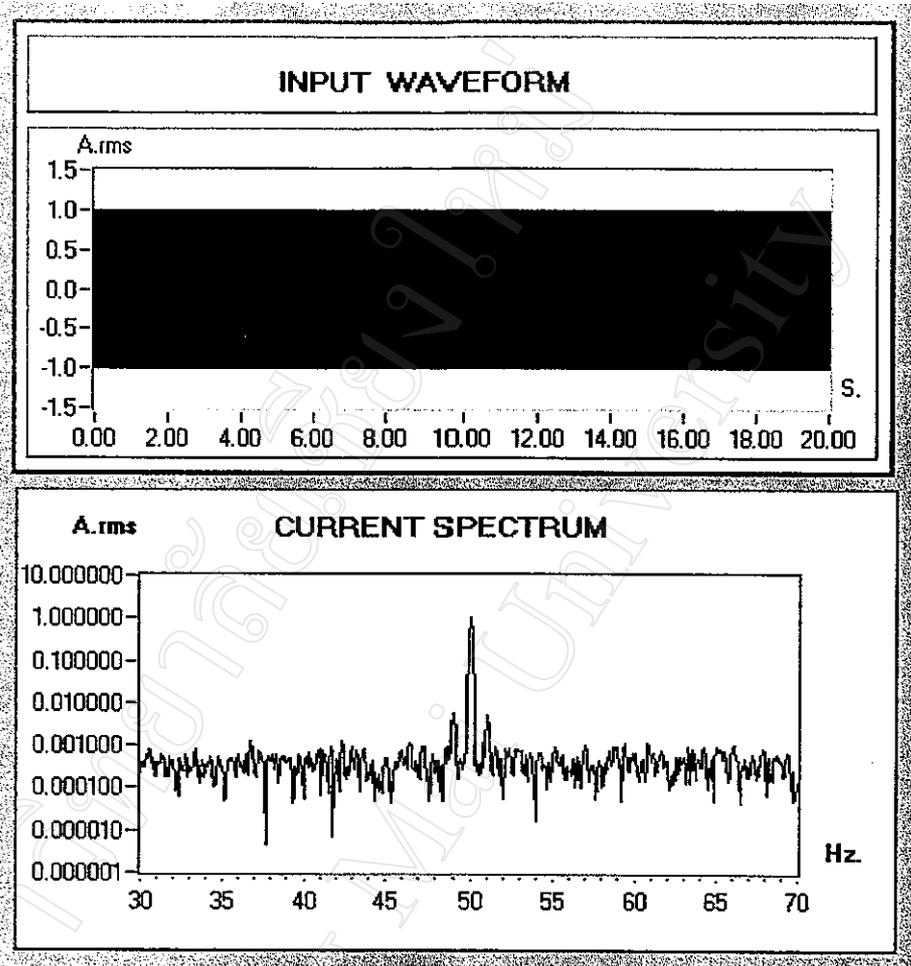
ตารางที่ 3.3 เปอร์เซนต์ความผิดพลาดของแอมพลิจูดและความถี่ที่ความถี่ในการสุ่มต่างๆ

ความถี่ในการสุ่ม (ตัวอย่างต่อวินาที)	แอมพลิจูด		ความถี่	
	ค่าที่วัดได้ (แอมป์)	ค่าเปอร์เซนต์ ความผิดพลาด (%)	ค่าที่วัดได้ (เฮิร์ต)	ค่าเปอร์เซนต์ ความผิดพลาด (%)
500	0.8963	10.37	-49.02	0.04
1.000	0.9436	5.64	49.02	0.04
2.000	0.9867	1.33	49.02	0.04
5.000	N A	N/A	N/A	N/A
10.000	N A	N/A	N/A	N/A

** จำลองที่ความถี่ 50 เฮิร์ต 100 แอมป์ และ 49 เฮิร์ต 1 แอมป์ และทำการวัดที่สเปกตรัม 49 เฮิร์ต



รูปที่ 3.33 สเปกตรัมของกระแสเมื่อความถี่ในการสุ่ม 2,000 ตัวอย่าง/วินาที



รูปที่ 3.34 สเปกตรัมของกระแสเมื่อความถี่ในการสุ่ม 1,000 ตัวอย่าง/วินาที

จากรูปที่ 3.31-3.34 แสดงสเปกตรัมของกระแสที่ทำการจำลองที่ความถี่ในการสุ่มต่าง ๆ จะเห็นว่าในกรณีที่ความถี่ในการสุ่มสูง ความละเอียดของสเปกตรัมจะมีค่าต่ำ (จำนวนข้อมูลมีค่าคงที่ เท่ากับ 20,000 ข้อมูล) ตามความสัมพันธ์ที่กล่าวข้างต้น และกรณีที่ความถี่ 5,000 และ 10,000 ตัวอย่างต่อวินาที ไม่สามารถตรวจจับสเปกตรัมที่ 49 เฮิร์ต และ 51 เฮิร์ตได้ เนื่องจากความห่างของความถี่มีค่ามากเกินไป ส่วนในกรณีที่ความถี่ในการสุ่ม 2,000 และ 1,000 ตัวอย่างต่อวินาที สามารถตรวจจับสเปกตรัมที่ความถี่ดังกล่าวได้ โดยค่าความผิดพลาดของแอมพลิจูดที่ความถี่ในการสุ่ม 2,000 ตัวอย่างต่อวินาทีมีค่าต่ำที่สุด