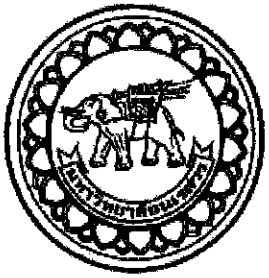


# การพัฒนาอัลกอริทึมในการตรวจสอบสับปะรด

## Algorithm Development for checking a pineapple

นายประเมษฐ ชำนาญยา รหัส 44370245
นายเชษฐา โทสวนจิต รหัส 44370583
นายธนภาค ธนาเอกภิญโญ รหัส 46370227
นายกฤษณะ อัศวกุลเดช รหัส 47361795

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ 25 / W.A. 2553 /
เลขทะเบียน 15001953
เลขเรียกหนังสือ ปร.
มหาวิทยาลัยนเรศวร ป.963 2550

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร
ปีการศึกษา 2550

# ใบรับรองโครงงานวิศวกรรม

| | | |
|---|---|---|
| **หัวข้อโครงงาน** | การพัฒนาอัลกอริทึมในการตรวจสอบสับปะรด | |
| **ผู้ดำเนินโครงงาน** | นายประเมษ ชำนาญยา | รหัส 44370245 |
| | นายเชษฐา โทสวนจิต | รหัส 44370583 |
| | นายธนภาค ธนาเอกภิญโญ | รหัส 46370227 |
| | นายกฤษณะ อัศวกุลเดช | รหัส 47361795 |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มแม่น | |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ | |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ | |
| **ปีการศึกษา** | 2550 | |

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้โครงงานฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์

คณะกรรมการสอบโครงงานวิศวกรรม

.....ประธานกรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มแม่น)

.....กรรมการ

(ดร.ไพศาล มุณีสว่าง)

.....กรรมการ

(ดร.สมพร เรืองสินชัยวานิช)

| | |
|---|---|
| **หัวข้อโครงงาน** | การพัฒนาอัลกอริทึมในการตรวจสอบสับปะรด |
| **ผู้ดำเนินโครงงาน** | นายประเมษฐ ชำนาญยา รหัส 44370245 |
| | นายเชษฐา โทสวนจิต รหัส 44370583 |
| | นายธนภาค ธนาเอกภิญโญ รหัส 46370227 |
| | นายกฤษณะ อัศวกุลเดช รหัส 47361795 |
| **อาจารย์ที่ปรึกษา** | ผู้ช่วยศาสตราจารย์ ดร. สุชาติ แย้มแม่น |
| **สาขาวิชา** | วิศวกรรมคอมพิวเตอร์ |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2550 |


## บทคัดย่อ

โครงงานนี้ พัฒนาโปรแกรมสำหรับการวิเคราะห์เสียงเคาะที่เปลือกของสับปะรดในการคัดเลือกรสชาติของสับปะรด 3 ลักษณะ กล่าวคือ หวาน เปรี้ยว และเน่า โดยการใช้วิธีการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform หรือ FFT) ร่วมกับวิธีสหสัมพันธ์ข้าม (Cross-Correlation Function หรือ CCF) เพื่อเปรียบเทียบค่าความถี่สูงสุดของเสียงเคาะที่เปลือกกับเสียงต้นแบบ

จากผลการทดลอง โปรแกรมที่พัฒนาขึ้นสามารถนำไปใช้ในการวิเคราะห์เสียงเคาะที่เปลือกของสับปะรดและแยกรสชาติของสับปะรดได้ถูกต้องประมาณ 80% ส่วนผลการวิเคราะห์ที่ผิดพลาดเกิดจากไม่ได้จัดเก็บเสียงเคาะต้นแบบจากสับปะรดที่มีรสชาติเน่าบางส่วนปนกับรสชาติหวานบางส่วนกระจายตามเนื้อของสับปะรด

อย่างไรก็ตาม โปรแกรมที่พัฒนาขึ้นนี้ สามารถปรับปรุงวิธีการวิเคราะห์ในอนาคตได้โดยการใช้หลักสถิติเข้ามาช่วยทำให้ผลการวิเคราะห์นั้นได้ถูกต้องมากขึ้น

| | | |
|---|---|---|
| **Project Title** | Algorithm Development for checking a pineapple | |
| **Name** | Mr. Pramed Chomnanya | ID. 44370245 |
| | Mr. Chetta Tosounchit | ID. 44370583 |
| | Mr. Thanapark Thanaekpinyo | ID. 46370227 |
| | Mr. Kritsana Asawakuladet | ID. 47361795 |
| **Project Advisor** | Assistant Professor Suchart Yammen, Ph.D. | |
| **Major** | Computer Engineering | |
| **Department** | Electrical and Computer Engineering | |
| **Academic Year** | 2007 | |



## ABSTRACT

This project is conducted to develop a program for analyzing a peel beating voice of a pineapple. In selecting pineapple, there are three features of favor: sweet, sour and rotten. Fast Fourier Transform or FFT and Cross-Correlation Function or CCF is proposed to compare the highest frequency value of the beating voice with that to the model beating voice.

From the experiment, the developed program can be used to analyze the peel beating voice and to classify the favor of pineapple up to 80% correctly. The failed analysis result is caused by not recording the model beating voice of some pineapples that taste sweet and rotten.

However, in the future work the developed program can be used to improve the analysis method using statistics for the most accurate results.

# กิตติกรรมประกาศ

โครงงานฉบับนี้ จะไม่สามารถสำเร็จลงได้หากไม่ได้รับความสนับสนุนจากบุคคลจำนวนมาก ก่อนอื่นผู้จัดทำโครงงานใคร่ขอขอบคุณ ผู้ช่วยศาสตราจารย์ ดร.สุชาติ แย้มแม่น อาจารย์ที่ปรึกษาโครงงานด้านวิศวกรรมศาสตร์ ที่ให้ความสนับสนุนในทำโครงงาน ตลอดจนให้คำชี้แนะและมอบแนวทางการแก้ไขปัญหาของโครงงานเพื่อให้โครงงานสามารถดำเนินต่อไปได้ด้วยดีตลอดมา

ผู้จัดทำโครงการใคร่ขอกราบขอบพระคุณ อาจารย์ ดร.ไพศาล มุณีสว่าง และ อาจารย์ ดร.สมพร เรืองสินชัยวานิช ที่กรุณาเสียสละเวลาอันมีค่าในการตรวจสอบเนื้อหาของโครงงานฉบับนี้ และให้ความกรุณาเป็นกรรมการในการสอบโครงงาน

ในท้ายที่สุดนี้ ผู้จัดทำโครงการใคร่ขอกราบขอบพระคุณ บิดา มารดา ที่ให้กำลังใจช่วยเหลือในด้านต่างๆ ทั้งทางด้านปัญหาส่วนตัว ปัญหาด้านการเรียน และปัญหาในการทำงานต่างๆ ผู้จัดทำโครงการจะไม่ลืมพระคุณของท่านทั้งสองตลอดชั่วชีวิต

# สารบัญ

หน้า

บทคัดย่อภาษาไทย ..... ก

บทคัดย่อภาษาอังกฤษ ..... ข

กิตติกรรมประกาศ ..... ค

สารบัญ ..... ง

สารบัญตาราง ..... ช

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ

1.1 ที่มาและความสำคัญของโครงงาน ..... 1

1.2 วัตถุประสงค์ของโครงงาน ..... 1

1.3 ขอบเขตของโครงงาน ..... 1

1.4 ขั้นตอนการดำเนินงาน ..... 2

1.5 ผลที่คาดว่าจะได้รับ ..... 3

1.6 งบประมาณที่ใช้ ..... 3

บทที่ 2 บทความที่เป็นประโยชน์ต่อโครงงานและทฤษฎีที่เกี่ยวข้อง

2.1 เรื่องทั่วไปของสับปะรด ..... 4

2.2 บทความที่เป็นประโยชน์ต่อโครงงาน ..... 6

2.2.1 ความถี่ธรรมชาติ ..... 6

2.2.2 Investigation of change exponential frequency index as related to durian maturity..... 7

2.2.3 การศึกษาการใช้คุณสมบัติของเสียงเพื่อตรวจสอบและ คัดแยกอาหารกระป๋อง 8

2.3.4 การแปลงฟูเรียร์ไม่ต่อเนื่อง ..... 9

2.3.5 ฟังก์ชันสหสัมพันธ์ข้าม ..... 13

บทที่ 3 การออกแบบและการวิเคราะห์

3.1 หลักการทำงานของโปรแกรม MATLAB เพื่อใช้ในการวิเคราะห์เสียง..... 17

3.2 การเก็บข้อมูลสัญญาณเสียงเคาะผลสับปะรด ..... 18

# สารบัญ (ต่อ)

หน้า

3.3 การอ่านข้อมูลที่บันทึกไว้ และหาขอบเขตของสัญญาณเสียง ..... 19
3.4 การวิเคราะห์สัญญาณในโดเมนเวลา ..... 20
3.5 การวิเคราะห์สัญญาณในโดเมนความถี่ ..... 22
3.6 การออกแบบโปรแกรมสำหรับการคัดแยกสับปะรด ..... 24
3.7 การวิเคราะห์ปัจจัยที่มีผลต้อข้อมูล ..... 28
3.8 สรุปผลจากการวิเคราะห์ปัจจัยที่มีผลต่อข้อมูลที่จะนำมาวิเคราะห์ ..... 33

บทที่ 4 การทดลอง

4.1 การทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ ..... 34
4.2 การวิเคราะห์เสียงด้วยวิธีแปลงฟูเรียร์แบบเร็ว โดยการใช้โปรแกรม MATLAB ..... 35
4.3 การทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ด้วยวิธีแปลงฟูเรียร์แบบเร็ว..... 36
4.4 การทดสอบสับปะรดตัวอย่างกับสับปะรดต้นแบบด้วยการใช้โปรแกรม ..... 40
4.5 ปัจจัยที่มีผลต่อการทดสอบที่ทำให้ได้ผลที่ผิดพลาด ..... 46
4.6 การวิเคราะห์เสียงเคาะผลสับปะรดโดยวิธีสหสัมพันธ์ข้าม (Cross correlation) ..... 47
4.6.1 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดหวาน..... 48
4.6.2 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดเปรี้ยว ..... 53
4.6.3 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดเน่า ..... 58
4.6.4 การทดลองเพื่อหาหลักเกณฑ์ในการวิเคราะห์เสียง ..... 60
4.6.5 หลักเกณฑ์ในการวิเคราะห์ ..... 63
4.7 ผลการเปรียบเทียบโดยใช้วิธีแปลงฟูเรียร์แบบเร็วกับวิธีสหสัมพันธ์ข้าม..... 68

บทที่ 5 สรุปและข้อเสนอแนะ

5.1 สรุปผลโครงงาน ..... 71
5.2 ข้อเสนอแนะ ..... 71

เอกสารอ้างอิง ..... 72

# สารบัญ (ต่อ)

หน้า

ภาคผนวก

ภาคผนวก ก. ..... 73

ภาคผนวก ข. ..... 88

ประวัติผู้เขียน ..... 92

มหาวิทยาลัยนเรศวร

# สารบัญตาราง

ตารางที่ หน้า

1.1 ตารางแสดงแผนการดำเนินงาน ..... 2

4.1 ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดหวาน ..... 40

4.2 ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดเปรี้ยว ..... 41

4.3 ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดผลเน่า ..... 42

4.4 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 1 ..... 43

4.5 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 2 ..... 44

4.6 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 3 ..... 44

4.7 ใช้ File Sweetest 1.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 47

4.8 ใช้ File Sweetest 2.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 47

4.9 ใช้ File Sweetest 3.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 48

4.10 ใช้ File Sweetest 4.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 48

4.11 ใช้ File Sweetest 5.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 49

4.12 ใช้ File Sweetest 6.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 49

4.13 ใช้ File Sweetest 7.wav เป็นต้นแบบ ของสับปะรดผลหวาน ..... 50

4.14 ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบของผลหวาน ..... 50

4.15 ใช้ File Sour 1.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 52

4.16 ใช้ File Sour 2.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 52

4.17 ใช้ File Sour 3.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 53

4.18 ใช้ File Sour 4.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 53

4.19 ใช้ File Sour 5.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 54

4.20 ใช้ File Sour 6.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 54

4.21 ใช้ File Sour 7.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว ..... 55

4.22 ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบ ของผลเปรี้ยว ..... 55

4.23 ใช้ File Exp 1.wav เป็น ต้นแบบ ของสับปะรดผลเน่า ..... 57

4.24 ใช้ File Exp 2.wav เป็น ต้นแบบ ของสับปะรดผลเน่า ..... 57

4.25 ใช้ File Exp 3.wav เป็น ต้นแบบ ของสับปะรดผลเน่า ..... 57

4.26 ใช้ File Exp 4.wav เป็น ต้นแบบ ของสับปะรดผลเน่า ..... 58

# สารบัญตาราง(ต่อ)


| ตารางที่ | หน้า |
|---|---|
| 4.27 ใช้ File Exp 5.wav เป็น ต้นแบบ ของสับปะรดผลเน่า | 58 |
| 4.28 ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบ ของผลเน่า | 58 |
| 4.29 ข้อมูลที่ได้จากการนำสับปะรดหวานมาเทียบกับสับปะรดต้นแบบ | 60 |
| 4.30 ข้อมูลที่ได้จากการนำสับปะรดเปรี้ยวมาเทียบกับสับปะรดต้นแบบ | 60 |
| 4.31 ข้อมูลที่ได้จากการนำสับปะรดเน่ามาเทียบกับสับปะรดต้นแบบ | 60 |
| 4.32 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 1 | 66 |
| 4.33 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 2 | 67 |
| 4.34 ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 3 | 67 |
| 4.35 การนำสัญญาณเสียงที่ทราบลักษณะมาเปรียบเทียบกัน | 68 |
| 4.36 การนำสัญญาณเสียงมาปรียบเทียบกันโดยทำการสุ่มเลือกผลสับปะรด | 69 |

# สารบัญรูป


| รูปที่ | หน้า |
|---|---|
| 2.1 สัญญาณ $h[k]$ ที่ได้จากการทำการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง | 10 |
| 2.2 สัญญาณ $h[k]$ ที่ได้จากการทำฟูเรียร์ไม่ต่อเนื่อง | 11 |
| 2.3 สัญญาณ $h_1[k]$ ที่ได้จากการทำฟูเรียร์ไม่ต่อเนื่อง | 11 |
| 2.4 (a) ส่วนของสัญญาณสองสัญญาณที่จะนำมาทำการหาค่าของฟังก์ชันสหสัมพันธ์ข้าม (b) การประเมินค่าของฟังก์ชันสหสัมพันธ์ข้าม | 12 |
| 2.5 รูปของสัญญาณ $x(n)$ ซึ่งเป็นสัญญาณที่กำหนดขึ้น | 13 |
| 2.6 รูปการสร้างสัญญาณ $y(n)$ จากสัญญาณ $x(n)$ | 14 |
| 2.7 รูปของสัญญาณ $y(n)$ | 14 |
| 2.8 รูปผลการทำสหสัมพันธ์ข้ามของสัญญาณ $x(n)$ และ $y(n)$ | 14 |
| 3.1 ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ | 16 |
| 3.2 หน้าต่างของโปรแกรม Sound Recorder | 17 |
| 3.3 หน้าต่างของโปรแกรม MATLAB | 18 |
| 3.4 ภาพที่ได้จากอ่านข้อมูลแฟ้มที่ชื่อ sweet.wav เพื่อกำหนดขอบเขตของสัญญาณ | 18 |
| 3.5 ภาพหน้าต่างของโปรแกรม SPTOOL | 19 |
| 3.6 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนเวลา โดยใช้คำสั่ง SPTOOL | 20 |
| 3.7 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ โดยใช้คำสั่ง SPTOOL | 22 |
| 3.8 ภาพหน้าต่างของโปรแกรม GUIDE QUICK START | 23 |
| 3.9 ภาพหน้าต่างของโปรแกรม GUI ที่ได้มีการสร้างไว้เป็นที่เรียบร้อย | 24 |
| 3.10 ภาพหน้าต่างของโปรแกรม GUI ที่ได้ผ่านการปะมวลผล | 24 |
| 3.11 ภาพหน้าต่างของของสัญญาณเสียงที่นำมาทดสอบกับโปรแกรม | 25 |
| 3.12 ภาพหน้าต่างของการเปรียบเทียบค่าทดสอบสับปะรดตัวอย่างเทียบกับสับปะรดต้นแบบ | 26 |
| 3.13 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Js1 ซึ่งเคาะโดยผู้หญิง | 28 |
| 3.14 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Ns1 ซึ่งเคาะโดยผู้ชาย | 28 |
| 3.15 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jb1 ซึ่งเคาะบริเวณด้านข้างผล | 29 |
| 3.16 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jm1ซึ่งเคาะผลที่น้ำหนัก 1.7 k | 30 |
| 3.17 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jw1 ซึ่งเคาะโดยใช้ด้ามไขควง | 31 |
| 4.1 ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ | 33 |
| 4.2 แผนผังการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ | 36 |

# สารบัญรูป (ต่อ)


| รูปที่ | หน้า |
|---|---|
| 4.3 แผนผังการทำงานของโปรแกรมส่วน FFT | 36 |
| 4.4 แผนผังการทำงานของโปรแกรมส่วน Compare | 37 |
| 4.5 แผนผังการทำงานของโปรแกรมส่วน Compare2 | 38 |
| 4.6 แผนผังการทำงานของโปรแกรมส่วน Compare3 | 38 |
| 4.7 แผนผังการทำงานของโปรแกรมส่วน Compare4 | 39 |
| 4.8 แผงผังการทำงานของโปรแกรมส่วน Output | 39 |
| 4.9 ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลหวาน | 40 |
| 4.10 ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลเปรี้ยว | 41 |
| 4.11 ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลเน่า | 42 |
| 4.12 ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์โดยใช้สหสัมพันธ์ข้าม | 46 |
| 4.13 กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบของผลหวาน | 51 |
| 4.14 กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบของผลเปรี้ยว | 56 |
| 4.15 กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบของผลเน่า | 59 |
| 4.16 แผนผังแสดงเกณฑ์ที่ใช้ในการแบ่งแยกสับปะรดแบบต่างๆ | 63 |
| 4.17 ภาพของสัญญาณต้นแบบและสัญญาณเสียงที่นำมาทดสอบโดยการใช้โปรแกรม | 64 |
| 4.18 ภาพของสัญญาณต้นแบบและสัญญาณเสียงที่นำมาทดสอบโดยการใช้ Cross-correlation ทดสอบสับปะรดต้นแบบกับตัวอย่างสับปะรดชุดที่ 1 | 65 |
| ข.-1 ภาพหน้าแรกของการติดตั้ง | 87 |
| ข.-2 ภาพเลือกชนิดที่ต้องการติดตั้ง | 87 |
| ข.-3 ภาพใส่ชื่อ ชื่อองค์กร และ Personal License Password(PLP) | 88 |
| ข.-4 ภาพข้อตกลงในการติดตั้งโปรแกรม | 88 |
| ข.-5 ภาพประเภทของการติดตั้ง | 88 |
| ข.-6 ภาพไดเรคทอรี่และองค์ประกอบ | 89 |
| ข.-7 การเลือกตระกูลไฟล์และส่วนที่ต้องการให้แสดงการเข้าสู่โปรแกรม | 89 |
| ข.-8 ภาพยืนยันการติดตั้งโปรแกรม | 89 |

# สารบัญรูป (ต่อ)


| รูปที่ | หน้า |
|---|---|
| ข.-9 ภาพสถานะการติดตั้ง | 90 |
| ข.-10 ภาพ Product Configuration Notes | 90 |
| ข.-11 ภาพเสร็จสิ้นการติดตั้งโปรแกรม MATLAB | 90 |

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของโครงงาน

ในปัจจุบันการเลือกสับปะรดของพ่อค้าแม่ค้าทั่วไปนั้น จะอาศัยการเคาะแล้วฟังเสียง โดยอาศัยประสบการณ์เพียงอย่างเดียว ซึ่งไม่มีมาตรฐานการควบคุมที่แน่นอน และอาจจะเกิดการผิดพลาดขึ้นได้ ถ้ามีการใช้ระบบคอมพิวเตอร์ซึ่งเป็นเทคโนโลยีที่ทันสมัยเข้ามาใช้งาน จะทำให้การทำงานสะดวกและรวดเร็วขึ้น ดังนั้นโครงงานนี้ จึงได้มีการศึกษาการวิเคราะห์การคัดแยกสับปะรด โดยใช้คุณลักษณะของสัญญาณเสียงเป็นตัวบ่งชี้หรือบอกลักษณะคร่าว ๆ ของสับปะรด ซึ่งเป็นเครื่องมือที่ช่วยในการคัดแยกและการวิเคราะห์ คือ เครื่องคอมพิวเตอร์ และ การ์ดเสียง (Sound Card) ซึ่งจะมีคุณสมบัติในการบันทึกสัญญาณของเสียงในรูปของโดเมนเวลา (Time Domain) จากนั้นจะใช้โปรแกรม MATLAB เป็นตัวแปลงสัญญาณที่บันทึกได้จากการ์ดเสียง ให้อยู่ในรูปของโดเมนความถี่ (Frequency Domain) และสรุปความเป็นไปได้ในการคัดเลือกผลไม้โดยการวิเคราะห์จากเสียง เพื่อป้องกันการผิดพลาดจากการคัดแยกนั้น น่าจะทำให้สะดวกรวดเร็วในการอาศัยวิธีการแบบเดิมเพื่อเหมาะกับการส่งออกและโรงงานอุตสาหกรรม

## 1.2 วัตถุประสงค์ของโครงงาน

เพื่อศึกษาและพัฒนาการคัดเลือกสับปะรดโดยการวิเคราะห์เสียง

## 1.3 ขอบเขตของโครงงาน

1.3.1 ศึกษาเรื่องทั่วไปเกี่ยวกับสับปะรด และคุณสมบัติทางเสียงของสับปะรด

1.3.2 ศึกษาการบันทึกเสียงโดยใช้การ์ดเสียง (Sound Card) และวิเคราะห์สัญญาณเสียงโดยใช้โปร แกรม MATLAB

1.3.3 ศึกษาการใช้โปรแกรม MATLAB เพื่อวิเคราะห์สัญญาณเสียงในด้าน โดเมนเวลา (Time Domain) และด้าน โดเมนความถี่ (Frequency Domain)

1.3.4 ศึกษาความเป็นไปได้ในการคัดเลือกสับปะรด โดยใช้วิธีการวิเคราะห์สัญญาณเสียง

## 1.4 ขั้นตอนการดำเนินงาน

ทำการบันทึกเสียงจากการเคาะผลสับปะรด โดยใช้การ์ดเสียง (Sound Card) และบันทึกข้อมูลที่เกี่ยวข้องและเป็นประโยชน์ต่อการวิเคราะห์ข้อมูล เช่น ขนาด น้ำหนัก เป็นต้น จากนั้นนำเสียงที่บันทึกไว้มาทำการตรวจสอบจุดเริ่มต้นและจุดสิ้นสุดของข้อมูลเสียง แล้วนำสัญญาณที่ได้มาทำการวิเคราะห์สัญญาณเสียงในโดเมนเวลา (Time Domain) และโดเมนความถี่ (Frequency Domain)

**ตารางที่ 1.1** ตารางแสดงแผนการดำเนินงาน

| ลำดับที่ | กิจกรรม | เดือน | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | มิ.ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค |
| 1. | ศึกษาค้นคว้าข้อมูลเกี่ยวกับโครงงาน | ← | → | | | | | | |
| 2. | ศึกษาการบันทึกเสียงและทดสอบบันทึกเสียง | ← | → | | | | | | |
| 3. | ทดสอบการเคาะสับปะรดและจัดเก็บเสียง | ← | → | | | | | | |
| 4. | วิเคราะห์สัญญาณเสียงใน Time domain | | ← | → | | | | | |
| 5. | วิเคราะห์สัญญาณเสียงใน Frequency domain | | | ← | → | | | | |
| 6. | ศึกษาการใช้โปรแกรม MATLAB | | | | | ← | → | | |
| 7. | วิเคราะห์สัญญาณเสียงโดยใช้โปรแกรม MATLAB | | | | | | ← | → | |
| 8. | สรุปความเป็นไปได้ของการคัดเลือกสับปะรดด้วยเสียง | | | | | | ← | → | |

## 1.5 ผลที่คาดว่าจะได้รับ

1.6.1 ได้ศึกษาหลักการบันทึกเสียง โดยใช้การ์ดเสียง (Sound Card)

1.6.2 ได้ศึกษาหลักการวิเคราะห์สัญญาณเสียงในโดเมนเวลา (Time domain) และในโดเมนความถี่ (Frequency domain)

1.6.3 ได้ศึกษาการวิเคราะห์เสียงโดยใช้โปรแกรม MATLAB

1.6.4 ได้ศึกษาและพัฒนาการวิเคราะห์เสียงเพื่อคัดเลือกความเปรี้ยวและหวานของสับปะรดได้

## 1.6 งบประมาณ

| | |
|---|---|
| 1.7.1 ค่าวัสดุอิเล็กทรอนิกส์ | 1000 บาท |
| 1.7.2 ค่าวัสดุคอมพิวเตอร์ | 1000 บาท |
| 1.7.3 ค่าผลไม้ | 1000 บาท |
| งบประมาณที่ใช้ทั้งสิ้น | 3000 บาท |
| | (สามพันบาทถ้วน) |

มหาวิทยาลัยนเรศวร

# บทที่ 2
# บทความที่เป็นประโยชน์ต่อโครงงานและทฤษฎีที่เกี่ยวข้อง

จากการศึกษาค้นคว้าข้อมูลเพื่อที่จะต้องนำมาใช้ในการคัดแยกสับปะรดนั้น เราจำเป็นต้องทราบข้อมูลทั่วไปเกี่ยวกับสับปะรด รวมถึงบทความ และงานวิจัยต่าง ๆ ที่เป็นประโยชน์ต่อการทำโครงงาน และในการวิเคราะห์สัญญาณนั้นจำเป็นต้องเข้าใจทฤษฎีของ การแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) ซึ่งเป็นการแปลงสัญญาณให้อยู่ในรูปของโดเมนความถี่ (Frequency Domain) และทฤษฎีฟังก์ชันสหสัมพันธ์แบบข้าม (Cross Correlation Function) เพื่อใช้ในการเปรียบเทียบสัญญาณที่ต้องการทดสอบกับสัญญาณต้นแบบ

## 2.1 เรื่องทั่วไปของสับปะรด

สับปะรด ชื่อทางพฤกษศาสตร์คือ แอนนานาส คอสมอสูต (Ananas cosmosus)[1] สับปะรดเป็นพืชใบเลี้ยงเดี่ยวจำพวกไม้เนื้ออ่อน ที่มีอายุหลายปีสามารถทนต่อสภาพแวดล้อมต่าง ๆ ได้ดี ปลูกได้ในดินแทบทุกแห่งในประเทศไทย มีช่อดอกที่ส่วนยอดของลำต้น ซึ่งเมื่อเจริญเป็นผลแล้วจะเจริญต่อไป โดยตาที่ลำต้น จะเติบโตเป็นต้นใหม่ได้อีก สับปะรดแบ่งออกตามลักษณะความเป็นอยู่ได้สามประเภทใหญ่ ๆ คือ พวกที่มีระบบราก หาอาหารอยู่ในดิน หรือเรียกว่า ไม้ดิน พวกอาศัยอยู่ตามคาคบไม้หรือลำต้นไม้ใหญ่ ได้แก่ ไม้อากาศต่าง ๆ ที่ไม่แย่งอาหารจากต้นไม้ที่มันเกาะอาศัยอยู่ พวกนี้ส่วนใหญ่จะเป็นไม้ประดับ และพวกที่เจริญเติบโตบนผาหินหรือโขดหิน ส่วนสับปะรดที่เราใช้บริโภคจัดเป็นไม้ดิน แต่ยังมีลักษณะบางประการของไม้อากาศเอาไว้ คือ สามารถเก็บน้ำไว้ตามซอกใบได้เล็กน้อย มีเซลล์พิเศษสำหรับเก็บน้ำเอาไว้ในใบ ทำให้ทนทานในช่วงแล้งได้ ในประเทศไทยการปลูกสับปะรดสามารถทำได้เกือบตลอดทั้งปี ดังนั้นการเก็บผลสับปะรดก็สามารถทำได้เกือบตลอดทั้งปีเช่นกัน แต่ที่สับปะรดให้ผลชุกที่สุดมี 2 ช่วง คือ ช่วงสับปะรดปี ซึ่งจะเก็บผลได้มากกว่าสับปะรดทะวายประมาณ 3 เท่า ช่วงนี้จะอยู่ระหว่างเดือนเมษายนถึงมิถุนายน และช่วงสับปะรดทะวาย ซึ่งออกในเดือนตุลาคมถึงธันวาคม การสังเกตผลแก่ของสับปะรดพิจารณาได้จากลักษณะภายนอกผลดังนี้ ผิวเปลือก จะเปลี่ยนจากสีเขียวเป็นเขียวอมเหลืองอมส้มหรือเขียวเข้มเป็นมัน ใบเล็ก ๆ ของตาย่อย จะเหี่ยวแห้ง เป็นสีน้ำตาลหรือชมพู ตาย่อยจะนูนเด่นชัดเรียกว่าตาเต็ม ร่องตาจะตื้นเต็มที่ขนาดของผลไม่เพิ่มขึ้นอีกคมกลิ่น ผลสับปะรดแก่จะส่งกลิ่นหอมเฉพาะตัว ความแน่นของผล จะลดลงเมื่อใช้นิ้วดีดหรือไม้เคาะเพื่อฟังเสียง ถ้าเสียงโปร่งแสดงว่ายังไม่แก่ แต่ถ้าเสียงทึบ แสดงว่าแก่จัดได้ที่แล้ว

**2.1.1 การเก็บผลเพื่อบริโภคผลสด**

ใช้มีดตัดที่ก้านผลให้เหลือขั้วติดผลไว้บ้าง และคงให้มีจุกติดอยู่กับผลเพื่อป้องกัน การเน่าของผล อันเนื่องจากแผลที่เกิดจาก การปลิดจุกหรือขั้วผลออก หลังจากตัดผลแล้วให้ใช้มีดฟันใบต้นเดิมออกเสียบ้าง เพื่อให้หน่อได้รับแสงแดดอย่างเต็มที่ และเหลือหน่อดินไว้แทนต้นเดิมตั้งแต่ 1 ถึง 2 หน่อเท่านั้น ส่วนหน่อที่เหลือก็ขุดหรือปลิดออกจากต้นนำไปปลูก ขยายเนื้อที่หรือจำหน่าย ต่อไปได้ พันธุ์ภูเก็ต จะนิยมปลิดจุกตั้งแต่ผลมีอายุประมาณ 2 เดือน ส่วนพันธุ์อินทรชิตและพันธุ์ขาวจะตัดจุกทิ้งประมาณหนึ่งในสองส่วน ในเวลาที่เก็บผลจำหน่าย

**2.1.2 การขยายพันธุ์**

ส่วนต่าง ๆ ที่ใช้ในการขยายพันธุ์สับปะรด มีดังนี้

- หน่อดิน เกิดจากตาที่อยู่ในบริเวณลำต้นใต้ดิน ซึ่งจะเริ่มแทงขึ้นมาพ้นผิวดิน หลังจากเกิดสร้างดอกแล้ว มีจำนวนน้อย รูปทรงเล็กเรียว ใบยาวกว่าหน่อข้าง
- หน่อข้าง เกิดจากตาที่พักตัวอยู่บนลำต้นในบริเวณโคนใบหน่อข้างเหล่านี้ จะมีน้ำหนักต่าง กันไปตั้งแต่ 0.5 ถึง 1 กิโลกรัม ให้ผลเมื่อมีอายุ 14 ถึง 18 เดือน ใช้ขยายพันธุ์ได้ดี
- ตะเกียง เกิดจากตาบนก้านผลที่อยู่ในบริเวณโคนผล ตะเกียงมีน้ำหนักเฉลี่ยทั่วไปอยู่ระหว่าง 0.3 ถึง 0.5 กิโลกรัม ให้ผลเมื่อมีอายุ 18 ถึง 20 เดือน
- จุก เติบโตขึ้นเหนือผลสับปะรดหลังจากดอกโรยไปแล้วจุกจะมีน้ำหนักทั่วไป ตั้งแต่ 0.075 ถึง 0.2 กิโลกรัม ให้ผลตามธรรมชาติเมื่ออายุ 22 ถึง 24 เดือน เมื่อเก็บผลสับปะรดก็จะปลิดจุกออกจากผล และหลังจากเก็บเกี่ยวผลไปแล้วประมาณ 6 สัปดาห์ ก็จะปลิดหน่อออกจากต้น หน่อที่มีขนาดเหมาะแก่การขยายพันธุ์คือ มีความยาวประมาณ 50 ถึง 75 เซนติเมตร หลังจาก เก็บหน่อตะเกียง หรือจุกมาแล้ว ให้นำมาผึ่งแดดโดยคว่ำยอดลงสู่พื้นดิน ให้โคนแผลได้รับแสงแดดจนรอยแผลแห้งรัดตัวเป็นการฆ่าเชื้อโรคด้วย แล้วนำมามัดรวมกันเป็นกองเพื่อรอการปลูกหรือนำไปขายต่อไป ก่อนปลูกต้องลอกกาบใบล่างออก 3 ถึง 4 ชั้น เพื่อให้รากแทงออกมาได้สะดวกและเร็วขึ้น

**2.1.3 ประโยชน์ของสับปะรด**

สับปะรดมีส่วนต่าง ๆ ที่ใช้ประโยชน์ได้กว้างขวาง ดังนี้

- **เนื้อ** ใช้รับประทานสดหรือแปรรูปเป็นสับปะรดแช่อิ่ม สับปะรดกวน สับปะรดแห้ง แยมสับปะรด หรือบรรจุกระป๋อง และคั้นทำน้ำสับปะรด ในภาคตะวันออกเฉียงเหนือใช้เนื้อสับปะรดผสมกับปลา และเกลือหมักไว้ทำเป็นอาหารที่เรียกว่า " เค็มหมากนัด "
- **ผลพลอยได้จากเศษเหลือ** เศษเหลือของสับปะรดส่วนใหญ่นั้นได้จากอุตสาหกรรมบรรจุกระป๋อง สามารถนำมาแปรรูปทำอย่างอื่นได้ เช่น น้ำเชื่อม แอลกอฮอล์ น้ำส้มสายชู ไวน์ อาหารสำหรับเลี้ยงวัว เป็นต้น

- **ใบ** เส้นใยจากใบสับปะรด นำมาทอเป็นผ้าใยสับปะรด ในฟิลิปปินส์เรียกว่า " ผ้าบารอง " มีราคาแพง ใช้นิยมตัดเป็นชุดสากลประจำของชาติฟิลิปปินส์และได้หวัน อีกตัวอย่างคือ เยื่อกระดาษจากใยใบสับปะรด จะได้กระดาษที่มีคุณสมบัติพิเศษ คือ ความบางมาก มีผิวนุ่มเนียน สามารถบิดงอ หรือเปลี่ยนรูปร่างได้ง่าย โดยไม่เสียหาย ในหลายประเทศนิยมใช้เป็นกระดาษสำหรับพิมพ์ธนบัตร

- **เปลือก** เศษเหลือทิ้งส่วนใหญ่ได้มาจากโรงงานสับปะรดนั้น นั่นก็คือ เปลือกและแกนกลางซึ่งจะมีน้ำอยู่สูงถึงร้อยละ 90 เมื่อคิดต่อน้ำหนักสดส่วนเหลือทิ้งจะมีโปรตีน และโภชนะย่อยได้ทั้งหมดประมาณร้อยละ 0.7 และ 7 เมื่อคิดต่อน้ำหนักแห้ง จะมีค่าโปรตีน และโภชนะย่อยได้สูงถึงร้อยละ 7 และ 70 ตามลำดับ ปกติวัวชอบกินเปลือกสับปะรด ยิ่งเปลือกที่ทิ้งไว้ 2 ถึง 3 วัน สีออกเป็นน้ำตาลเทา ๆ มีกลิ่นเหม็นเล็กน้อย วัวจะชอบกินมากกว่าเปลือกสด ดังนั้นหากเลี้ยงวัวในแหล่งที่มีโรงงานสับปะรด จึงใช้เปลือกสับปะรดเป็นอาหารเลี้ยงวัวได้ โดยนำเปลือกมากองทิ้งไว้อย่างน้อย 24 ชั่วโมง จึงใช้เป็นอาหารเลี้ยงวัวได้เป็นการลงทุนที่น้อยที่สุด แต่ให้ผลตอบแทนสูง

## 2.2 บทความที่เป็นประโยชน์ต่อโครงงาน

**2.2.1 ความถี่ธรรมชาติ (Natural Frequency)** [2] ความถี่ของการสั่นของวัตถุ ซึ่งเกิดขึ้นจากการรบกวนวัตถุนั้น หรือการให้พลังงานกับวัตถุนั้น ซึ่งพลังงานที่ให้แก่วัตถุจะทำให้วัตถุเกิดการสั่นเราเรียกว่าความถี่ธรรมชาติ โดยขึ้นอยู่กับขนาดรูปร่างของวัตถุนั้นๆ ความถี่ธรรมชาติในผลไม้ได้มีการศึกษาอย่างเป็นระบบซึ่งเป็นการนำเทคนิคการวัดความถี่ธรรมชาติของ วัตถุทางอุตสาหกรรมมาประยุกต์ใช้ เทคนิคดังกล่าวมี 2 วิธีหลัก ๆ คือ

- **วิธี Forced vibration** เป็นวิธีที่กระตุ้นให้วัตถุสั่นด้วยวิธีทางกล เช่น การใช้เครื่องเขย่า (Vibrator) เป็นต้น วัตถุจะถูกวางลงบนเครื่องเขย่า และถูกทำให้สั่นในช่วงความถี่ครอบคลุมความถี่ธรรมชาติของวัตถุขณะที่สั่น จะใช้เครื่องวัดความเร่ง (Accelerometer) มาวัดขนาดการสั่นที่ถ่ายทอด ผ่านวัตถุ ความถี่ที่ให้ขนาดการสั่นได้ดีที่สุด คือ ความถี่ธรรมชาติ

- **วิธี Free vibration** วิธีนี้จะกระตุ้นให้วัตถุสั่นที่ความถี่ธรรมชาติโดยตรง โดยอาจใช้วิธีการเคาะแล้ววัดขนาดการสั่นของเสียงเคาะด้วยเครื่องขยายเสียง ความถี่ธรรมชาติของวัตถุคือความถี่เสียง

**2.2.2 Investigation of change exponential frequency index as related to durian maturity**

[3] หลักการวัดความถี่ธรรมชาติ แบบโดยตรง (Free Vibration) ถูกนำมาศึกษา และประยุกต์ใช้กับทุเรียนเพื่อนำมาหาดัชนีความถี่ธรรมชาติของทุเรียน โดยมุ่งเน้นการนำดัชนีความถี่ธรรมชาติซึ่งคำนวนจากความถี่ธรรมชาติ ($f$) และน้ำหนัก ($m$) ของผลไม้ มาวัดความแก่ (Maturity) หรืออายุของทุเรียนขณะเจริญเติบโตบนต้น การวิจัยใช้ทุเรียนพันธุ์หมอนทอง 6 ต้นต้นละ 5 ผล เป็นตัวอย่าง เพื่อวัดหาความถี่ธรรมชาติโดยเคาะทุเรียนแต่ละผล และวัดเสียง รวมทั้งวัดปริมาตรเพื่อหาน้ำหนักทุก 1 อาทิตย์ เริ่มตั้งแต่ทุเรียนมีอายุหลังดอกบาน ประมาณ 55 วัน จนถึง 132 วัน ซึ่งเป็นเวลาที่ ทุเรียนแก่เต็มที่ นำค่า $f$ และ $m$ มาศึกษาเพื่อหาดัชนีความถี่ธรรมชาติที่เหมาะสม และนำค่าดัชนีนี้มาหาความสัมพันธ์กับดัชนีที่นิยมใช้กำหนดการเก็บทุเรียนผลที่แก่เต็มที่ซึ่งคืออายุหลังดอกบาน เมื่อวิเคราะห์ข้อมูลเพื่อหาดัชนีความถี่ธรรมชาติที่เหมาะสมพบว่า ค่า $f2$ ของกลุ่มทุเรียนที่มีอายุหลังดอกบานเท่ากัน จะแปรผกผันกับ $m^{2/3}$ ในลักษณะเลขชี้กำลัง (Exponential) นั่นคือ ผลใหญ่จะมี $f$ น้อยกว่า หลังจากศึกษาความสัมพันธ์นี้ในกลุ่มทุเรียนอื่น ๆ ที่อายุบนต้นต่าง ๆ กัน ได้ข้อสรุปคือ ความสัมพันธ์แบบเลขชี้กำลัง (Exponential) $f2$ และ $m^{2/3}$ ระหว่าง ปรากฏอยู่ในทุกกลุ่ม ทุเรียน แต่อัตราการลดลงแบบเลขชี้กำลัง (Exponential)จะสูงขึ้นในกลุ่มทุเรียนที่อายุมากกว่า ดังนั้น ค่าดัชนีความถี่ธรรมชาติที่เหมาะสมและเป็นตัวแทนของอัตราการลดลงดังกล่าวคือ $(2/3)f2\ln(m)$ ซึ่งเมื่อนำมาเขียนกราฟกับอายุหลังดอกบาน จะพบว่า เมื่ออายุหลังดอกบานมากขึ้น หรือเมื่อทุเรียนแก่มากขึ้น ดัชนีความถี่ธรรมชาติ $(2/3)f2\ln(m)$ ของทุเรียนมีแนวโน้มลดลงอย่างมีนัยสำคัญ และจะมีค่าเข้าหาค่า ๆ หนึ่ง เมื่อทุเรียนแก่เต็มที่

**2.2.3 การศึกษาการใช้คุณสมบัติของเสียงเพื่อตรวจสอบและ คัดแยกอาหารกระป๋อง** [4]

น่ายรณฤทธิ์ ฤทธิรณ และนายอนุพันธ์ เทอดวงศ์วรกุล ใช้การตรวจสอบน้ำผลไม้กระป๋องที่บุบ ด้วยเทคนิคการทดสอบแบบไม่ทำลาย (Nondestructive Technique) และนำสัญญาณมาแปรรูปด้วย ฟังก์ชันการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) เพื่อให้ได้ค่าความถี่ เมื่อนำค่าความถี่ที่มีแอมพลิจูด (Amplitude) ของ กำลังสเปกตรัม (Power Spectrum) สูงสามค่าแรกมาทำการเรียงลำดับค่าความถี่ใหม่จากน้อยไปมาก พบว่าเมื่อนำช่วงค่าความถี่อันดับที่ 2 ที่ระดับนัยสำคัญ 0.8 ของน้ำผลไม้กระป๋องที่มีคุณภาพดีมาใช้ในการแยกน้ำผลไม้กระป๋องที่บกพร่องได้ 62 % คนส่วนมากมักจะบอกความสุกและคุณภาพอื่น ๆ ของผลไม้ เช่น แอปเปิล แตงโม สับปะรด ด้วยการฟังเสียงที่เราเคาะผลไม้นั้น จากวิธีการดังกล่าวจึงมีนักวิจัยหลายท่านได้พยายามพิสูจน์ ข้อกล่าวอ้างข้างต้น โดยศึกษาการตอบสนองของเสียงที่ได้ฟังจากผลไม้ต่อการกระตุ้นวิธีการนี้เรียกว่า การตอบสนองทางเสียง (Acoustic Response) ในปี ค.ศ.1980 Yamamoto et. al. ได้พัฒนาเทคนิคการวิเคราะห์แบบไม่ทำลาย (Nondestructive Technique) สำหรับการวัดคุณภาพเนื้อสัมผัสของแอปเปิ้ล และแตงโม โดยอาศัยการตอบสนองทางเสียง (Acoustic Response) ของผลไม้ ซึ่งวิธีนี้อาศัยอุปกรณ์ที่ไม่ยุ่งยาก กล่าวคือ เทคนิคการสั่นทางกล (Forced Vibration) ต้องใช้ เครื่องวัดความเร่ง (Accelerometer) ติดกับผิวของผลไม้ เพื่อรับสัญญาณการสั่นของผลไม้ แต่ถ้าวิธีการตอบสนองทางเสียง (Acoustic Response) นั้น จะใช้เครื่องขยายเสียง (Microphone) แทนเครื่องวัดความเร่ง (Accelerometer) ซึ่งไม่ต้องสัมผัสกับผลไม้ และใช้ลูกตุ้มที่ทำด้วยไม้ (Wooden-ball Pendulum) เป็นแหล่งกำเนิดสัญญาณเสียงแทนการสั่นทางกล พวกเขาพบว่าจะได้ ความถี่การสั่น (Resonance Frequency) ของผลไม้จากเสียงที่ถูกบันทึกครั้งแรกจากการตี หรือเคาะผลไม้ด้วยลูกตุ้มที่ทำด้วยไม้จากนั้นใช้ การแปลงฟูเรียร์ (Fourier Transformation) กับสัญญาณเสียงนั้นเพื่อหากำลังสเปกตรัม (Power Spectrum) ของคลื่นเสียง พวกเขาพบว่าความถี่การสั่น (Resonance Frequency) ของแอปเปิ้ลและแตงโมจะลดลงตามระยะเวลาที่เก็บ (Storage Time) และดัชนีความแน่น เนื้อ (Firmness Index) สามารถแสดงเป็นฟังก์ชันของความถี่การสั่น น้ำหนัก และความหนาแน่น ของผลไม้ การหาปัจจัยที่มีผลกระทบค่าความถี่การสั่น (Resonance Frequency) ได้ว่าวิธีการสร้างแหล่ง กำเนิดเสียง ตำแหน่งที่ตี หรือเคาะ และวิธีการถือหรือจับผลไม้ไม่มีผลต่อค่าความถี่การสั่น (Resonance Frequency) แต่มีผลกระทบต่อขนาดของการสั่นสะเทือนที่ความถี่การสั่น (Resonance Frequency) อันดันสองขึ้นไป และตำแหน่งของเครื่องขยายเสียง (Microphone) ที่ตรงข้ามกับตำแหน่งเคาะจะให้ค่าของการสั่นสะเทือนชัดเจนที่สุด

**2.2.4 การแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform ) [5]**

การแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) หรือเรียกสั้น ๆ ว่า DFT เป็นการแปลงสัญญาณที่อยู่ในรูปของ โดเมนเวลา (Time Domain) ให้อยู่ในรูปของ โดเมนความถี่ (Frequency Domain) ซึ่งการแปลงฟูเรียร์ไม่ต่อเนื่องนี้ จะต่างจากการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง (Discrete-time Fourier Transform) ซึ่งหาได้จากลำดับ (Sequence) ที่ความยาวจำกัด (Finite Length) หรือความยาวไม่จำกัด (Infinite Length) ก็ได้ ส่วนการแปลงฟูเรียร์ไม่ต่อเนื่อง หาได้จากลำดับที่จำกัดเท่านั้น และการแปลงฟูเรียร์ไม่ต่อเนื่องพัฒนามาจากการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง

พิจารณาลำดับ (Sequence) ที่มีจำนวน $N$ ตัว ซึ่ง

$$f[k] \quad \text{สำหรับ} \quad k = 0, 1, 2, \ldots, N-1$$

ในการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง (Discrete-time Fourier Transform) นั้น จะได้ความถี่แบบ ต่อเนื่อง (Continuous) ซึ่งมีสมการคือ

$$F(\varpi) = \sum_{k=-\infty}^{\infty} f[k] e^{-j\varpi k} = \sum_{k=0}^{N-1} f[k] e^{-j\varpi k} \tag{2.1}$$

มีคาบเป็น $2\pi$ มีความถี่อยู่ในช่วง $(-\pi, \pi]$ ถ้า $f[k]$ มี $N$ จุดที่เราสามารถหาค่า $\varpi$ ได้ทั้งหมด $N$ ค่าในช่วง $(-\pi, \pi]$ ซึ่งทั้งหมด $N$ ค่าหาได้จาก

$$\varpi_m = m\frac{2\pi}{N} \tag{2.2}$$

โดยที่ $m = 0, 1, 2, \ldots, N-1$ ทำให้ได้ค่าความถี่หรือ $\varpi$ มีขอบเขต (Range) อยู่ในช่วง $[0, 2\pi)$ และสามารถหา $F[m] := F(\varpi m) = F\left(\frac{2\pi m}{N}\right)$, สำหรับ $m = 0, 1, 2, \ldots, N-1$ จะได้

$$F[m] = F\left(m\frac{2\pi}{N}\right) = \sum_{k=0}^{N-1} f[k] e^{-jmk2\pi/N} \tag{2.3}$$

ในสมการที่ (2.3) ซึ่งเป็นสมการของการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) ของ $f[k]$ ซึ่งสมการที่ได้นี้ จะเป็นฟังก์ชันของความถี่ที่เป็นแบบ ไม่ต่อเนื่อง (Discrete) และในทางกลับกันเราสามารถหาส่วนกลับ (Inverse) ของการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) โดย

$$f[k]=\frac{1}{2\pi}\int_0^{2\pi}F(\varpi)e^{jk\varpi}d\varpi\approx\frac{1}{2\pi}\sum_{m=0}^{N-1}F(\varpi_m)e^{jkm2\pi/N}\cdot\left(\frac{2\pi}{N}\right) \tag{2.4}$$

หรือ

$$f[k]\approx\frac{1}{N}\sum_{M=0}^{N-1}F[m]e^{jkm2\pi/N} \tag{2.5}$$

สำหรับ $k=0,1,2,\ldots,N-1$ และจะให้ $f[k]=0$ สำหรับ $k<0$ และ $k>N-1$ และให้ $W=e^{-j2\pi/N}$ เพราะฉะนั้น ได้สมการของการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) ของ $f[k]$ เป็น

$$F[m]=\sum_{k=0}^{N-1}f[k]e^{-j2\pi k/N}=\sum_{k=0}^{N-1}f[k]W^{km} \tag{2.6}$$

สำหรับ $m=0,1,2,\ldots,N-1$

และ ส่วนกลับ (Inverse) สมการของการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) ของ $f[m]$ คือ

$$f[k]=\frac{1}{N}\sum_{m=0}^{N-1}F[m]e^{j2\pi mk/N}=\frac{1}{N}\sum_{m=0}^{N-1}F[m]W^{-mk} \tag{2.7}$$

สำหรับ $k=0,1,2,\ldots,N-1$

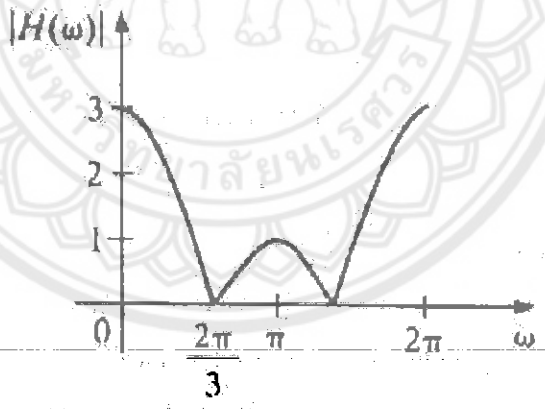
การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) คือรูปแบบการของการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) ชนิดหนึ่ง โดยใช้ในการคำนวณบนเครื่องคอมพิวเตอร์ เพื่อให้การคำนวณมีความรวดเร็วขึ้น โดยใช้คุณสมบัติที่ว่า ในการคำนวณการแปลงฟูเรียร์ไม่ต่อเนื่อง เมื่อให้จำนวนลำดับ (Sequence) $N = 2n$ ค่าที่ใช้ในการคำนวณจำนวนมากจะซ้ำซ้อนกัน การเขียนขั้นตอนการคำนวณที่อาศัยคุณสมบัติค่าซ้ำซ้อนนั้น จะช่วยให้คอมพิวเตอร์ทำงานได้ง่ายและเร็วขึ้นมาก คือถ้าใช้คำนวณแบบการแปลงฟูเรียร์ไม่ต่อเนื่อง โดยตรงจะต้องใช้ถึง $N^2$ ครั้ง แต่ถ้าใช้คำนวณแบบ การแปลงฟูเรียร์แบบเร็ว จะใช้เพียงแค่ $0.5N\log 2N$ ครั้ง เท่านั้น

ตัวอย่างลำดับ (Sequence) ที่นำมาแปลงให้อยู่ในโดเมนความถี่ (Frequency Domain)

$$h[k] = \begin{cases} 1 & for \quad k = 0,1,2 \\ 0 & otherwise \end{cases}$$

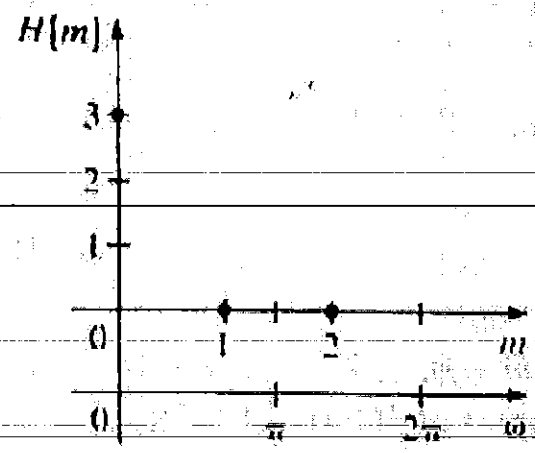
ซึ่งเป็นลำดับ (Sequence) ที่มีความยาว (Length) เป็น 3 ถ้าทำการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง

(Discrete-time Fourier Transform) จะได้ดังรูปดังนี้




**รูปที่ 2.1** สัญญาณ $h[k]$ ที่ได้จากการทำการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง

ถ้าทำการแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) จะได้รูปดังนี้



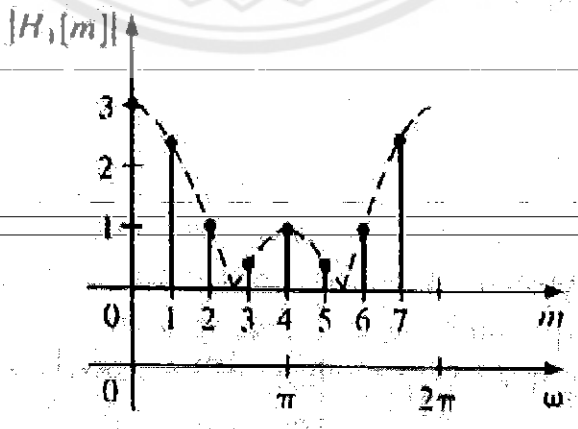

**รูปที่ 2.2** สัญญาณ $h[k]$ ที่ได้จากการทำฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform)

จะมีจำนวนค่าเท่ากับความยาว (Length) ของลำดับ (Sequence) คือ 3 ค่า ซึ่งเพียงพอที่จะแปลงกลับแล้วได้ลำดับ (Sequence) เดิม

แต่ถ้าลำดับ (Sequence) ที่นำมาพิจารณาเป็นดังนี้

$$h_1[k] = \begin{cases} h[k] & for \quad k = 0,1,2 \\ 0 & for \quad k = 3,4,5,6,7 \end{cases}$$

ซึ่งเป็นลำดับ (Sequence) ที่มีความยาว (Length) เป็น 8 ถ้าทำการแปลงฟูเรียร์เวลาไม่ต่อเนื่อง (Discrete-time Fourier Transform) จะได้เหมือนกับ $h[k]$ ดังรูปที่ 2.1 แต่ถ้าทำการแปลงแปลงฟูเรียร์ไม่ต่อเนื่อง (Discrete Fourier Transform) จะได้รูปดังนี้




**รูปที่ 2.3** สัญญาณ $h_1[k]$ ที่ได้จากการทำฟูเรียร์ไม่ต่อเนื่อง ( Discrete Fourier Transform )

**2.2.5 ฟังก์ชันสหสัมพันธ์ข้าม (Cross-correlation Function)**

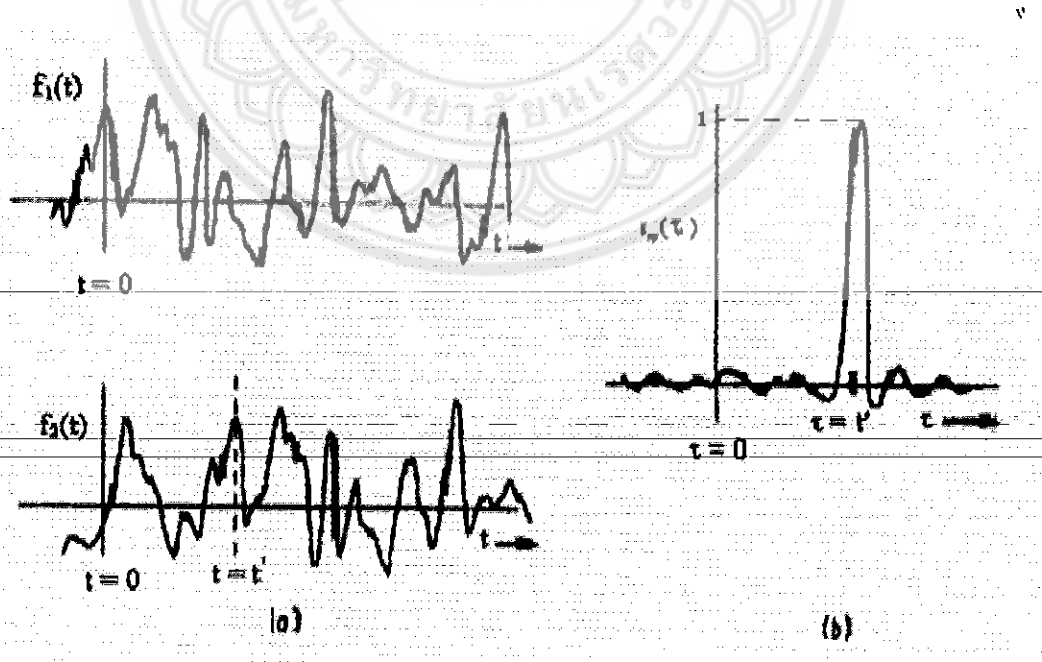
ฟังก์ชันสหสัมพันธ์ข้าม (Cross-correlation Function) เป็นการแสดงความสัมพันธ์ระหว่างสัญญาณ 2 สัญญาณ ซึ่งหมายถึง สัญญาณ $f_1(t)$ และ $f_2(t)$ โดยสามารถหาได้จากสมการ

$$r_{xy}(\tau)=\lim_{T_0\to\infty}\frac{1}{T_0}\int_{-T_0/2}^{T_0/2} f_1(t)\cdot f_2(t+\tau)\cdot dt \qquad (2.8)$$

ซึ่ง $\tau$ คือ ค่าการเลื่อนเวลาของสัญญาณอันใดอันหนึ่ง

สิ่งที่สำคัญอีกอย่างคือ ฟังก์ชันสหสัมพันธ์ข้าม (Cross-correlation Function หรือ CCF) และฟังก์ชันสหสัมพันธ์ในตัวเอง (Auto-correlation Function หรือ ACF) จะคล้ายกันมาก แต่จะแตกต่างเพียงฟังก์ชันสหสัมพันธ์ข้ามจะใช้กับ 2 สัญญาณที่แตกต่างหรือเป็นคนละตัวกัน ซึ่งเป็นการเปรียบเทียบ

สัญญาณ ส่วนฟังก์ชันสหสัมพันธ์ในตัวเอง จะแสดงความสัมพันธ์ของสัญญาณเดียวกัน ซึ่งตัวหนึ่งเป็นตัวที่ไม่ถูกเลื่อนเวลา แต่อีกตัวเป็นตัวที่ถูกเลื่อนเวลา ซึ่งค่าฟังก์ชันสหสัมพันธ์ข้าม จะเป็นฟังก์ชั่นต่อเนื่อง (Continuous Function) *xyr* แสดงตัวอย่างดังรูปที่ 2.4




**รูปที่ 2.4** (a) ส่วนของสัญญาณสองสัญญาณที่จะนำมาทำการหาค่าของฟังก์ชันสหสัมพันธ์ข้าม (b) การประเมินค่าของฟังก์ชันสหสัมพันธ์ข้าม (Cross correlation Function)

สัญญาณ $f_1(t)$ และ $f_2(t)$ เมื่อ $f_2(t)$ ถูกเลื่อนเวลาไป $t = t'$ จะมีลักษณะของสัญญาณเหมือนกับสัญญาณ $f_1(t)$ ที่จุด หรือ $f_2(t) = f_1(t - t')$ ซึ่งค่าเฉลี่ยของการคูณของ $[f_1(t) \cdot f_2(t)]$ ในช่วงอื่น ๆ จะเข้าใกล้ศูนย์ จนมาถึง เมื่อเลื่อน $\tau$ เท่ากับ $t$ จะได้ค่าฟังก์ชันสหสัมพันธ์ข้าม หรือ ค่าที่มีค่าสูงสุด $r_{xy}$ ส่วนฟังก์ชันสหสัมพันธ์ข้ามที่เป็นแบบดิจิทัล (Digital) ยังคงมีการเลื่อนเวลาการคูณของสัญญาณที่จะหาความสัมพันธ์และการหาปริพันธ์ (Integrate) แต่ในทางดิจิทัลจะทำในรูปแบบผลรวม (Summation) ซึ่งเป็นไปตามสมการ

$$r_{xy}(\tau) = r_{xy}(mT) \approx \frac{1}{N} \sum_{n=1}^{N} x(n) \cdot y(n+m) \qquad (2.9)$$

โดยที่ $N$ คือ จำนวนการสุ่มตัวอย่าง (Sampling) ของสัญญาณ $x(n)$ และ $y(n)$ ซึ่งต้องมีจำนวน

การสุ่มตัวอย่างเท่ากัน

$m$ คือ จำนวนครั้งของการเลื่อนเวลา เพื่อหาค่า $r_{xy}$

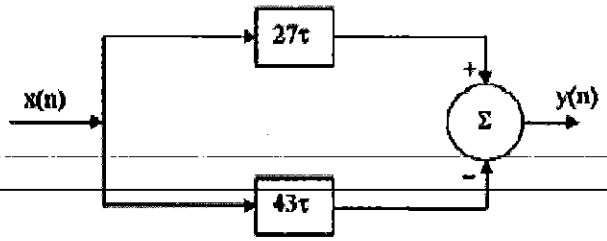
$T$ คือ คาบของการเลื่อนเวลาในแต่ละครั้ง

ตัวอย่างดังรูปที่ 2.5

**รูปที่ 2.5** รูปของสัญญาณ $x(n)$ ซึ่งเป็นสัญญาณที่กำหนดขึ้น แล้วนำสัญญาณนี้มาสร้างสัญญาณ

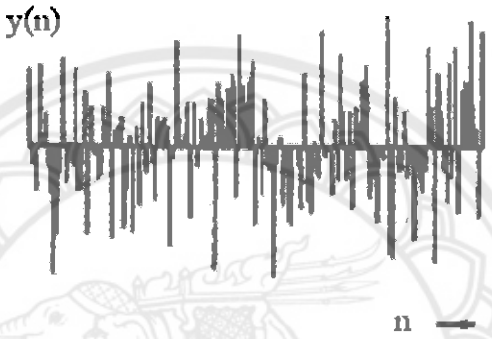
ซึ่งสัมพันธ์ระหว่าง $x(n)$ และ $y(n)$ แสดงได้ดังสมการนี้

$$y(n) = x(n - 27) - x(n - 43) \qquad (2.10)$$



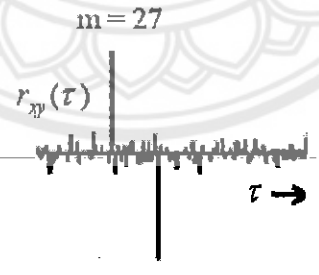

**รูปที่ 2.6** รูปการสร้างสัญญาณ $y(n)$ จากสัญญาณ $x(n)$

จะได้สัญญาณ $y(n)$ ดังรูปที่ 2.7




**รูปที่ 2.7** รูปของสัญญาณ $y(n)$

จากนั้นนำสัญญาณ $x(n)$ และ $y(n)$ มาทำการสหสัมพันธ์ข้าม (Cross correlation) กัน จะได้ผลดังรูปที่ 2.8




**รูปที่ 2.8** รูปผลการทำสหสัมพันธ์ข้าม (Cross correlation) ของสัญญาณ $x(n)$ และ $y(n)$

ซึ่งหาได้จากสมการนี้

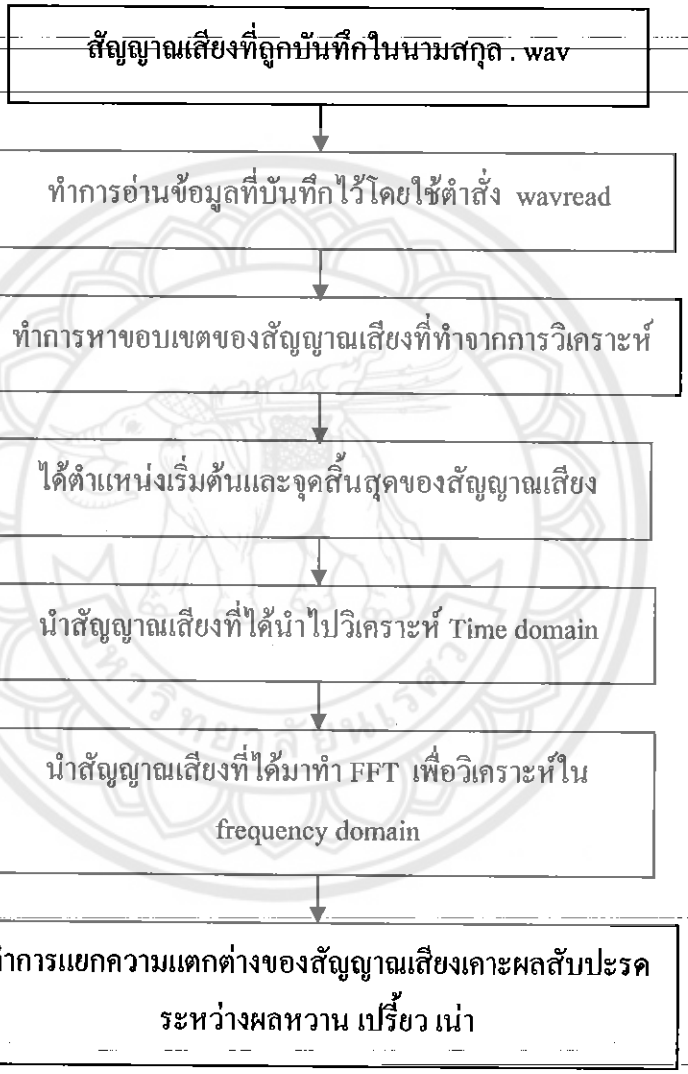
$$r_{xy}(\tau) = r_{xy}(mT) \approx \frac{1}{N}\sum_{n=1}^{N} x(n) \cdot y(n+m) \quad , with\ N = 500 \qquad (2.11)$$

โดยมีค่าที่ถูกเลื่อนไปแตกต่างกัน 100 ค่า คือ ($\tau = mT, m = 0, 1, 2, \ldots, 99$)

จากรูปที่ 2.8 เราจะสังเกตเห็นว่า การคำนวณฟังก์ชันสหสัมพันธ์ข้าม (Cross-correlation Function) จะได้ค่าการสหสัมพันธ์ข้ามที่มีค่าน้อย ยกเว้นที่ $m = 27$ และที่ $m = 43$ แสดงว่า $y(n)$ เกิดจากการทับซ้อน (Superposition) ของ ค่า $x(n)$ ทั้ง 2 แบบ คือ ตัวแรกเป็นตัวที่ถูกลดไป 27 ตำแหน่ง และตัวที่สองเป็นตัวที่ถูกลดไป 43 ตำแหน่ง ซึ่งตรงข้อสรุปของสัญญาณ $y(n)$ ที่ได้สร้างขึ้น

# บทที่ 3
# การออกแบบและการวิเคราะห์

## 3.1 หลักการทำงานของโปรแกรม MATLAB เพื่อใช้ในการวิเคราะห์เสียง

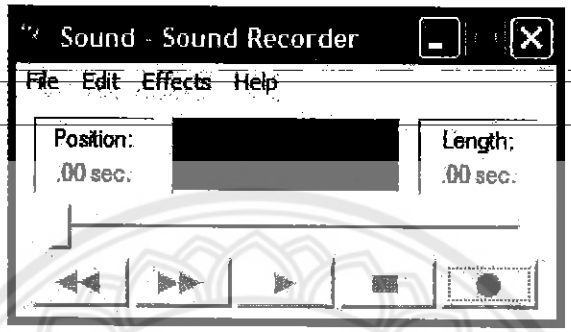



**รูปที่ 3.1** ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์

## 3.2 การเก็บข้อมูลสัญญาณเสียงเคาะผลสับปะรด

มีขั้นตอนการทำงานดังนี้

3.2.1 ทำการบันทึกเสียงโดยการใช้โปรแกรม Sound Recorder ของไมโครซอฟท์วินโดว์ โดยเริ่มที่ Start >> Programs >> Accessories >> Entertainment >> Sound Recorder จะได้หน้าต่างของโปรแกรม Sound Recorder ดังภาพข้างล่างนี้




รูปที่ 3.2 หน้าต่างของโปรแกรม Sound Recorder

3.2.2 ทำการบันทึกเสียงโดยเลือกที่ปุ่มบันทึก (ปุ่มสีแดง)

3.2.3 ทำการจัดเก็บเสียงที่บันทึกแล้ว โดยเลือก File >> Save As จากนั้นทำการตั้งชื่อแฟ้ม (File) แล้วเลือก OK เป็นอันเสร็จสิ้นการบันทึกเสียง

## 3.3 การอ่านข้อมูลที่บันทึกไว้ และหาขอบเขตของสัญญาณเสียง

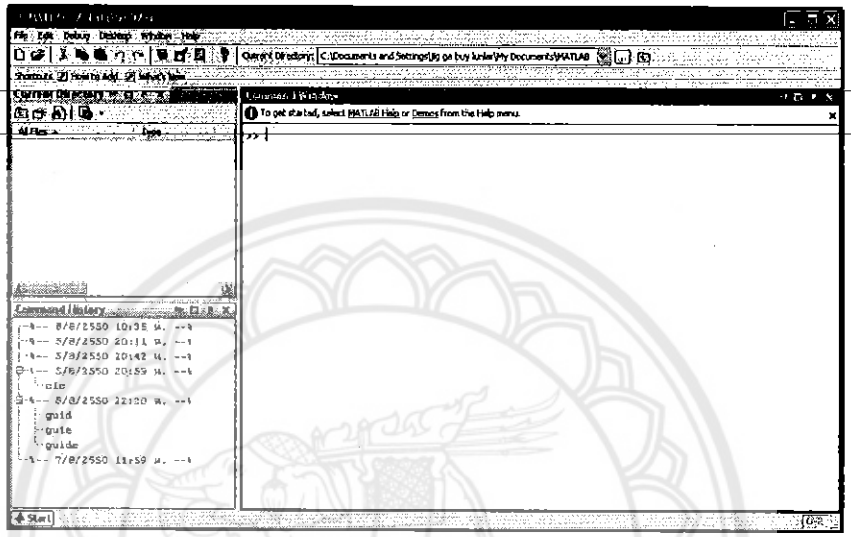
มีขั้นตอนการทำงานดังนี้

3.3.1 ทำการเปิดโปรแกรม MATLAB ขึ้นมา

3.3.2 ทำการเลือก Current Directory ที่เก็บข้อมูลเสียงนามสกุล .wav ที่บันทึกไว้

3.3.3 พิมพ์คำสั่ง wavread แล้วตามด้วยชื่อแฟ้ม (File) ที่ต้องการอ่านข้อมูล

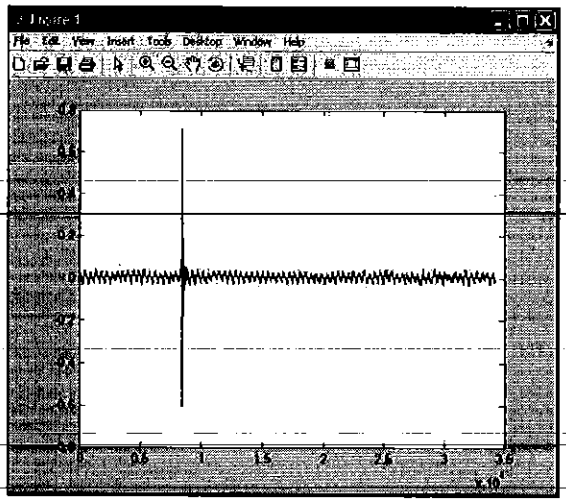
3.3.4 ใช้คำสั่ง plot เพื่อทำการดูภาพของสัญญาณที่เลือกช่วงเรียบร้อยแล้ว



**รูปที่ 3.3** หน้าต่างของโปรแกรม MATLAB

**ตัวอย่าง**ถ้าต้องการอ่านข้อมูลแฟ้มที่ชื่อ sweet.wav จะมีขั้นตอนการป้อนคำสั่ง**ดังนี้**

```
>> [y,fs] = wavread('sweet.wav');      : คำสั่งเพื่อทำการอ่านข้อมูลที่บันทึกไว้
>> plot(y)                             : คำสั่งเพื่อเรียกดูภาพของสัญญาณเสียง
>> y1=y(2400:2800);                    : คำสั่งกำหนดตำแหน่งเริ่มต้น และสิ้นสุดของสัญญาณโดย
                                         จำนวนข้อมูลที่กำหนดตำแหน่งคือ 4000 ข้อมูล
>> plot(y1)                            : คำสั่งเพื่อเรียกดูภาพสัญญาณที่กำหนดตำแหน่งเริ่มต้นและ
                                         สิ้นสุด
```



**รูปที่** 3.4 ภาพที่ได้จากอ่านข้อมูลแฟ้มที่ชื่อ sweet.wav เพื่อกำหนดขอบเขตของสัญญาณ

## 3.4 การวิเคราะห์สัญญาณในโดเมนเวลา (Time Domain)

ขั้นตอนการวิเคราะห์สัญญาณในโดเมนเวลา (Time domain)

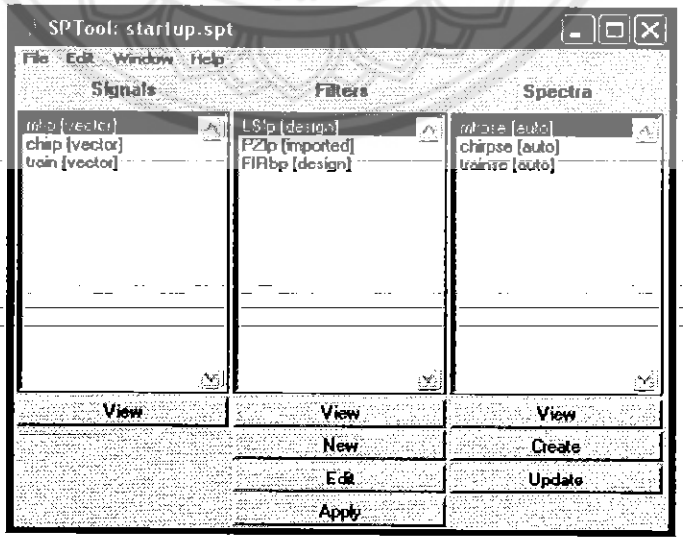
3.4.1 ทำการเปิดโปรแกรม MATLAB ขึ้นมา

3.4.2 ทำการเลือก Current Directory ที่เก็บข้อมูลเสียงนามสกุล .wav ที่บันทึกไว้

3.4.3 พิมพ์คำสั่ง wavread แล้วตามด้วยชื่อแฟ้ม ที่ต้องการอ่านข้อมูล

3.4.4 ใช้คำสั่ง plot เพื่อทำการดูภาพของสัญญาณที่เลือกช่วงเรียบร้อยแล้ว

3.4.5 ใช้คำสั่ง sptool เพื่อเข้าสู่โปรแกรมที่ใช้ในการคำนวนจะได้หน้าต่างโปรแกรม sptool




**รูปที่ 3.5** ภาพหน้าต่างของโปรแกรม SPTOOL

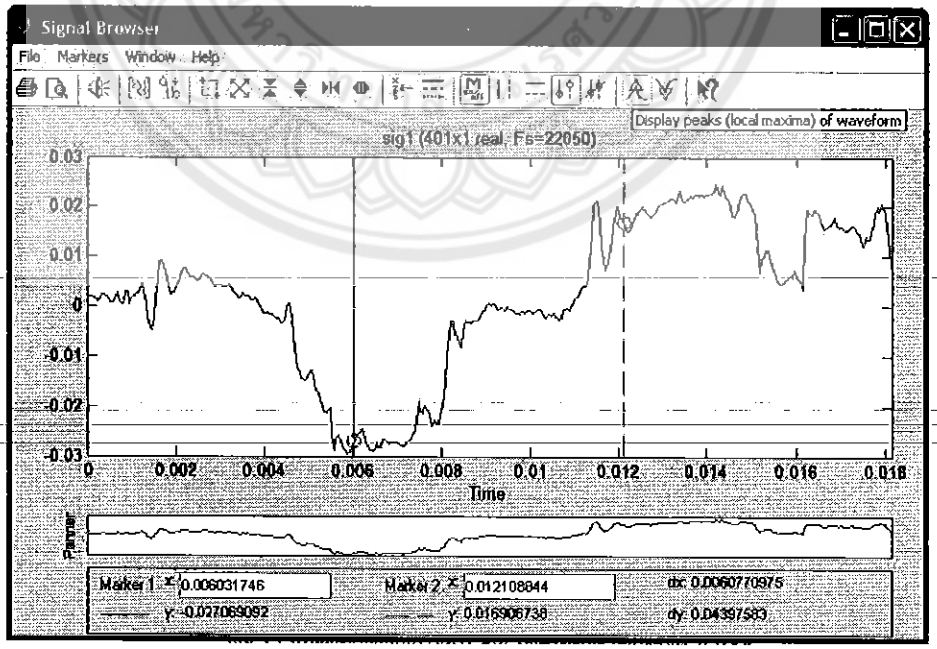
3.4.6 ทำการนำเข้า (Import) ข้อมูลโดยเลือกเข้าไปที่ File เลือก import แล้วทำการป้อนค่า y1 (สัญญาณที่ได้ กำหนดขอบเขตเริ่มต้นและสิ้นสุดของสัญญาณแล้ว) เป็นข้อมูล (Data) และป้อนค่า fs ความถี่ของการสุ่ม (Sampling) เป็นข้อมูลความถี่การสุ่ม (Sampling Frequency)

3.4.7 จะได้สัญญาณ (Signal) เป็นข้อมูลของสัญญาณที่ทำการนำเข้า (Import) แล้ว

3.4.8 เลือกที่ View เพื่อดูภาพของสัญญาณที่ได้

ตัวอย่างถ้าต้องการอ่านข้อมูลที่ชื่อแฟ้มคือ sweet.wav จะมีขั้นตอนการป้อนคำสั่งดังนี้

```
>> [y,fs] = wavread('sweet.wav');      : คำสั่งเพื่อทำการอ่านข้อมูลที่บันทึกไว้
>> plot(y)                             : คำสั่งเพื่อเรียกดูภาพของสัญญาณเสียง
>> y1=y(2400:2800);                    : คำสั่งกำหนดตำแหน่งเริ่มต้นและสิ้นสุดของสัญญาณ
                                         โดยจำนวนข้อมูลที่กำหนดตำแหน่งคือ 4000 ข้อมูล
>> plot(y1)                            : คำสั่งเพื่อเรียกดูภาพสัญญาณที่กำหนดตำแหน่ง
เริ่มต้น
                                         และสิ้นสุดแล้ว
>>sptool                              : คำสั่งเพื่อเข้าสู่หน้าต่างโปรแกรมเพื่อใช้วิเคราะห์
สัญญาณ
                                         ในโดเมนเวลา (Time domain)
```




**รูปที่ 3.6** ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนเวลา (Time Domain) โดยใช้คำสั่ง SPTOOL

การวิเคราะห์สัญญาณในโดเมนเวลานั้น เป็นการพิจารณาลักษณะของสัญญาณอย่างคร่าว ๆ ด้วยการเปรียบเทียบสัญญาณของสับปะรดแต่ละผล โดยการมองด้วยตาเปล่า ซึ่งเป็นการพิจารณาเบื้องต้นเท่านั้น

## 3.5 การวิเคราะห์สัญญาณในโดเมนความถี่ (Frequency Domain)

ขั้นตอนการวิเคราะห์สัญญาณในโดเมนความถี่ (Frequency Domain)

3.5.1 ทำการเปิดโปรแกรม MATLAB ขึ้นมา

3.5.2 ทำการเลือก Current Directory ที่เก็บข้อมูลเสียงนามสกุล .wav ที่บันทึกไว้

3.5.3 พิมพ์คำสั่ง wavread แล้วตามด้วยชื่อแฟ้ม ที่ต้องการอ่านข้อมูล

3.5.4 ใช้คำสั่ง plot เพื่อทำการดูภาพของสัญญาณ และทำการเลือกช่วงของสัญญาณที่ต้องการ

3.5.5 ใช้คำสั่ง plot เพื่อทำการดูภาพของสัญญาณที่เลือกช่วงเรียบร้อยแล้ว

3.5.6 ใช้คำสั่ง sptool เพื่อเข้าสู่โปรแกรมที่ใช้ในการคำนวนจะได้หน้าต่างโปรมแกรมดังรูปที่ 3.5

3.5.7 ทำการนำเข้า (Import) ข้อมูลโดยคลิกเข้าไปที่ File เลือก Import จากนั้นทำการป้อนค่า y1 (สัญญาณที่ได้กำหนด ขอบเขตเริ่มต้นและสิ้นสุดของสัญญาณแล้ว) เป็นข้อมูล (Data) และป้อนค่า fs ความถี่ของการสุ่ม (Sampling Frequency) เป็นข้อมูลการสุ่ม

3.5.8 จะได้สัญญาณ (Signal) เป็นข้อมูลของสัญญาณที่ทำการนำเข้า (Import) แล้ว

3.5.9 เลือกที่ Create ที่ช่องของ Spectra เพื่อทำการแปลงสัญญาณที่ได้ให้เป็นการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) โดยทำการเลือกค่าในช่อง Method เป็นแบบฟูเรียร์แบบเร็ว และค่า Nfft = 4096 โดยที่ Nfft คือ จำนวนข้อมูลที่ทำการแปลงข้อมูลเป็นการแปลงฟูเรียร์แบบเร็ว

ในการกำหนดค่า Nfft นั้นจะกำหนดให้มีค่าใกล้เคียงกับจำนวนข้อมูลที่มีอยู่ และการกำหนดนั้นจะกำหนดค่า $Nfft = 2^n$ เพื่อให้ข้อมูลที่ทำการแปลงฟูเรียร์แบบเร็ว เกิดความคลาดเคลื่อนน้อยที่สุด ซึ่งในการตัดสัญญาณนั้น สัญญาณที่พิจารณามีจำนวนข้อมูลทั้งหมด 4000 จุด ดังนั้นจึงกำหนดค่า $Nfft = 2^{12} = 4096$ ซึ่งเป็นค่าที่ใกล้เคียงกับจำนวนข้อมูลมากที่สุดและเป็นค่าที่มากกว่าจำนวนข้อมูล เพราะการแปลงฟูเรียร์แบบเร็ว ใน MATLAB จะเพิ่มข้อมูลที่มีค่าเป็น 0 เข้าไปให้ครบ แต่ถ้าค่า Nfft น้อยกว่าค่าจำนวนข้อมูลที่มี ในการแปลงฟูเรียร์แบบเร็วในโปรแกรม MATLAB นั้น จะมีการตัดสัญญาณบางส่วนซึ่งอาจเป็นส่วนที่มีความจำเป็นออกไป ซึ่งอาจทำให้เกิดความผิดพลาดได้

ตัวอย่างถ้าต้องการอ่านข้อมูลที่ชื่อแฟ้มคือ sweet.wav จะมีขั้นตอนการป้อนคำสั่งดังนี้

>> [y,fs] = wavread('sweet.wav'); : คำสั่งเพื่อทำการอ่านข้อมูลที่บันทึกไว้

>> plot(y) : คำสั่งเพื่อเรียกดูภาพของสัญญาณเสียง

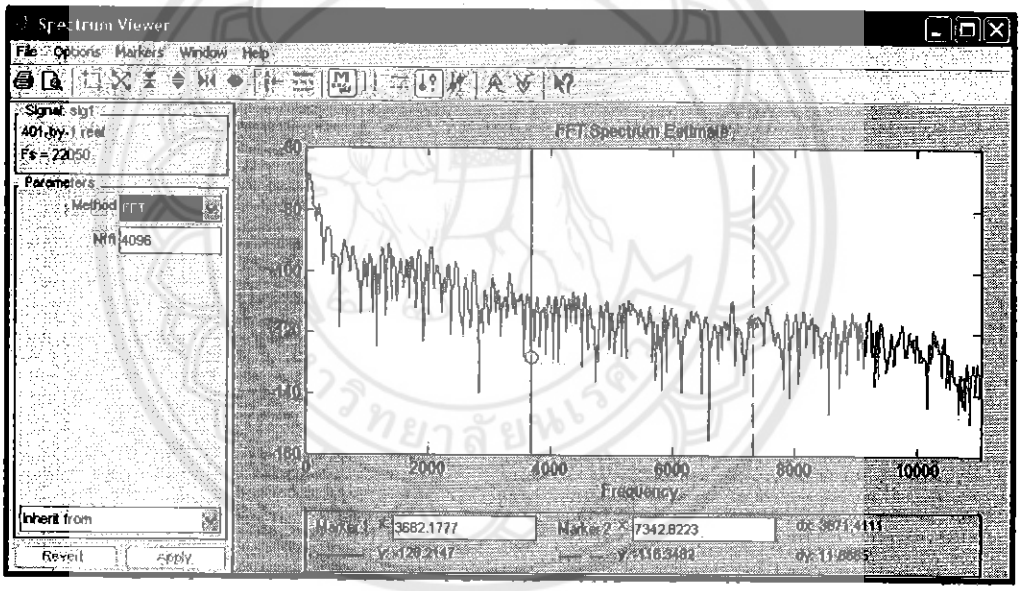
>> y1=y(2400:2800); : คำสั่งกำหนดตำแหน่งเริ่มต้น และสิ้นสุดของสัญญาณ

โดยจำนวนข้อมูลที่กำหนดตำแหน่งคือ 4000 ข้อมูล

>> plot(y1) : คำสั่งเพื่อเรียกดูภาพสัญญาณที่กำหนดตำแหน่งเริ่มต้นและสิ้นสุดแล้ว

>>sptool : คำสั่งเพื่อเข้าสู่หน้าต่างโปรแกรมเพื่อใช้วิเคราะห์สัญญาณ

ในโดเมนความถี่ (Frequency domain)




**รูปที่ 3.7** ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ โดยใช้คำสั่ง SPTOOL

จากรูปที่ 3.7 เป็นรูปของสัญญาณที่ทำการนำเข้า (Import) ข้อมูลของ y1 เข้ามาวิเคราะห์ใน SPTOOL โดยนำสัญญาณที่ได้มาทำการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) ซึ่งก็จะได้สัญญาณที่การแปลงฟูเรียร์แบบเร็วแล้ว จากโปรแกรม SPTOOL

## 3.6 การออกแบบโปรแกรมสำหรับการคัดแยกผลสับปะรด

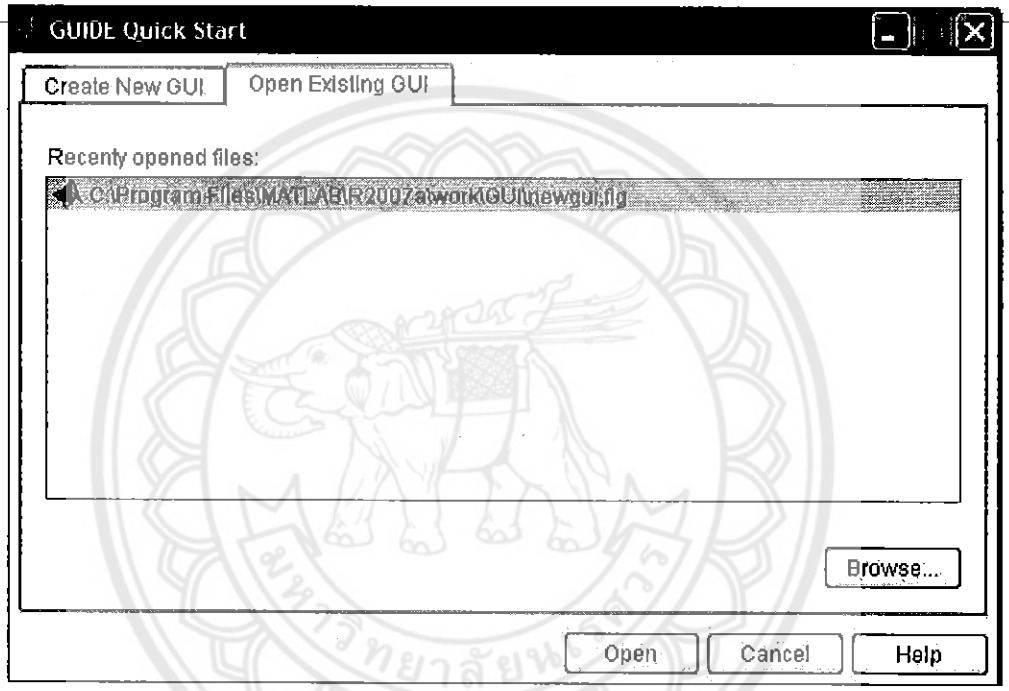
มีขั้นตอนการทำงานดังนี้

3.6.1 ทำการเปิดโปรแกรม MATLAB ขึ้นมา

3.6.2 ทำการเลือก Current Directory ที่เก็บข้อมูลเสียงนามสกุล .wav ที่บันทึกไว้

3.6.3 ใช้คำสั่ง guide เพื่อทำการสร้าง GUI

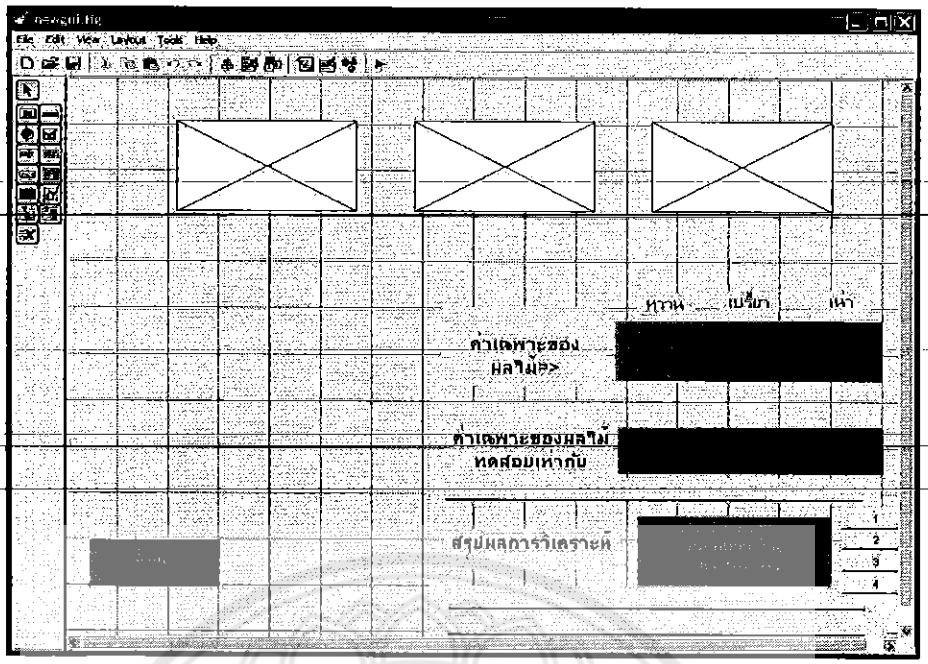
3.6.4 หน้าต่าง GUIDE QUICK START แสดงขึ้นมา ให้เลือกแถบ Open Existing GUI ในที่นี้ได้มีการจัดสร้าง GUI ไว้เรียบร้อบ แล้วกด Open




รูปที่ 3.8 ภาพหน้าต่างของโปรแกรม GUIDE QUICK START

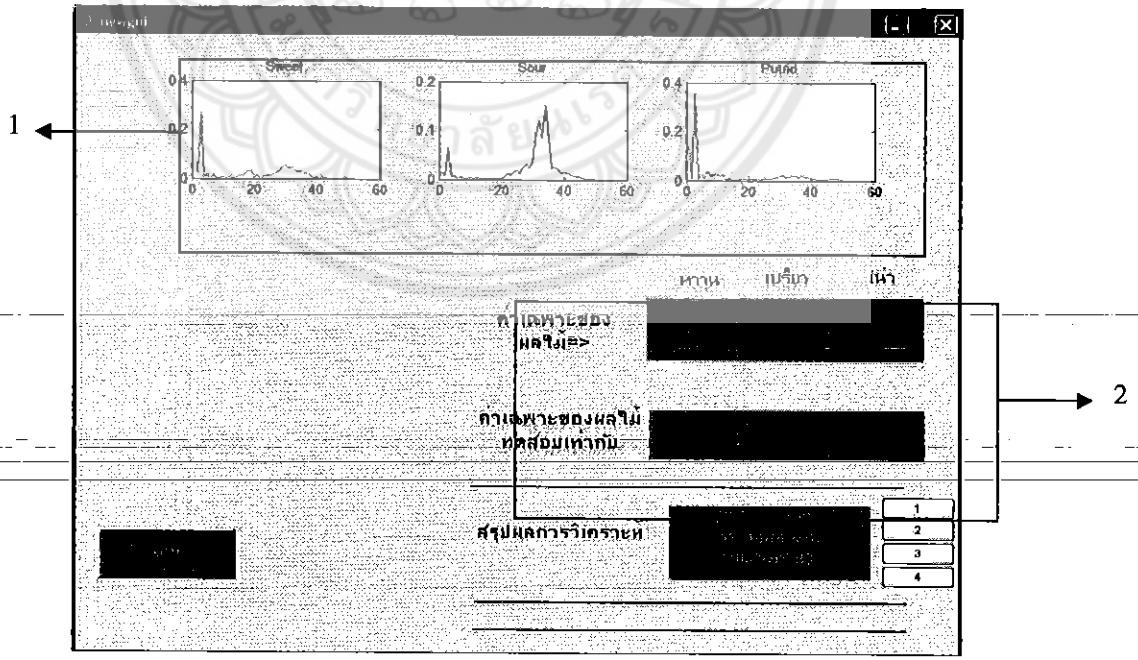
ห้องสมุดคณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร 1800136

วศ.
ก796
2550.




**รูปที่ 3.9** ภาพหน้าต่างของโปรแกรม GUI ที่ได้มีการสร้างไว้เป็นที่เรียบร้อย

3.6.5 ทำการประมวลผล GUI ที่ได้มีการสร้างไว้ กดปุ่ม สามเหลี่ยมสีเขียว เพื่อทำการประมาลผล




**รูปที่ 3.10** ภาพหน้าต่างของโปรแกรม GUI ที่ได้ผ่านการปะมวลผล

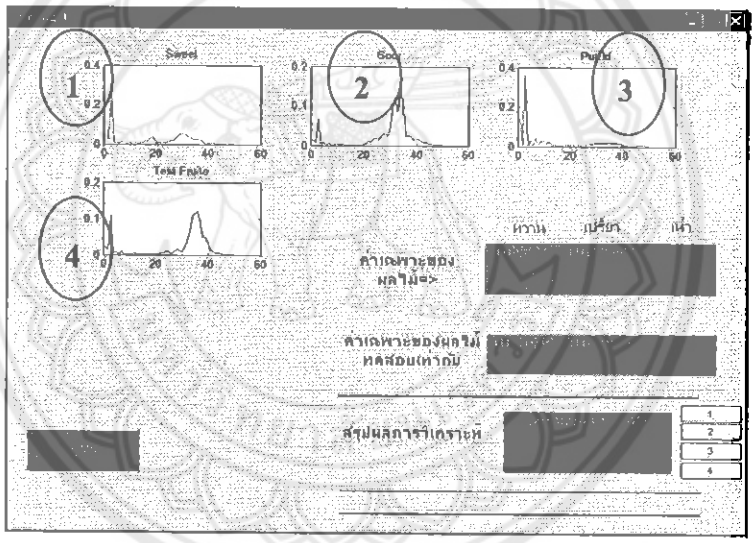
รูปแบบลักษณะแนวคิดการออกแบบ หน้าตาของ GUI ในรูปที่ 3.16

หมายเลข 1 หมายถึง ฐานข้อมูลของสัญญาณเสียงต้นแบบที่มีการ คัดแยกผลสับปะรด หวาน เปรี้ยว และ เน่า เป็นสัญญาณเสียงแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform)

หมายเลข 2 หมายถึง ค่าการแปลงฟูเรียร์แบบเร็ว ของผลสับปะรดทั้ง 3 ชนิด โดยผ่าน โปรแกรมที่ได้มีการหาค่าโปรแกรมเพื่อทดสอบสับปะรดตัวอย่างเทียบกับสับปะรดต้นแบบ สร้างเป็น M file ในการทำงานควบคุม GUI การแปลงเป็นสัญญาณเสียงในหมายเลข 1 และแปลงเป็นค่าของเสียงต้นแบบ ในช่อง หมายเลข 2

3.6.6 ทำการเปรียบเทียบข้อมูล โดยกดปุ่ม Load เพื่อทำการเพิ่มค่า

3.6.7 เปรียบเทียบค่าที่ทำการเพิ่มกับค่าเสียงต้นแบบ ในการคัดแยกผลสับปะรด




**รูปที่ 3.11** ภาพหน้าต่างของของสัญญาณเสียงที่นำมาทดสอบกับโปรแกรม

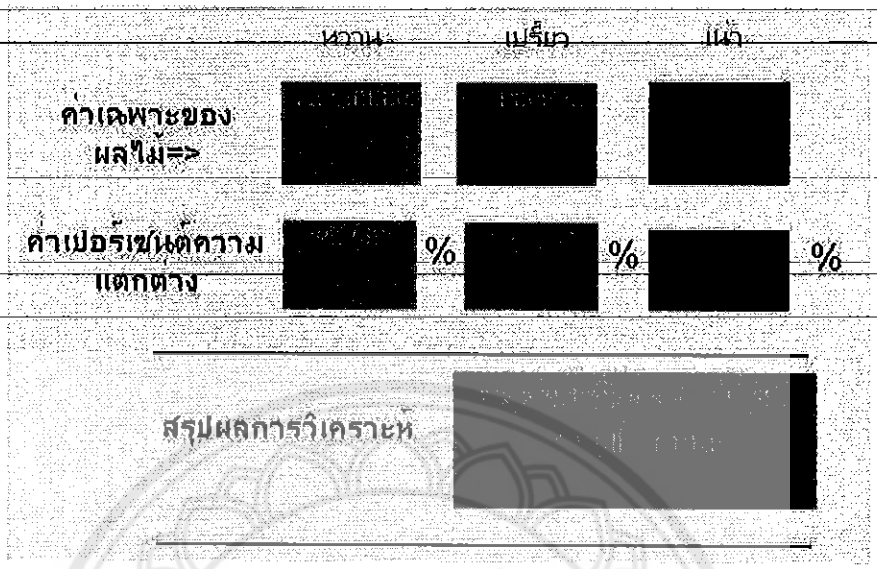
หมายเลข 1 คือ สัญญาณ FFT ของ Sweet.wav ซึ่งเป็นต้นแบบของผลหวานแสดงให้ดูในช่วงที่มีผลมาก

หมายเลข 2 คือ สัญญาณ FFT ของ Sour.wav ซึ่งเป็นต้นแบบของผลเปรี้ยวแสดงให้ดูในช่วงที่มีผลมาก

หมายเลข 3 คือ สัญญาณ FFT ของ exp.wav ซึ่งเป็นต้นแบบของผลเน่าแสดงให้ดูในช่วงที่มีผลมาก

หมายเลข 4 คือ สัญญาณ FFT ของผลที่ต้องการนำมาทดสอบในการเคาะ

นำตัวอย่างสับปะรดมาทดสอบกับสัญญาณเสียงต้นแบบแล้วดูผลการทดลองที่เกิดขึ้น เพื่อทดสอบความถูกต้องของโปรแกรม ค่าที่ออกมาดูได้จากรูปที่ 3.12




**รูปที่ 3.12** ภาพหน้าต่างของการเปรียบเทียบค่าทดสอบสับปะรดตัวอย่างเทียบกับสับปะรดต้นแบบ

สรุปการทดสอบจากการทดสอบจะเห็นได้ว่าในการทดสอบโปรแกรมนั้นเมื่อใช้แฟ้ม sour.wav เป็นตัวต้นแบบแล้ว ค่าเฉพาะของผลเปรี้ยวที่ได้จะมีค่าใกล้เคียงหรือเท่ากันกับตัวต้นแบบ ซึ่งจะเป็นผลที่น่าเชื่อถือมากที่สุด ดังนั้นจึงเลือกใช้แฟ้ม sour.wav เป็นต้นแบบของสับปะรดผลเปรี้ยว เป็นการทดสอบโปรแกรมว่ามีการทำงานถูกต้องตามที่คาดไว้หรือไม่

## 3.7 การวิเคราะห์ปัจจัยที่มีผลต่อข้อมูล

อุปกรณ์ที่นำมาศึกษา ได้แก่

- สับปะรดจำนวน 3 ลูก
- ไม้เคาะ
- ไขควง ใช้เปรียบเทียบเมื่อใช้วัสดุในการเคาะต่างกัน

การทดสอบการเคาะสับปะรดด้วยวิธีที่แตกต่างกัน เพื่อจุดประสงค์และมีวิธีการทดลองดังนี้

**3.7.1 ทดสอบว่าคนเคาะต่างกันมีผลกับข้อมูลที่ได้หรือไม่** โดยการที่เปลี่ยนคนเคาะ แต่จะควบคุมปัจจัยอื่น ๆ อันได้แก่ ใช้อุปกรณ์เคาะชนิดเดียวกัน ผลสับปะรดผลเดียวกัน เคาะบริเวณเดียวกัน แล้วนำผลที่ได้มาเปรียบเทียบกัน

Js1 เคาะโดยผู้หญิง

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Js2 เคาะโดยผู้หญิงเคาะครั้งที่ 2

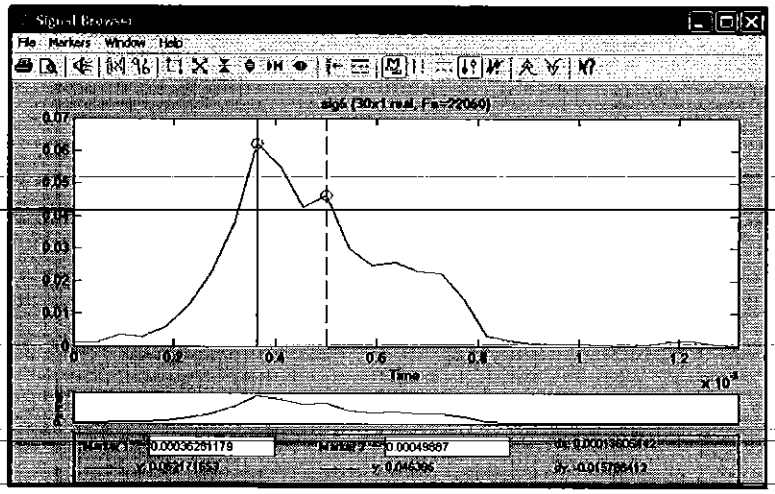
- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Ns1 เคาะโดยผู้ชาย

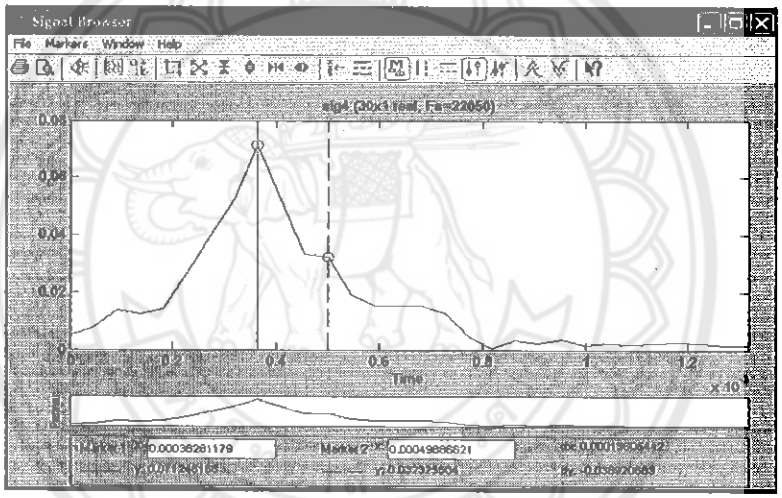
- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Ns2 เคาะโดยผู้ชายเคาะครั้งที่ 2

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz




**รูปที่** 3.13 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Js1 ซึ่งเคาะโดยผู้หญิง




**รูปที่** 3.14 ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Ns1 ซึ่งเคาะโดยผู้ชาย

ผลจากการทดลองจะเห็นได้ว่าเมื่อนำข้อมูลที่แปลงฟูเรียร์แบบเร็ว แล้วนำมาเปรียบเทียบกัน จะได้ผลที่เหมือนกัน สรุปว่าไม่ว่าจะเป็นใครเคาะ และความแรงในการเคาะไม่มีผลต่อข้อมูลเสียงที่บันทึกได้ จะมีผลแค่แอมพลิจูด (Amplitude) ของสัญญาณเสียงในโดเมนเวลา (Time Domain) เท่านั้น แต่ผลที่ได้ในโดเมนความถี่ (Frequency Domain) ได้ผลที่เหมือนกัน คือ ยอด (Peak) ของสัญญาณจะขึ้นที่ความถี่เดียวกัน

**3.7.2 ทดสอบว่าตำแหน่งที่เคาะของผลสับปะรดที่แตกต่างกันมีผลกับข้อมูลที่ได้หรือไม่** โดยการเคาะที่ด้านข้าง เปรียบเทียบกับเคาะที่บริเวณท้ายของผลสับปะรด โดยควบคุมตัวแปรตัวอื่น ๆ ได้แก่ ใช้คนเคาะคนเดียวกัน ใช้ผลสับปะรดผลเดียวกัน ใช้อุปกรณ์เคาะอันเดียวกัน แล้วนำผลที่ได้มาเปรียบเทียบกัน

Js1 เคาะที่บริเวณด้านข้างของผล

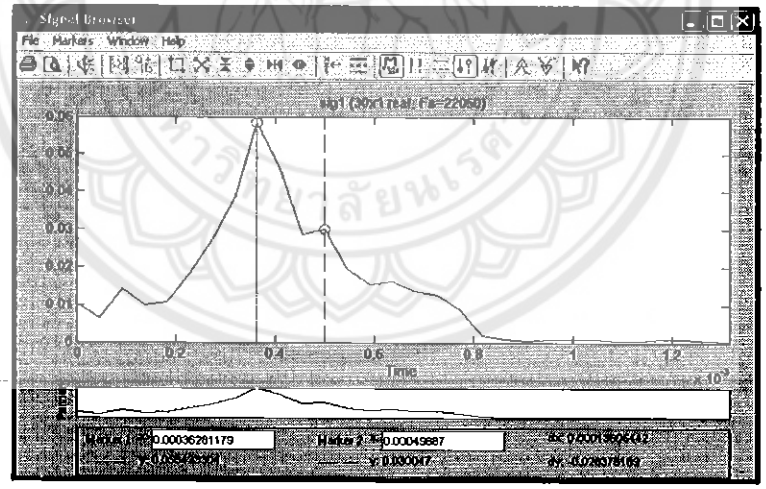
- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Jb1 เคาะที่บริเวณด้านท้ายของผล

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Jb2 เคาะที่บริเวณด้านท้ายของผลครั้งที่ 2

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz




**รูปที่ 3.15** ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jb1 ซึ่งเคาะบริเวณด้านข้างผล

ผลการทดลองเมื่อนำมาเปรียบเทียบจะเห็นได้ว่า เมื่อทำให้อยู่ในโดเมนความถี่(Frequency Domain) ได้ยอด (Peak) แรก อยู่ที่ความถี่เดียวกัน คือประมาณ 362 MHz แต่ยอด (Peak) ที่ 2 อยู่ที่ความถี่ต่างกัน

**3.7.3 ทดสอบว่าขนาดของผลสับปะรดแตกต่างกันมีผลต่อข้อมูลที่ได้หรือไม่** โดยการหาสับปะรด ที่สุกเท่า ๆ กัน และลักษณะ พันธุ์ เหมือนกัน แต่มีน้ำหนักที่แตกต่างกันมาเคาะแล้วนำผลที่ได้ มาเปรียบเทียบกัน โดยควบคุมตัวแปรอื่น ๆ ที่เหลือเช่นกัน

Js1 เคาะผลที่มีน้ำหนัก 1.2 kg

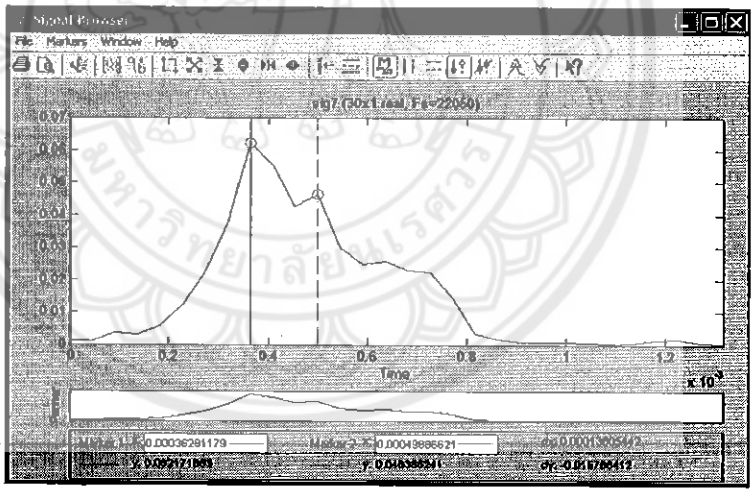
- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Jm1 เคาะผลที่มีน้ำหนัก 1.7 kg

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Jm2 เคาะผลที่มีน้ำหนัก 1.7 kg

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz




**รูปที่ 3.16** ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jm1 ซึ่งเคาะผลที่น้ำหนัก 1.7 kg

ผลการทดลองที่ได้จากการนำสับปะรดที่พันธุ์เดียวกัน มีรูปร่างกลมเหมือนกัน สุกเท่า ๆ กัน แต่มีน้ำหนักที่ไม่เท่ากัน มาทดลองและทำให้อยู่ในโดเมนความถี่ (Frequency Domain) แล้วมาเปรียบเทียบกัน จะได้ว่าผลของยอด (Peak) แรกจะมีค่าเท่ากัน ประมาณ 362 MHz

**3.7.4 ทดสอบว่าอุปกรณ์ที่นำมาเคาะแตกต่างกันมีผลต่อข้อมูลที่ได้หรือไม่** โดยการนำอุปกรณ์อื่นๆได้แก่ ไม้รูปร่างอื่น ด้ามไขควงซึ่งเป็นพลาสติก และปลายไขควงที่เป็นเหล็ก มาเปรียบเทียบผลที่ได้ โดยควบคุมตัวแปรอื่น ๆ ที่เหลือให้คงที่

Js1 เคาะผลโดยใช้สันไม้

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz

Ji1 เคาะผลโดยใช้ปลายไขควงซึ่งเป็นเหล็ก

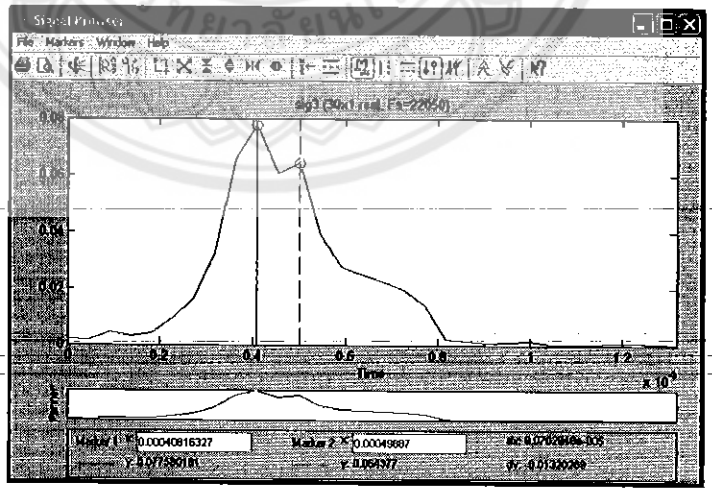
- peak 1 = 362.281179 MHz
- peak 2 = 498.87 M Hz

Jp1 เคาะผลโดยใช้ด้ามไขควงที่เป็นพาสติก

- peak 1 = 408.16327 MHz
- peak 2 = 498.87 M Hz

Jw1 เคาะผลโดยใช้ด้ามไม้

- peak 1 = 362.281179 MHz
- peak 2 = 498.86621 MHz




**รูปที่ 3.17** ภาพสัญญาณที่ทำการวิเคราะห์ในโดเมนความถี่ของแฟ้ม Jw1 ซึ่งเคาะโดยใช้ด้ามไขควง

ผลการทดลองเมื่อใช้อุปกรณ์ที่แตกต่างกันสรุปได้ว่า เมื่อใช้ไม้เคาะเหมือนกัน แม้ว่าไม้นั้นจะมีลักษณะที่แตกต่างกัน ก็จะได้ผลความถี่ทั้งสองยอด (Peak) เป็นที่เดียวกัน แต่เมื่อเปลี่ยนเป็นอุปกรณ์อื่น คือเป็นเหล็กและเป็นพลาสติก จะได้ผลที่ยอด (Peak) แรกที่ความถี่เดียวกัน แต่ยอด (Peak) ที่ 2 ความถี่ต่างกัน

## 3.8 สรุปผลจากการวิเคราะห์ปัจจัยที่มีผลต่อข้อมูลที่จะนำมาวิเคราะห์

3.8.1 ไม่ว่าจะเป็นใครเคาะ และความแรงในการเคาะไม่มีผลต่อการวิเคราะห์

3.8.2 ตำแหน่งเคาะที่แตกต่างกันไม่มีผลต่อการวิเคราะห์

3.8.3 ขนาดของสับปะรดที่ต่างกันไม่มีผลต่อการวิเคราะห์

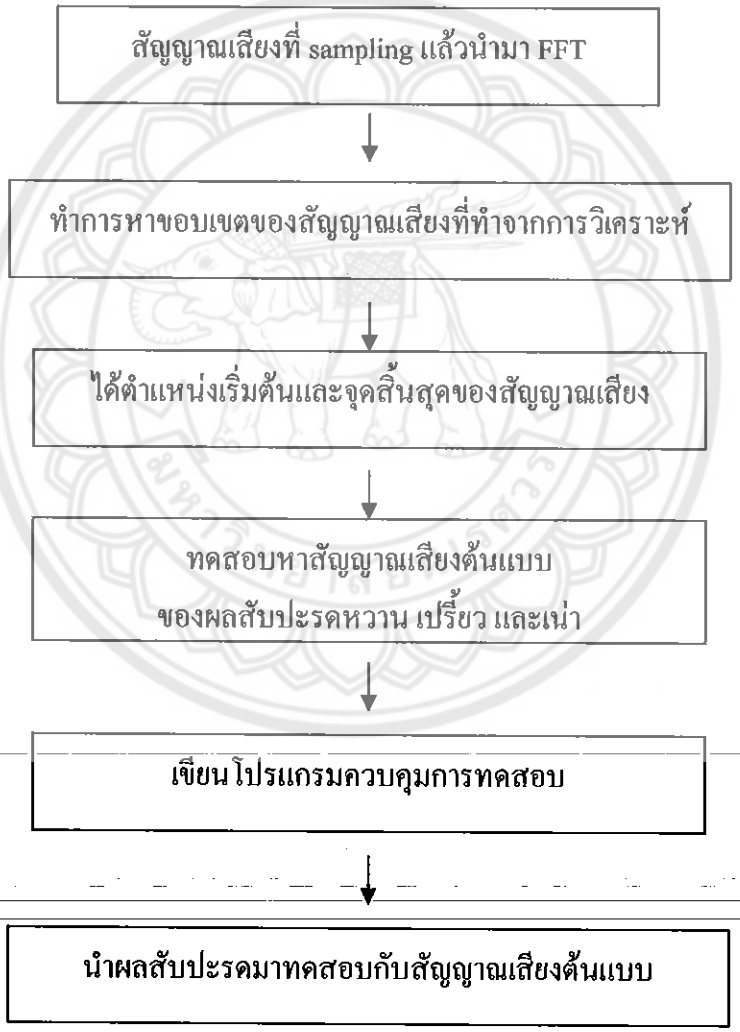
3.8.4 อุปกรณ์เคาะที่แตกต่างกันไม่มีผลต่อการวิเคราะห์

# บทที่ 4

# การทดลอง

## 4.1 สรุปการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์

เมื่อได้สัญญาณที่กำหนดตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดแล้ว ก็จะนำข้อมูลที่ได้มาทำการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) และนำมาเปรียบเทียบกับการแปลงฟูเรียร์แบบเร็ว ของสัญญาณต้นแบบของสับปะรดหวาน เปรี้ยว และผลเน่า ซึ่งหลักการทำงานจะถูกแบ่งเป็นขั้นตอน ดังรูปที่ 4.1




**รูปที่ 4.1** ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์

## 4.2 การวิเคราะห์เสียงด้วยวิธีแปลงฟูเรียร์แบบเร็ว โดยการใช้โปรแกรม MATLAB

หลักการวิเคราะห์เสียงโดยวิธีการแปลงฟูเรียร์แบบเร็ว จะถูกแบ่งออกเป็น 5 ขั้นตอน คือ เริ่มแรกเมื่อมีสัญญาณเสียงที่กำหนดตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดแล้ว จะนำสัญญาณที่ได้มาทำการแปลงฟูเรียร์แบบเร็ว ต่อจากนั้นนำสัญญาณที่ได้จากการแปลงฟูเรียร์แบบเร็วนั้น มาทำการเปรียบเทียบกับสัญญาณข้อมูลตัวอื่นๆ นำค่าที่ได้มาพิจารณาเพื่อหาสัญญาณต้นแบบของสับปะรดผลหวาน เปรี้ยว และเน่า จากนั้นทำการเขียนเป็นโปรแกรมเพื่อใช้ในการทดสอบและนำตัวอย่างอื่นๆมาทดสอบกับสับปะรดต้นแบบด้วยการใช้โปรแกรมที่เขียนขึ้นมา

ขั้นตอนการทำงาน

4.2.1 ทำการเปิดโปรแกรม MATLAB ขึ้นมา

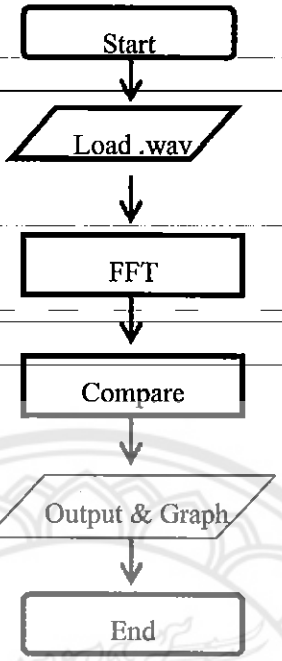
4.2.2 ทำการเลือก current directory ที่เก็บข้อมูลเสียงนามสกุล .wav ที่บันทึกไว้

4.2.3 ทำการ sampling สัญญาณเพื่อกำหนดตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดของสัญญาณ

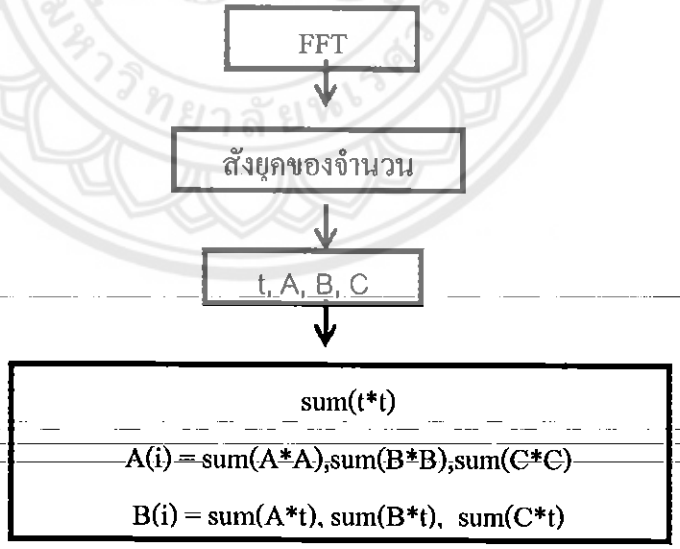
4.2.4 ใช้คำสั่ง FFT เพื่อเปลี่ยนสัญญาณในโดเมนเวลา (Time Domain) ให้อยู่ในโดเมนความถี่ (Frequency Domain) โดยวิธีการแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform)

4.2.5 เขียนโปรแกรมเพื่อทดสอบสับปะรดตัวอย่างเทียบกับสับปะรดต้นแบบ ในส่วนแรกนี้จะเป็นการนำตัวอ้างอิงที่เราได้มา มาตัดสัญญาณโดยใช้ตาเปล่า เพื่อให้ได้สัญญาณอ้างอิง ที่มีลักษณะคงที่ในทุก ๆ ครั้งที่มีการทดสอบผลสับปะรดทำให้การวิเคราะห์เป็นไปอย่างถูกต้องมากที่สุด

## 4.3 การทำงานของโปรแกรมที่ใช้ในการวิเคราะห์ด้วยวิธีแปลงฟูเรียร์แบบเร็ว

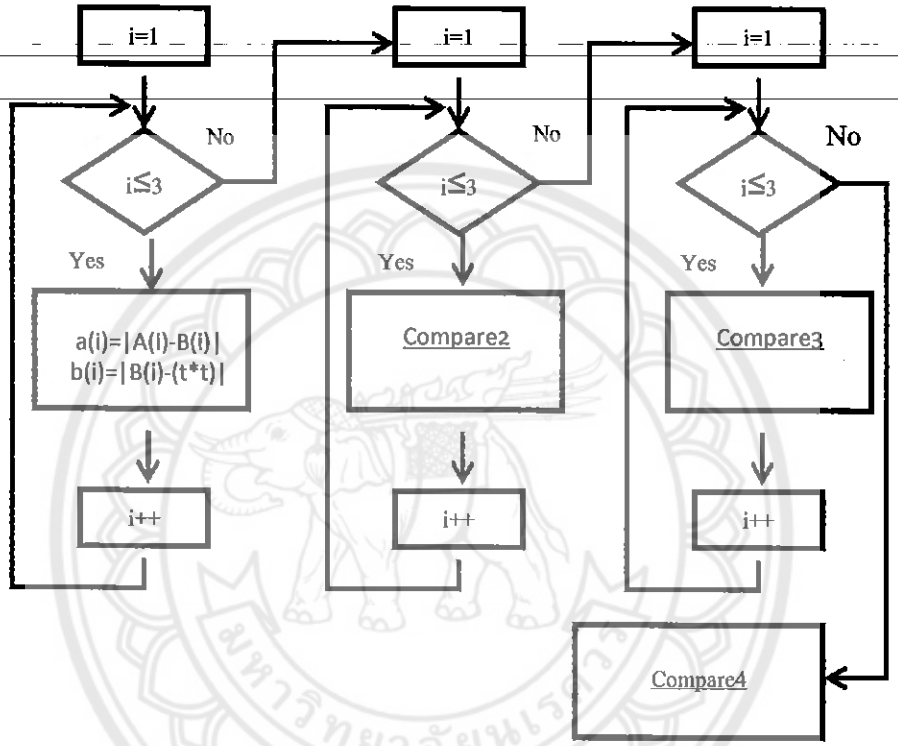



**รูปที่ 4.2** แผนผังการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์

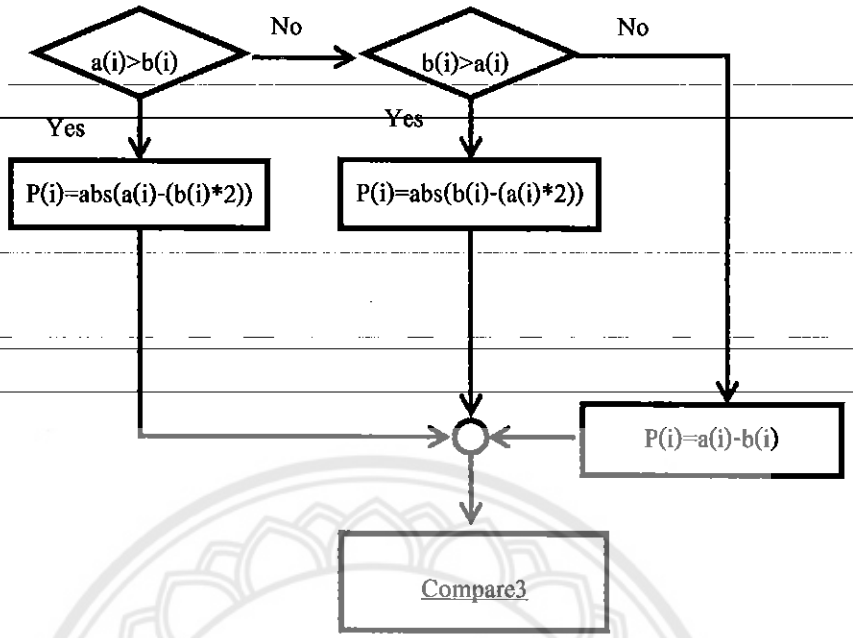



**รูปที่ 4.3** แผนผังการทำงานของโปรแกรมส่วน FFT

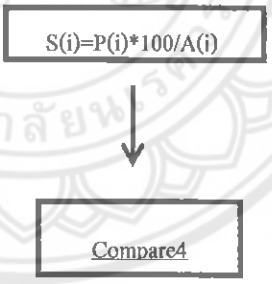
จากรูปที่ 4.3 คือการรับไฟล์เสียงที่จะนำมาวิเคราะห์มาเข้าขบวนการแปลงฟูเรียร์แบบเร็ว แล้วเราจะได้จำนวนเชิงซ้อน เราเอาจำนวนเชิงซ้อนคูณกับค่าสังยุคของตัวเอง เพื่อเราจะเอาเฉพาะค่าจำนวนเต็มจากนั้นก็เก็บไว้ที่ตัวแปร t ส่วนตัวแปร A คือค่าของสับปะรดหวาน B คือค่าของสับปะรดเปรี้ยว C คือค่าของสับปะรดเน่า



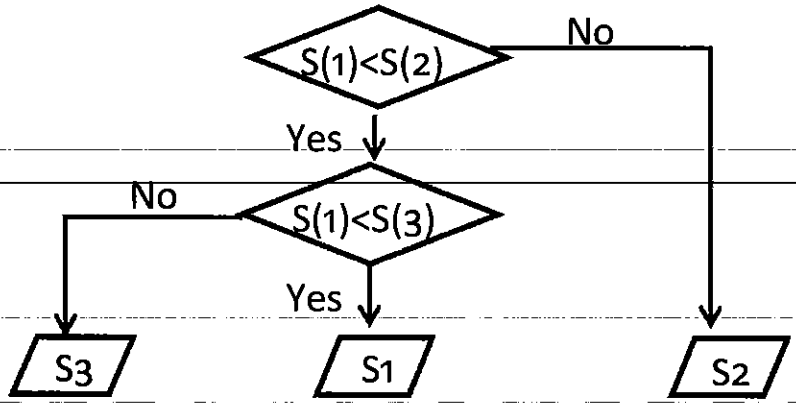

**รูปที่ 4.4** แผนผังการทำงานของโปรแกรมส่วน Compare



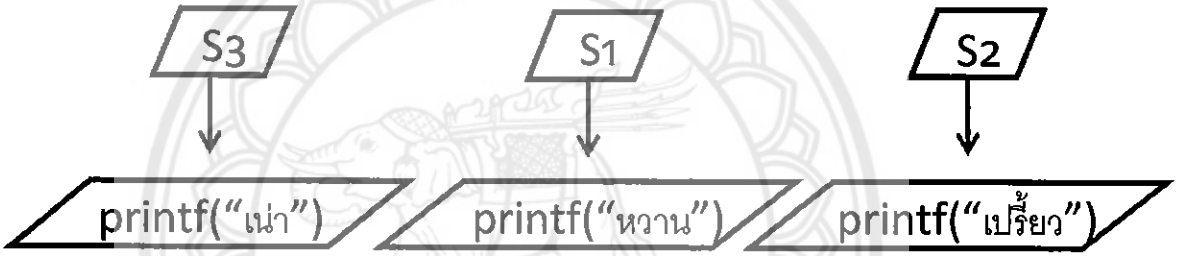

**รูปที่ 4.5** แผนผังการทำงานของโปรแกรมส่วน Compare2




**รูปที่ 4.6** แผนผังการทำงานของโปรแกรมส่วน Compare3




รูปที่ 4.7 แผนผังการทำงานของโปรแกรมส่วน Compare4




รูปที่ 4.8 แผงผังการทำงานของโปรแกรมส่วน Output

## 4.4 การทดสอบสับปะรดตัวอย่างกับสับปะรดต้นแบบด้วยการใช้โปรแกรม

**4.4.1** นำผลสับปะรดหวานมาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่าเพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี soundSIG ได้ผลดังตารางที่ 4.1

**ตารางที่ 4.1** ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดหวาน

| File | ลักษณะภายใน | Sweet.wav | Sour.wav | Exp.wav |
|---|---|---|---|---|
| Sweetest 1 | หวาน | 6.53276 | 63.6719 | 236.5 |
| Sweetest 2 | หวาน | 25.9564 | 62.4592 | 402.66 |
| Sweetest 3 | หวาน | 5.90256 | 34.6205 | 269.648 |
| Sweetest 4 | หวาน | 0 | 31.282 | 107.525 |
| Sweetest 5 | หวาน | 0.246484 | 53.151 | 115.262 |
| Sweetest 6 | หวาน | 0.936577 | 58.9018 | 211.999 |
| Sweetest 7 | หวาน | 1.98965 | 36.9789 | 138.393 |

หมายเหตุ ค่าในตารางคิดเป็นร้อยละในการเปรียบเทียบเสียงต้นแบบ

ลักษณะภายในของสับปะรดแต่ละลูก

Sweetest 1.wav : เนื้อภายในปล้นขาว เนื้อมีสีเหลือง รสชาติหวาน เปลือกสีเหลือง

Sweetest 2-3.wav: เนื้อภายในปล้นขาว เนื้อมีสีเหลือง เปลือกเหลืองปนเขียว

Sweetest 4.wav : เนื้อภายในช่ำ เนื้อมีเหลืองบ่นช้ำ รสชาติหวาน เปลือกสีเหลืองพอดี

Sweetest 5-7.wav: เนื้อภายในช่ำ เนื้อมีเหลืองบ่นช้ำ รสชาติหวาน เปลือกสีเหลืองมาก

**รูปที่ 4.9** ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลหวาน

**4.4.2** นำผลสับปะรดเปรี้ยวมาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่า เพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี soundSIG ได้ผลดังตารางที่ 4.2

**ตารางที่ 4.2** ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดเปรี้ยว

| File | ลักษณะภายใน | Sweet.wav | Sour.wav | Exp.wav |
|---|---|---|---|---|
| Sour 1 | เปรี้ยว | 18.5353 | 7.37397 | 263.777 |
| Sour 2 | เปรี้ยว | 84.2591 | 1.41712 | 630.205 |
| Sour 3 | เปรี้ยว | 25.7727 | 22.7316 | 176.778 |
| Sour 4 | เปรี้ยว | 222.472 | 4.75175 | 965.028 |
| Sour 5 | เปรี้ยว | 133.267 | 0 | 1203.34 |
| Sour 6 | เปรี้ยว | 193.975 | 24.4011 | 775.467 |
| Sour 7 | เปรี้ยว | 26.6753 | 13.9305 | 161.964 |

หมายเหตุ ค่าในตารางคิดเป็นร้อยละ ในการเปรียบเทียบเสียงต้นแบบ

ลักษณะภายในของสับปะรดแต่ละลูก

Sour 1.wav : เนื้อภายในขาว เนื้อมีสีขาวซีด รสชาติเปรี้ยวมาก เปลือกสีเขียวเข้ม

Sour 2-4.wav: เนื้อภายในขาว เนื้อมีขาว เปลือกเหลืองปนเขียว

Sour 5.wav : เนื้อภายในขาว เนื้อมีขาวปนเหลือง รสชาติเปรี้ยว เปลือกเขียวปนเหลือง

Sour 6-7.wav: เนื้อภายในช่ำ เนื้อมีค่อนข้างเหลือง รสชาติหวานอมเปรี้ยว เปลือกค้างขางเหลือง

**รูปที่ 4.10** ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลเปรี้ยว

**4.4.3** นำผลสับปะรดเน่ามาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่าเพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี soundSIG ได้ผลดังตารางที่ 4.3

**ตารางที่ 4.3** ตารางข้อมูลที่ได้จากการทดลองเมื่อเปรียบเทียบกับสับปะรดผลเน่า

| File | ลักษณะภายใน | Sweet.wav | Sour.wav | Exp.wav |
|---|---|---|---|---|
| Exp 1 | เน่า | 21.9453 | 57.6488 | 0 |
| Exp 2 | เน่า | 28.0476 | 40.744 | 5.34764 |
| Exp 3 | เน่า | 72.5409 | 87.5338 | 53.3332 |
| Exp 4 | เน่า | 52.7264 | 81.2982 | 28.2286 |
| Exp 5 | เน่า | 25.5928 | 86.3575 | 127.227 |

ลักษณะภายในของสับปะรดแต่ละลูก

Exp 1.wav : เนื้อภายในเหลืองมาก เนื้อมีสีเหลืองปนช่ำบริเวณ ด้านข้าง มีกลิ่น น้ำเยอะ เปลือกสีเหลืองเข้มปนดำ

Exp 2-4.wav : เนื้อภายในเหลืองมาก มีกลิ่น มีน้ำเยอะ เปลือกเหลืองอ่อน

Exp 5.wav : เนื้อภายในเหลืองมาก เนื้อมีสีเหลืองปนช่ำบริเวณ ด้านข้าง มีกลิ่น มีน้ำเยอะ เปลือกเหลือง

**รูปที่ 4.11** ภาพของสับปะรดที่ใช้เป็นต้นแบบของสับปะรดผลเน่า

**4.4.4 ทดสอบสับปะรดต้นแบบกับตัวอย่างสับปะรดชุดที่ 1**

นำข้อมูลของสับปะรดที่มีอยู่คือกลุ่ม Sweetest, Sour และ Exp มาทำการทดสอบกับสัญญาณ เสียงต้นแบบด้วยโปรแกรมที่เขียนขึ้นมา ได้ผลการทดลองดังตารางที่ 4.4

**ตารางที่ 4.4** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 1

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| Sweetest 1 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 2 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 3 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 4 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 5 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 6 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 7 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sour 1 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 2 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 5 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 6 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 7 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Exp 1 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 2 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 3 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 4 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 5 | เน่า | หวาน | หวาน | หวาน | หวาน | ผิด |

**4.4.5 ทดสอบสับปะรดต้นแบบกับตัวอย่างสับปะรดชุดที่ 2**

ทำการนำตัวอย่างสับปะรดทั้งหมด 3 ลูกที่ไม่ทราบลักษณะภายในมาทดสอบกับสัญญาณเสียงต้นแบบ แล้วดูผลการทดลองที่เกิดขึ้น เพื่อทดสอบความถูกต้องของโปรแกรม ได้ผลดังตารางที่ 4.5

**ตารางที่ 4.5** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 2

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| 14_Aug 1 | หวาน | หวาน | หวาน | เปรี้ยว | หวาน | ถูก |
| 14_Aug 2 | หวาน | เปรี้ยว | หวาน | หวาน | หวาน | ถูก |
| 14_Aug 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 14_Aug 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | หวาน | เปรี้ยว | ถูก |
| 14_Aug 5 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |

**4.4.6 ทดสอบสับปะรดต้นแบบกับตัวอย่างสับปะรดชุดที่ 3**

ทำการนำตัวอย่างสับปะรดทั้งหมด 4 ลูกที่ไม่ทราบลักษณะภายในมาทดสอบกับสัญญาณเสียงต้นแบบแล้วดูผลการทดลองที่เกิดขึ้น เพื่อทดสอบความถูกต้องของโปรแกรม ได้ผลดังตารางที่ 4.6

**ตารางที่ 4.6** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 3

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| 6_Sept 1 | เปรี้ยว | หวาน | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 6_Sept 2 | หวาน | เปรี้ยว | หวาน | หวาน | หวาน | ถูก |
| 6_Sept 3 | เน่า | เปรี้ยว | เน่า | เปรี้ยว | เปรี้ยว | ผิด |
| 6_Sept 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | หวาน | เปรี้ยว | ถูก |
| 6_Sept 5 | หวาน | หวาน | หวาน | เน่า | หวาน | ถูก |
| 6_Sept 6 | หวาน | เน่า | เน่า | เน่า | เน่า | ผิด |
| 6_Sept 7 | เปรี้ยว | เปรี้ยว | หวาน | หวาน | หวาน | ผิด |
| 6_Sept 8 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |

**4.5 ปัจจัยที่มีผลต่อการทดสอบที่ทำให้ได้ผลที่ผิดพลาด**

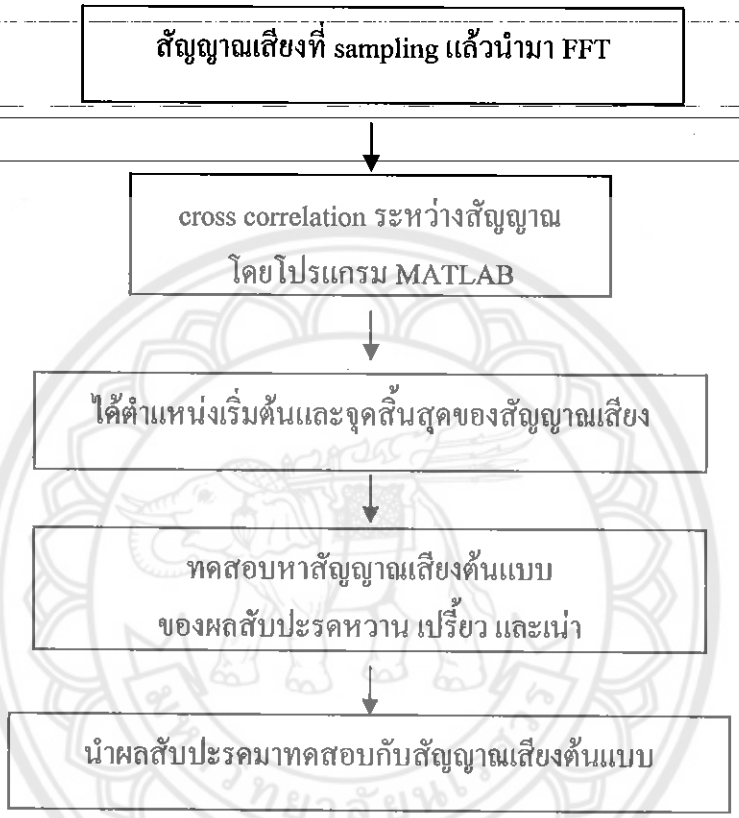
4.5.1 เมื่อสับปะรดที่นำมาทดสอบมีรูปร่างแคระแกรน บิดเบี้ยว ไม่สมบูรณ์ ผลที่ได้จากการทดสอบจะไม่ถูกต้อง

4.5.2. เมื่อสับปะรดที่นำมาทดสอบมีลักษณะที่เน่าเป็นบางส่วน ไม่เน่าทั้งลูก จะทำให้สัญญาณที่ได้นั้น มีลักษณะผสมกันระหว่างสัญญาณของผลดีและผลเน่า จะทำให้ได้ผลการทดสอบที่ผิดพลาด

มหาวิทยาลัยนเรศวร

## 4.6 การวิเคราะห์เสียงเคาะผลสับปะรดโดยวิธีสหสัมพันธ์ข้าม (Cross correlation)

เมื่อได้สัญญาณที่กำหนดตำแหน่งเริ่มต้นและตำแหน่งสิ้นสุดแล้ว ก็จะนำข้อมูลที่ได้มาทำการแปลงฟูเรียร์แบบเร็ว และนำมาเปรียบเทียบกับการแปลงฟูเรียร์แบบเร็วของสัญญาณต้นแบบของสับปะรดผลหวาน ผลเปรี้ยวและผลเน่า โดยใช้การเปรียบเทียบแบบสหสัมพันธ์ข้าม ซึ่งหลักการทำงานจะถูกแบ่งเป็นขั้นตอน ดังรูป




**รูปที่ 4.12 ขบวนการทำงานของโปรแกรมที่ใช้ในการวิเคราะห์โดยใช้สหสัมพันธ์ข้าม**

### 4.6.1 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดหวาน

**ตารางที่ 4.7** ใช้ File Sweetest 1.wav เป็นต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 1.0000 |
| Sweetest 2 .wav | 0.9254 |
| Sweetest 3 .wav | 0.9469 |
| Sweetest 4 .wav | 0.9859 |
| Sweetest 5 .wav | 0.9471 |
| Sweetest 6.wav | 0.9469 |
| Sweetest 7.wav | 0.9369 |

**ตารางที่ 4.8** ใช้ File Sweetest 2.wav เป็นต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9254 |
| Sweetest 2 .wav | 1.0000 |
| Sweetest 3 .wav | 0.9658 |
| Sweetest 4 .wav | 0.9415 |
| Sweetest 5 .wav | 0.9544 |
| Sweetest 6.wav | 0.9863 |
| Sweetest 7.wav | 0.9935 |

**ตารางที่ 4.9** ใช้ File Sweetest 3.wav เป็นต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9469 |
| Sweetest 2 .wav | 0.9658 |
| Sweetest 3 .wav | 1.0000 |
| Sweetest 4 .wav | 0.9751 |
| Sweetest 5 .wav | 0.9924 |
| Sweetest 6.wav | 0.9800 |
| Sweetest 7.wav | 0.9778 |

**ตารางที่ 4.10** ใช้ File Sweetest 4.wav เป็น ต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9859 |
| Sweetest 2 .wav | 0.9415 |
| Sweetest 3 .wav | 0.9751 |
| Sweetest 4 .wav | 1.0000 |
| Sweetest 5 .wav | 0.9767 |
| Sweetest 6.wav | 0.9571 |
| Sweetest 7.wav | 0.9522 |

**ตารางที่ 4.11** ใช้ File Sweetest 5.wav เป็น ต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9471 |
| Sweetest 2 .wav | 0.9544 |
| Sweetest 3 .wav | 0.9924 |
| Sweetest 4 .wav | 0.9767 |
| Sweetest 5 .wav | 1.0000 |
| Sweetest 6.wav | 0.9679 |
| Sweetest 7.wav | 0.9638 |

**ตารางที่ 4.12** ใช้ File Sweetest 6.wav เป็น ต้นแบบ ของสับปะรดผลหวาน

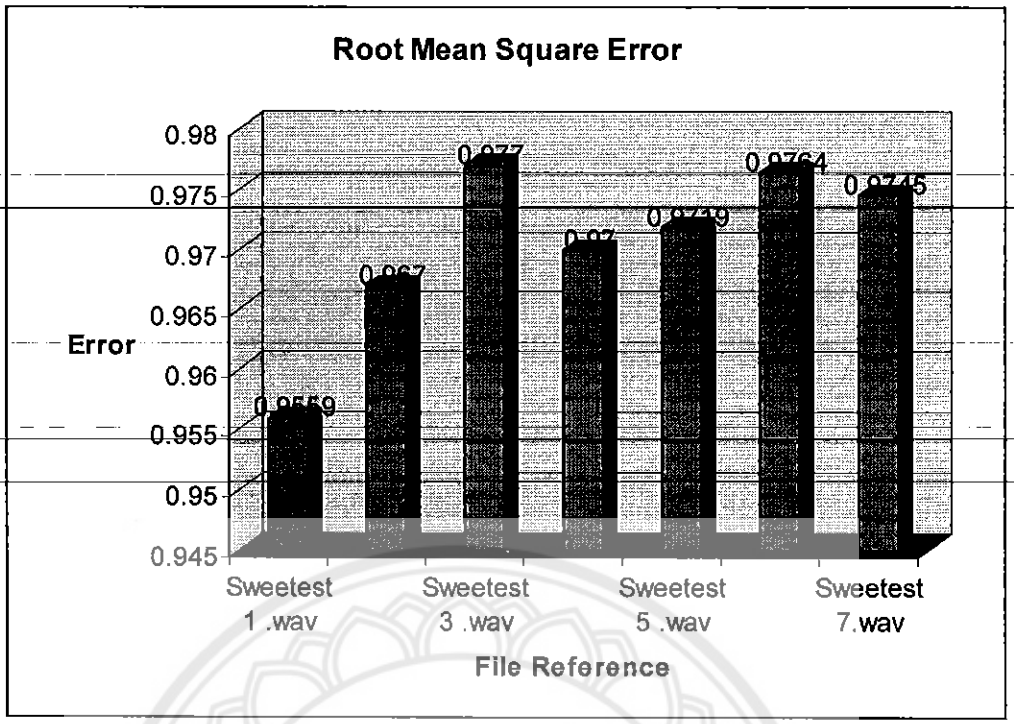
| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9469 |
| Sweetest 2 .wav | 0.9863 |
| Sweetest 3 .wav | 0.9800 |
| Sweetest 4 .wav | 0.9571 |
| Sweetest 5 .wav | 0.9679 |
| Sweetest 6.wav | 1.0000 |
| Sweetest 7.wav | 0.9954 |

**ตารางที่ 4.13** ใช้ File Sweetest 7.wav เป็น ต้นแบบ ของสับปะรดผลหวาน

| File Name | Max Correlation Value |
|---|---|
| Sweetest 1 .wav | 0.9369 |
| Sweetest 2 .wav | 0.9935 |
| Sweetest 3 .wav | 0.9778 |
| Sweetest 4 .wav | 0.9522 |
| Sweetest 5 .wav | 0.9638 |
| Sweetest 6.wav | 0.9954 |
| Sweetest 7.wav | 1.0000 |

**ตารางที่ 4.14** ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็นต้นแบบของผลหวาน

| File Name | Root Mean Square Error |
|---|---|
| Sweetest 1 .wav | 0.9559 |
| Sweetest 2 .wav | 0.9670 |
| Sweetest 3 .wav | 0.9770 |
| Sweetest 4 .wav | 0.9700 |
| Sweetest 5 .wav | 0.9719 |
| Sweetest 6.wav | 0.9764 |
| Sweetest 7.wav | 0.9745 |




**รูปที่ 4.13** กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบของผลหวาน

สรุปการทดสอบ จากการทดสอบจะเห็นได้ว่าในการทดสอบนั้นเมื่อใช้ File Sweetest 1.wav เป็นต้นแบบแล้ว ค่า Root Mean Square Error ที่ได้มีค่าต่ำกว่าต้นแบบตัวอื่นๆ ซึ่งจะเป็นผลที่น่าเชื่อถือมากที่สุด ดังนั้นจึงเลือกใช้ File Sweetest 1.wav เป็นต้นแบบของสับปะรดผลสุก

### 4.6.2 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดเปรี้ยว

**ตารางที่ 4.15** ใช้ File Sour 1.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 1.0000 |
| Sour 2 .wav | 0.9029 |
| Sour 3 .wav | 0.8724 |
| Sour 4 .wav | 0.8865 |
| Sour 5 .wav | 0.9633 |
| Sour 6.wav | 0.9324 |
| Sour 7.wav | 0.9468 |

**ตารางที่ 4.16** ใช้ File Sour 2.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.9029 |
| Sour 2 .wav | 1.0000 |
| Sour 3 .wav | 0.8583 |
| Sour 4 .wav | 0.9812 |
| Sour 5 .wav | 0.9275 |
| Sour 6.wav | 0.9161 |
| Sour 7.wav | 0.9226 |

**ตารางที่ 4.17** ใช้ File Sour 3.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.8724 |
| Sour 2 .wav | 0.8583 |
| Sour 3 .wav | 1.0000 |
| Sour 4 .wav | 0.8645 |
| Sour 5 .wav | 0.9519 |
| Sour 6.wav | 0.9708 |
| Sour 7.wav | 0.8875 |

**ตารางที่ 4.18** ใช้ File Sour 4.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.8865 |
| Sour 2 .wav | 0.9812 |
| Sour 3 .wav | 0.8645 |
| Sour 4 .wav | 1.0000 |
| Sour 5 .wav | 0.9322 |
| Sour 6.wav | 0.9255 |
| Sour 7.wav | 0.9414 |

**ตารางที่ 4.19** ใช้ File Sour 5.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.9633 |
| Sour 2 .wav | 0.9275 |
| Sour 3 .wav | 0.9519 |
| Sour 4 .wav | 0.9322 |
| Sour 5 .wav | 1.0000 |
| Sour 6.wav | 0.9879 |
| Sour 7.wav | 0.9687 |

**ตารางที่ 4.20** ใช้ File Sour 6.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

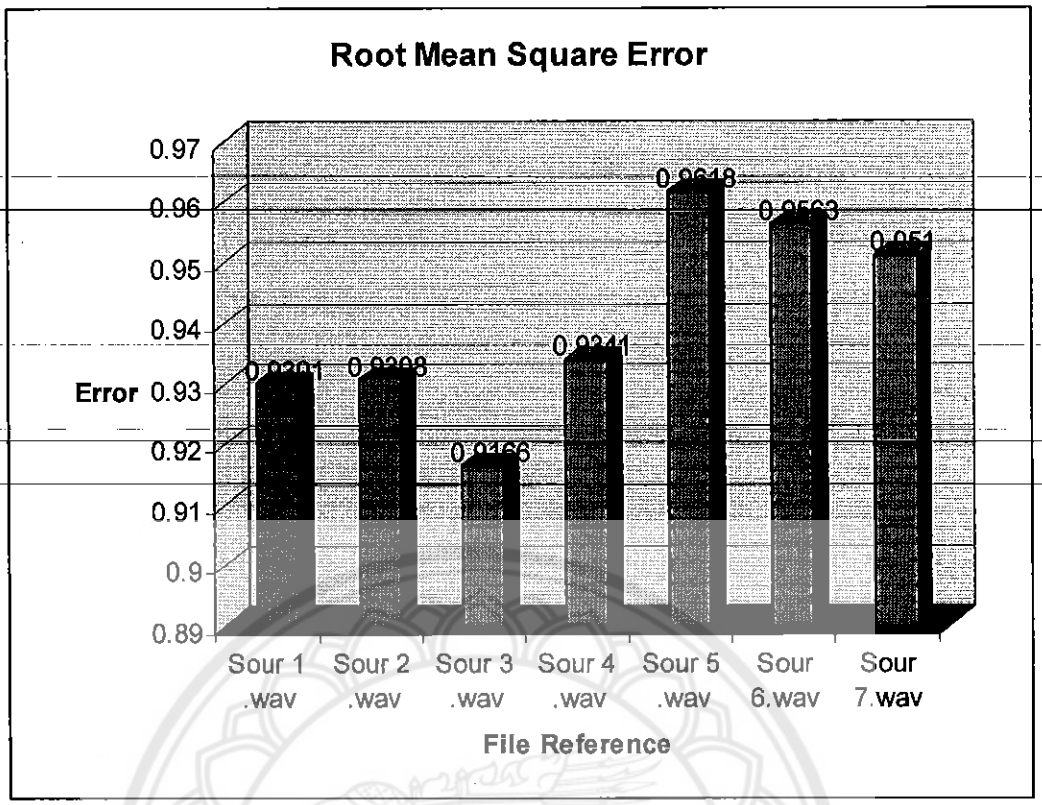
| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.9324 |
| Sour 2 .wav | 0.9161 |
| Sour 3 .wav | 0.9708 |
| Sour 4 .wav | 0.9255 |
| Sour 5 .wav | 0.9879 |
| Sour 6.wav | 1.000 |
| Sour 7.wav | 0.9583 |

**ตารางที่ 4.21** ใช้ File Sour 7.wav เป็น ต้นแบบ ของสับปะรดผลเปรี้ยว

| File Name | Max Correlation Value |
|---|---|
| Sour 1 .wav | 0.9468 |
| Sour 2 .wav | 0.9226 |
| Sour 3 .wav | 0.8875 |
| Sour 4 .wav | 0.9414 |
| Sour 5 .wav | 0.9687 |
| Sour 6.wav | 0.9583 |
| Sour 7.wav | 1.0000 |

**ตารางที่ 4.22** ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบ ของผลเปรี้ยว

| File Name | Root Mean Square Error |
|---|---|
| Sour 1 .wav | 0.9301 |
| Sour 2 .wav | 0.9308 |
| Sour 3 .wav | 0.9166 |
| Sour 4 .wav | 0.9341 |
| Sour 5 .wav | 0.9618 |
| Sour 6.wav | 0.9563 |
| Sour 7.wav | 0.9510 |




**รูปที่ 4.14** กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบของผลเปรี้ยว

สรุปการทดสอบ จากการทดสอบจะเห็นได้ว่าในการทดสอบนั้นเมื่อใช้ File Sour 3 .wav เป็น ต้นแบบ แล้วค่า Root Mean Square Error ที่ได้มีค่าต่ำกว่า ต้นแบบ ตัวอื่นๆ ซึ่งจะเป็นผลที่น่าเชื่อถือมากที่สุด ดังนั้นจึงเลือกใช้ File Sour 3 .wav เป็นต้นแบบของสับปะรดผลเปรี้ยว

### 4.6.3 การทดสอบเพื่อหาสัญญาณต้นแบบสับปะรดเน่า

**ตารางที่ 4.23** ใช้ File Exp 1.wav เป็น ต้นแบบ ของสับปะรดผลเน่า

| File Name | Max Correlation Value |
|---|---|
| Exp 1 .wav | 1.0000 |
| Exp 2 .wav | 0.8822 |
| Exp 3 .wav | 0.9867 |
| Exp 4 .wav | 0.9632 |
| Exp 5 .wav | 0.9393 |

**ตารางที่ 4.24** ใช้ File Exp 2.wav เป็น ต้นแบบ ของสับปะรดผลเน่า

| File Name | Max Correlation Value |
|---|---|
| Exp 1 .wav | 0.8822 |
| Exp 2 .wav | 1.0000 |
| Exp 3 .wav | 0.8421 |
| Exp 4 .wav | 0.8801 |
| Exp 5 .wav | 0.7031 |

**ตารางที่ 4.25** ใช้ File Exp 3.wav เป็น ต้นแบบ ของสับปะรดผลเน่า

| File Name | Max Correlation Value |
|---|---|
| Exp 1 .wav | 0.9867 |
| Exp 2 .wav | 0.9254 |
| Exp 3 .wav | 1.0000 |
| Exp 4 .wav | 0.9572 |
| Exp 5 .wav | 0.9608 |

**ตารางที่ 4.26** ใช้ File Exp 4.wav เป็น ต้นแบบ ของสับปะรดผลเน่า

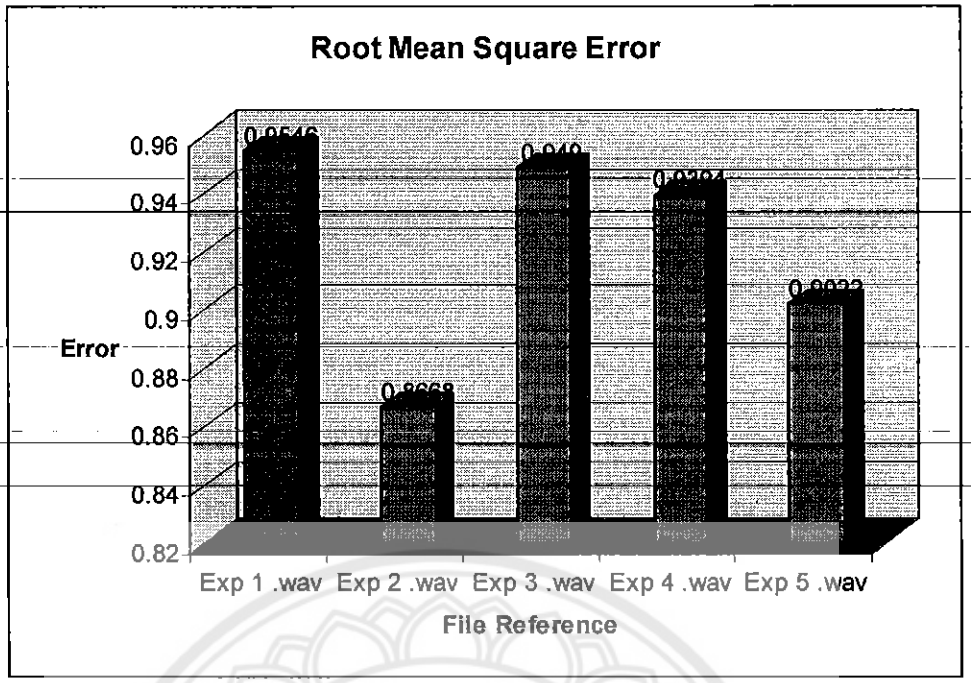
| File Name | Max Correlation Value |
|---|---|
| Exp 1 .wav | 0.9632 |
| Exp 2 .wav | 0.8801 |
| Exp 3 .wav | 0.9572 |
| Exp 4 .wav | 1.0000 |
| Exp 5 .wav | 0.8911 |

**ตารางที่ 4.27** ใช้ File Exp 5.wav เป็น ต้นแบบ ของสับปะรดผลเน่า

| File Name | Max Correlation Value |
|---|---|
| Exp 1 .wav | 0.9363 |
| Exp 2 .wav | 0.7031 |
| Exp 3 .wav | 0.9608 |
| Exp 4 .wav | 0.8911 |
| Exp 5 .wav | 1.0000 |

**ตารางที่ 4.28** ค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบ ของผลเน่า

| File Name | Root Mean Square Error |
|---|---|
| Exp 1 .wav | 0.9546 |
| Exp 2 .wav | 0.8668 |
| Exp 3 .wav | 0.9490 |
| Exp 4 .wav | 0.9394 |
| Exp 5 .wav | 0.9022 |




**รูปที่ 4.15** กราฟแสดงการเปรียบเทียบค่า Root Mean Square Error เมื่อใช้แต่ละตัวเป็น ต้นแบบของผลน่า

สรุปการทดสอบ จากการทดสอบจะเห็นได้ว่าในการทดสอบนั้นเมื่อใช้ File Exp 2 .wav เป็น ต้นแบบแล้ว ค่า Root Mean Square Error ที่ได้มีค่าต่ำกว่าต้นแบบตัวอื่นๆ ซึ่งจะเป็นผลที่น่าเชื่อถือมากที่สุด ดังนั้นจึงเลือกใช้ File Exp 2 .wav เป็นต้นแบบของสับปะรดผลน่า

**4.6.4 การทดลองเพื่อหาหลักเกณฑ์ในการวิเคราะห์เสียง**

4.6.4.1 นำผลสับปะรดหวานมาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่าเพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี Cross correlation ได้ผลดังตารางที่ 4.29

**ตารางที่ 4.29** ข้อมูลที่ได้จากการนำสับปะรดหวานมาเทียบกับสับปะรดต้นแบบ

| File | ลักษณะภายใน | Sweetest 1.wav | Sour 3 .wav | Exp 2 .wav |
|---|---|---|---|---|
| Sweetest 1 .wav | สุก | 1.0000 | 0.9493 | 0.8988 |
| Sweetest 2 .wav | สุก | 0.9254 | 0.7994 | 0.7154 |
| Sweetest 3 .wav | สุก | 0.9469 | 0.8664 | 0.7477 |
| Sweetest 4 .wav | สุก | 0.9859 | 0.9283 | 0.8481 |
| Sweetest 5 .wav | สุก | 0.9471 | 0.8747 | 0.7391 |
| Sweetest 6.wav | สุก | 0.9469 | 0.8519 | 0.7508 |
| Sweetest 7.wav | สุก | 0.9369 | 0.8257 | 0.7306 |

4.6.4.2 นำผลสับปะรดเปรี้ยวมาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่าเพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี Cross correlation ได้ผลดังตารางที่ 4.30

**ตารางที่ 4.30** ข้อมูลที่ได้จากการนำสับปะรดเปรี้ยวมาเทียบกับสับปะรดต้นแบบ

| File | ลักษณะภายใน | Sweetest 1.wav | Sour 3 .wav | Exp 2 .wav |
|---|---|---|---|---|
| Sour 1 .wav | เปรี้ยว | 0.9199 | 0.8724 | 0.7052 |
| Sour 2 .wav | เปรี้ยว | 0.9432 | 0.8583 | 0.7789 |
| Sour 3 .wav | เปรี้ยว | 0.9493 | 1.0000 | 0.8207 |
| Sour 4 .wav | เปรี้ยว | 0.9504 | 0.8645 | 0.8024 |
| Sour 5 .wav | เปรี้ยว | 0.9754 | 0.9519 | 0.7412 |
| Sour 6.wav | เปรี้ยว | 0.9846 | 0.8708 | 0.7907 |
| Sour 7.wav | เปรี้ยว | 0.9845 | 0.8875 | 0.7566 |

4.6.4.3 นำผลสับปะรดเน่ามาทดสอบกับสับปะรดต้นแบบที่มีลักษณะหวาน เปรี้ยวและเน่าเพื่อหาค่าที่ได้จากการวิเคราะห์ด้วยวิธี Cross correlation ได้ผลดังตารางที่ 4.31

**ตารางที่ 4.31** ข้อมูลที่ได้จากการนำสับปะรดเน่ามาเทียบกับสับปะรดต้นแบบ

| File | ลักษณะภายใน | Sweetest 1.wav | Sour 3 .wav | Exp 2 .wav |
|---|---|---|---|---|
| Exp 1 .wav | เน่า | 0.9921 | 0.8399 | 0.7822 |
| Exp 2 .wav | เน่า | 0.8988 | 0.9207 | 1.0000 |
| Exp 3 .wav | เน่า | 0.9774 | 0.9197 | 0.8421 |
| Exp 4 .wav | เน่า | 0.9663 | 0.8896 | 0.7801 |
| Exp 5 .wav | เน่า | 0.9210 | 0.8655 | 0.7031 |

**4.6.5 หลักเกณฑ์ที่ใช้ในการวิเคราะห์**

4.6.5.1 จากการทดลองเมื่อหาต้นแบบของสับปะรดผลหวานได้แล้ว ให้ผลดังนี้

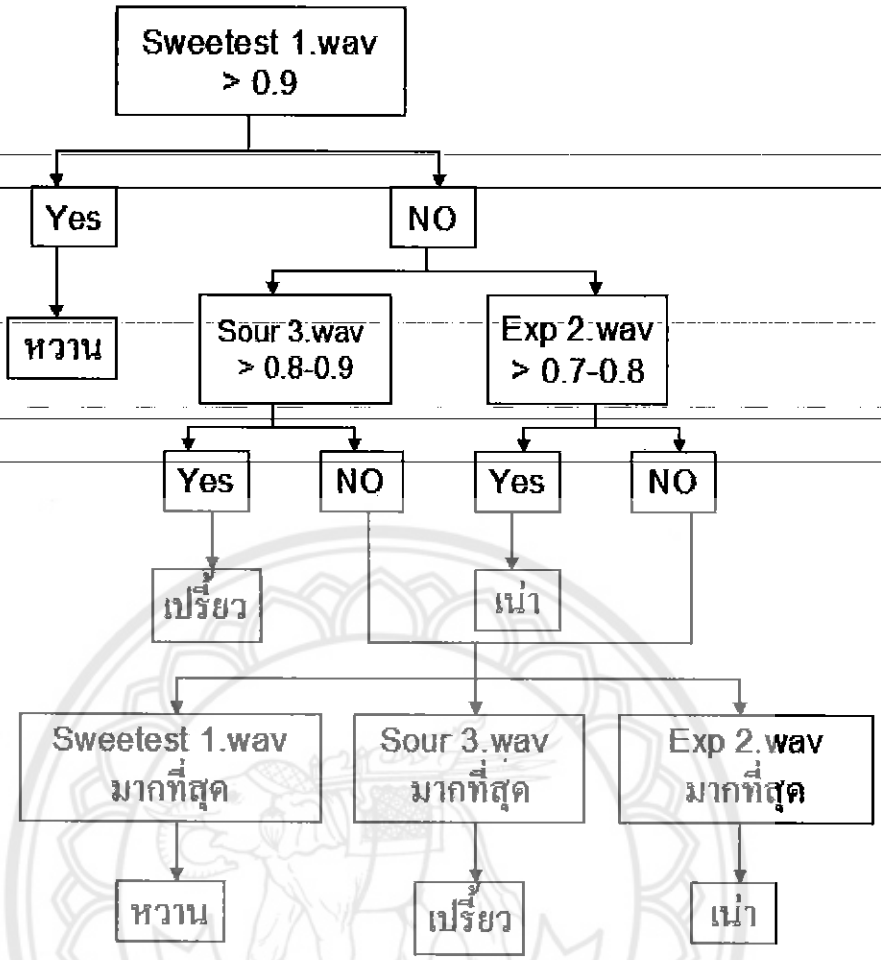
- เมื่อใช้สับปะรดผลหวาน Sweetest 1.wav เป็น ต้นแบบ และเทียบกับผลสุก
  ค่าที่ทำ xcorr ของ FFT จะมีค่ามากกว่า 0.9 ขึ้นไป
- เมื่อใช้สับปะรดผลหวาน Sweetest 1.wav เป็น ต้นแบบ และเทียบกับผลเปรี้ยว
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าต่ำกว่า 0.9 - 1.3
- เมื่อใช้สับปะรดผลหวาน Sweetest 1.wav เป็น ต้นแบบ และเทียบกับผลเน่า
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าต่ำกว่า 0.9

4.6.5.2 จากการทดลองเมื่อหาต้นแบบของสับปะรดผลเปรี้ยวได้แล้ว ให้ผลดังนี้

- เมื่อใช้สับปะรดผลเปรี้ยว Sour 3 .wav เป็น ต้นแบบ และเทียบกับผลสุก
  - ค่าที่ทำ xcorr ของ FFT จะมีค่ามากกว่า 0.9 ขึ้นไป
- เมื่อใช้สับปะรดผลเปรี้ยว Sour 3 .wav เป็น ต้นแบบ และเทียบกับผลเปรี้ยว
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าอยู่ในช่วง 0.8 - 0.9
- เมื่อใช้สับปะรดผลเปรี้ยว Sour 3 .wav เป็น ต้นแบบ และเทียบกับผลเน่า
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าอยู่ในช่วง 0.7 - 0.8

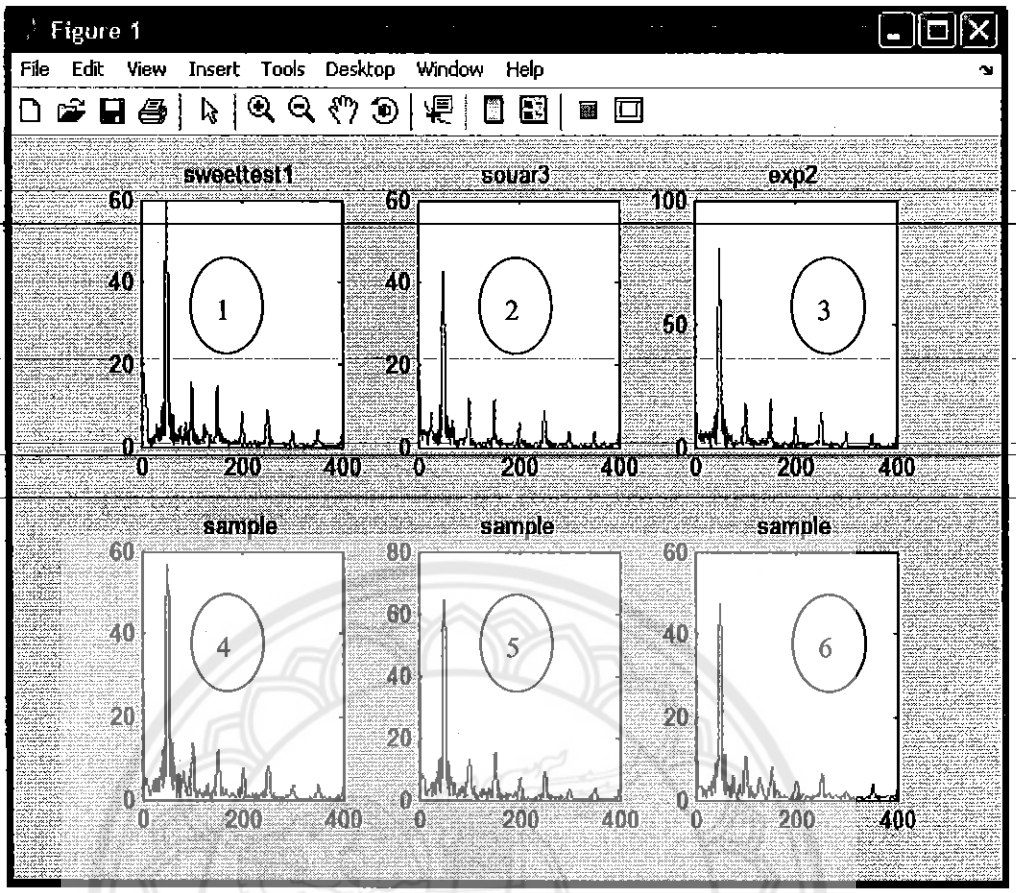
4.6.5.3 จากการทดลองเมื่อหาต้นแบบของสับปะรดผลเน่าได้แล้ว ให้ผลดังนี้

- เมื่อใช้สับปะรดผลเน่า Exp 2 .wav เป็น ต้นแบบ และเทียบกับผลสุก
  - ค่าที่ทำ xcorr ของ FFT จะมีค่ามากกว่า 0.9 ขึ้นไป
- เมื่อใช้สับปะรดผลเน่า Exp 2 .wav เป็น ต้นแบบ และเทียบกับผลเปรี้ยว
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าอยู่ในช่วง 0.8 - 0.9
- เมื่อใช้สับปะรดผลเน่า Exp 2 .wav เป็น ต้นแบบ และเทียบกับผลเน่า
  - ค่าที่ทำ xcorr ของ FFT จะมีค่าอยู่ในช่วง 0.7 - 0.8




**รูปที่ 4.16** แผนผังแสดงเกณฑ์ที่ใช้ในการแบ่งแยกสับปะรดแบบต่างๆ

**หมายเหตุ** สับปะรดผลเน่าที่เน่าเฉพาะส่วนไม่สามารถแยกแยะออกจากกลุ่มอื่นๆ ได้




**รูปที่** 4.17 ภาพของสัญญาณต้นแบบและสัญญาณเสียงที่นำมาทดสอบโดยการใช้โปรแกรม

1 คือสัญญาณของ Sweetest1.wav ซึ่งเป็นต้นแบบของผลหวาน แสดงให้ดูในช่วงที่มีผลมาก

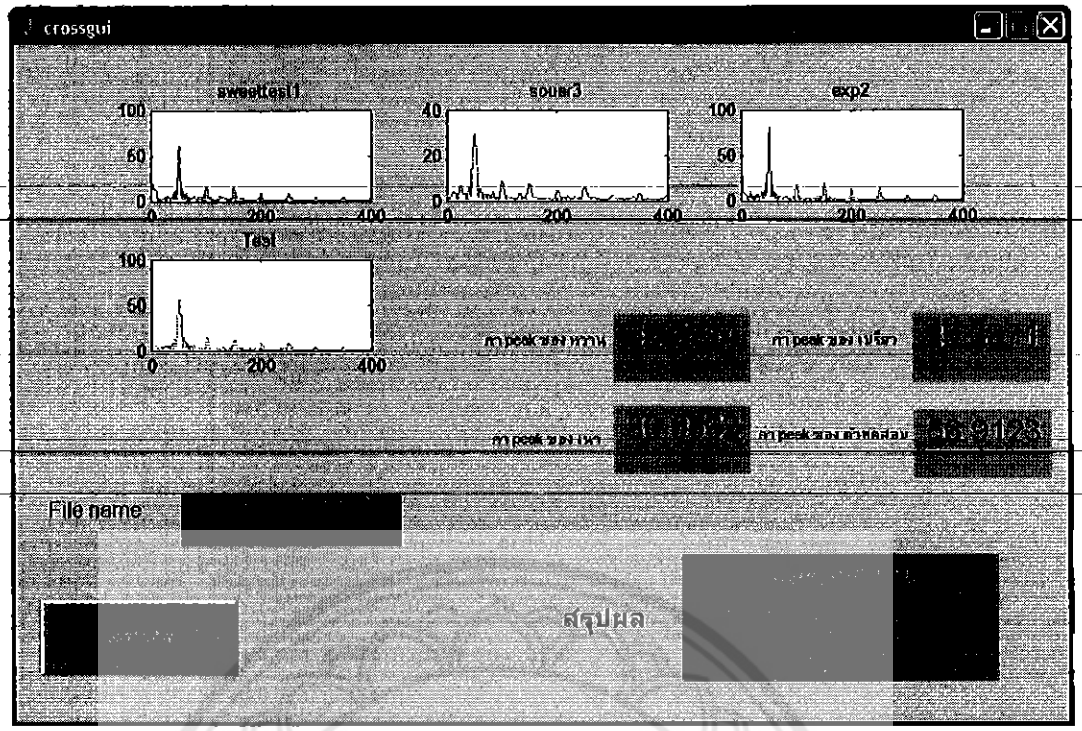
2 คือสัญญาณของ Sour3.wav ซึ่งเป็นต้นแบบของผลเปรี้ยว แสดงให้ดูในช่วงที่มีผลมาก

3 คือสัญญาณของ exp3.wav ซึ่งเป็นต้นแบบของผลเน่า แสดงให้ดูในช่วงที่มีผลมาก

4 คือสัญญาณของผลที่ต้องการนำมาทดสอบในการเคาะครั้งที่ 1 แสดงให้ดูในช่วงที่มีผลมาก

5 คือสัญญาณของผลที่ต้องการนำมาทดสอบในการเคาะครั้งที่ 2 แสดงให้ดูในช่วงที่มีผลมาก

6 คือสัญญาณของผลที่ต้องการนำมาทดสอบในการเคาะครั้งที่ 3 แสดงให้ดูในช่วงที่มีผลมาก




**รูปที่ 4.18** ภาพของสัญญาณต้นแบบและสัญญาณเสียงที่นำมาทดสอบโดยการใช้ Cross-correlation ทดสอบสับปะรดต้นแบบกับตัวอย่างสับปะรดชุดที่ 1

**ตารางที่ 4.32** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 1

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| Sweetest 1 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 2 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 3 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 4 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 5 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 6 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 7 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sour 1 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 2 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 4 | เปรี้ยว | หวาน | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 5 | เปรี้ยว | เปรี้ยว | หวาน | หวาน | หวาน | ผิด |
| Sour 6 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 7 | เปรี้ยว | หวาน | เปรี้ยว | หวาน | หวาน | ผิด |
| Exp 1 | เน่า | เน่า | หวาน | เน่า | เน่า | ถูก |
| Exp 2 | เน่า | หวาน | เน่า | เน่า | เน่า | ถูก |
| Exp 3 | เน่า | เน่า | เน่า | หวาน | เน่า | ถูก |
| Exp 4 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 5 | เน่า | หวาน | เน่า | เน่า | เน่า | ถูก |

**ตารางที่ 4.33** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 2

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| 14_Aug 1 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |
| 14_Aug 2 | หวาน | เปรี้ยว | เปรี้ยว | หวาน | เปรี้ยว | ผิด |
| 14_Aug 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 14_Aug 4 | เปรี้ยว | เปรี้ยว | หวาน | หวาน | หวาน | ผิด |
| 14_Aug 5 | หวาน | หวาน | หวาน | หวาน | หวาน | ถูก |

**ตารางที่ 4.34** ข้อมูลที่ได้จากการทดลองกับตัวอย่างสับปะรดชุดที่ 3

| File | ลักษณะสับปะรด | เคาะครั้งที่ 1 | เคาะครั้งที่ 2 | เคาะครั้งที่ 3 | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|---|
| 6_Sept 1 | เปรี้ยว | หวาน | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 6_Sept 2 | หวาน | เปรี้ยว | หวาน | หวาน | หวาน | ถูก |
| 6_Sept 3 | เน่า | เปรี้ยว | เน่า | เปรี้ยว | เปรี้ยว | ผิด |
| 6_Sept 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | หวาน | เปรี้ยว | ถูก |
| 6_Sept 5 | หวาน | หวาน | หวาน | เน่า | หวาน | ถูก |
| 6_Sept 6 | หวาน | หวาน | เน่า | หวาน | หวาน | ถูก |
| 6_Sept 7 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 6_Sept 8 | เน่า | เน่า | เน่า | เน่า | เน่า | ถูก |

## 4.7 ผลการเปรียบเทียบโดยใช้วิธีแปลงฟูเรียร์แบบเร็วกับวิธีสหสัมพันธ์ข้าม

**ตารางที่ 4.35** การนำสัญญาณเสียงที่ทราบลักษณะมาเปรียบเทียบกัน

| File | ลักษณะสัปปะรด | FFT function | CCF function | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|
| Sweetest 1 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 2 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 3 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 4 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 5 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 6 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sweetest 7 | หวาน | หวาน | หวาน | หวาน | ถูก |
| Sour 1 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 2 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 5 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 6 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Sour 7 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| Exp 1 | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 2 | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 3 | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 4 | เน่า | เน่า | เน่า | เน่า | ถูก |
| Exp 5 | เน่า | หวาน | เน่า | ข้อมูลไม่ตรงกัน | ผิด |

**ตารางที่ 4.36** การนำสัญญาณเสียงมาเปรียบเทียบกันโดยทำการสุ่มเลือกผลสับปะรด

| File | ลักษณะสับปะรด | FFT function | CCF function | สรุปผลที่ได้ | ความถูกต้อง |
|---|---|---|---|---|---|
| 14_Aug 1 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 14_Aug 2 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 14_Aug 3 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 14_Aug 4 | เปรี้ยว | เปรี้ยว | หวาน | ข้อมูลไม่ตรงกัน | ผิด |
| 14_Aug 5 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 6_Sept 1 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 6_Sept 2 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 6_Sept 3 | เน่า | เปรี้ยว | เปรี้ยว | ข้อมูลไม่ตรงกัน | ผิด |
| 6_Sept 4 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 6_Sept 5 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 6_Sept 6 | หวาน | เน่า | หวาน | ข้อมูลไม่ตรงกัน | ผิด |
| 6_Sept 7 | เปรี้ยว | หวาน | เปรี้ยว | ข้อมูลไม่ตรงกัน | ผิด |
| 6_Sept 8 | เน่า | เน่า | เน่า | เน่า | ถูก |
| 21_Sept 1 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 21_Sept 2 | เปรี้ยว | หวาน | เปรี้ยว | ข้อมูลไม่ตรงกัน | ผิด |
| 21_Sept 3 | หวาน | หวาน | หวาน | หวาน | ถูก |
| 21_Sept 4 | หวาน | หวาน | เน่า | ข้อมูลไม่ตรงกัน | ผิด |
| 21_Sept 5 | เปรี้ยว | เปรี้ยว | เปรี้ยว | เปรี้ยว | ถูก |
| 21_Sept 6 | หวาน | หวาน | หวาน | หวาน | ถูก |

จำนวนผลสับปะรดที่ทำการสุ่ม มีทั้งหมด 3 ลูก จากการสรุปจากตารางที่ 4.36 จะเห็น ได้ว่า มีบางเสียงของผลสับปะรดที่ FFT function สามารถวิเคราะห์ได้ตรงตามที่ต้องการ แต่ CCF function ไม่สามารถวิเคราะห์บางสัญญาณไม่ถูกต้อง เนื่องจากการคำนวณหาค่าของแต่ละ function มันแตกต่างกัน และสัญญาณเสียงที่วิเคราะห์ออกมาถูกต้องเพราะมีความถี่ใกล้เคียงกัน

# บทที่ 5
# สรุปและข้อเสนอแนะ

## 5.1 สรุปผลโครงงาน

จากการทำการศึกษาเรื่องการคัดแยกผลสับปะรดด้วยวิธีวิเคราะห์เสียงนั้น ในการทดลองเราจะศึกษาสัญญาณเสียงในโดเมนความถี่ (Frequency Domain) ซึ่งเราพบว่า การศึกษาสัญญาณในโดเมนความถี่นั้นเราสามารถแยกแยะสัญญาณได้ดีกว่าการวิเคราะห์ในโดเมนเวลา (Time Domain) และในการวิเคราะห์สัญญาณเสียงนั้น เราได้ใช้กระบวนการการเปรียบเทียบสัญญาณด้วย การแปลงฟูเรียร์แบบเร็ว (Fast Fourier Transform) เข้ามาเป็นวิธีวิเคราะห์เพื่อใช้ในการคัดแยกสับปะรด ซึ่งการเปรียบเทียบจะให้ค่าคำตอบออกมาเพื่อใช้ในการคัดแยกสับปะรดต่อไป จากการทดลองได้ข้อสรุปดังนี้

5.1.1 ค่าความถี่ของสัญญาณเสียงของสับปะรดผลดิบมีค่าสูงที่สุด โดยความถี่เสียงของสับปะรดผลสุกมีค่าต่ำลงมาและความถี่เสียงของสับปะรดผลเน่ามีความถี่เสียงที่ต่ำที่สุด

5.1.2 สับปะรดที่มีรูปร่างกลมรี พันธุ์และขนาดที่แตกต่างกันมากเมื่อเทียบกับสับปะรดต้นแบบ จะไม่สามารถวิเคราะห์ได้ถูกต้อง

5.1.3 สับปะรดที่มีรูปร่างแคระแกรน บิดเบี้ยว ไม่สมบูรณ์ สัญญาณเสียงที่วิเคราะห์ได้ในโดเมนความถี่ จะมีลักษณะที่แตกต่างจากสัญญาณต้นแบบ ทำให้ไม่สามารถวิเคราะห์ได้ถูกต้อง

5.1.4 สับปะรดที่มีลักษณะเป็นผลดิบแท้ คือ เนื้อภายในสับปะรดเป็นสีขาวล้วนเหมือนแน่น และแข็ง ลักษณะของสัญญาณที่ได้จะเหมือนกับสัญญาณของสับปะรดผลสุก ผลการวิเคราะห์จึงเกิดความผิดพลาดขึ้น

5.1.5 สับปะรดที่มีลักษณะเน่าบางส่วนจะมีผลต่อสัญญาณเสียง โดยสัญญาณเสียงที่ได้จะประกอบไปด้วยลักษณะของส่วนที่ดีและส่วนที่เน่า ซึ่งทำให้ไม่สามารถแยกแยะด้วยการวิเคราะห์เสียงได้

## 5.2 ข้อเสนอแนะ

จากการศึกษาพบว่า ยังมีปัจจัยอื่นอีกหลายอย่างที่มีผลต่อการวิเคราะห์เสียง และตัวแปรที่เราทำการศึกษาเป็นตัวแปรที่ยากต่อการควบคุม ตัวแปรมีลักษณะที่อ่อนไหวซึ่งทำให้ยากต่อการวิเคราะห์และการหาสับปะรดต้นแบบของสับปะรดผลดิบ ผลสุกและผลเน่านั้นยังไม่มีต้นแบบที่แน่นอน ซึ่งในการหาต้นแบบนั้นจะอาศัยจากข้อมูลในเชิงสถิติเข้ามาช่วย ซึ่งทำให้ยังมีข้อจำกัดที่สับปะรดต้นแบบที่ใช้วิเคราะห์นั้นจะใช้วิเคราะห์เฉพาะพันธุ์ที่เรามีต้นแบบและข้อมูลเพียงพอแล้วเท่านั้น การที่จะใช้การวิเคราะห์เสียงให้ได้ผลที่แน่นอนนั้น จำเป็นที่จะต้องศึกษาลงลึกถึงลักษณะทางพันธุกรรมของสับปะรดด้วย การคัดแยกสับปะรดนั้นนอกจากการวิเคราะห์เสียงแล้วยังควรใช้วิธีอื่นเข้ามาช่วยในการวิเคราะห์ด้วยเพื่อให้ได้ผลที่ถูกต้อง

# เอกสารอ้างอิง


1. รศ.ดร.มนัส สังวรศิลป์, วรรัตน์ ภัทรอมรกุล, 2543. คู่มือโปรแกรม MATLAB ฉบับสมบูรณ์. กรุงเทพฯ: สำนักพิมพ์ดวงกมล
2. รณฤทธิ์ ฤทธิรณ, อนุพันธ์ เทอดวงศ์วรกุล, 2543. การศึกษาการใช้คุณสมบัติของเสียงเพื่อตรวจสอบและ คัดแยกอาหารกระป๋อง. นิสิตปริญญาโทภาควิชาวิศวกรรมเคมี, ภาควิชาวิศวกรรมการอาหาร คณะ วิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น.
3. A.Terdwongworakul, 2542. Investigation of change of exponential frequency index as related to durian maturity. Department of Food Engineering, Faculty of Engineering, Kasetsart University, Kampangsan Campus.
4. The MathWork, 1999. Signal Processing Toolbox User's Guide. The MathWork Inc.
5. CHI-TONG CHEN, 1994.SYSTEM AND SIGNAL ANALYSIS. Sunders College Publishing.
6. http://ocrma.stanford.edu/~jos/mdft/Discrete_Time_fourier_Tranform.html.

# ภาคผนวก ก.

**ตัวอย่าง ต้องการเปรียบเทียบสัญญาณ File Sweetest1.wav กับสัญญาณ File Sweetest2.wav จะมีขั้นตอนการป้อนคำสั่งดังนี้**

| | |
|---|---|
| >> y = wavread('Sweetest1.wav'); | : คำสั่งเพื่อทำการอ่านข้อมูลที่บันทึกไว้ |
| >> plot(y) | : คำสั่งเพื่อเรียกดูภาพของสัญญาณเสียงที่อ่านค่าแล้ว |
| >> y1 = y(1900:2300); | : คำสั่งเพื่อกำหนดตำแหน่งเริ่มต้นและสิ้นสุดของสัญญาณ |
| >> plot(y1) | : คำสั่งเพื่อเรียกดูภาพสัญญาณที่ Sampling แล้ว |
| >> y2=fft(y1); | : คำสั่งเพื่อเปรียบเทียบสัญญาณให้อยู่ใน frequency domain โดยการทำ FFT |
| >> z = wavread('Sweetest2.wav'); | : คำสั่งเพื่อทำการอ่านข้อมูลที่บันทึกไว้ |
| >> plot(z) | : คำสั่งเพื่อเรียกดูภาพของสัญญาณเสียงที่อ่านค่าแล้ว |
| >> z1 = z(1900:2300); | : คำสั่งเพื่อกำหนดตำแหน่งเริ่มต้นและสิ้นสุดของสัญญาณ |
| >> plot(z1) | : คำสั่งเพื่อเรียกดูภาพสัญญาณที่ Sampling แล้ว |
| >> z2=fft(z1); | : คำสั่งเพื่อเปรียบเทียบสัญญาณให้อยู่ใน frequency domain โดยการทำ FFT |
| >> c = xcorr(abs(y2) , abs(z2) , 'coeff '); | : คำสั่งเพื่อเปรียบเทียบสัญญาณด้วยวิธี cross correlation |
| >> max ( c ) | : หาค่าสูงสุดจากการเปรียบเทียบสัญญาณด้วยวิธี cross correlation |
| >>0.9254 | : ค่าสูงสุดที่ได้จากการเปรียบเทียบ |

**ตัวอย่าง ได้มีสร้างการขึ้น Function ขึ้นมา หนึ่ง function โดยชื่อว่า soundSIG**

| | |
|---|---|
| function [p]=soundSig(filename) | [p] เป็นตัว output ของ function เท่า soundsig ชื่อของ function และส่วน filename คือ input |
| in=wavread(filename); | อ่านค่าของ input เก็บไว้ในตัวแปรชื่อ in |

```
f=fft(in,100000);                    fft คือ ฟูเลียร์ทรานฟอร์ม ตัวเลข คือ ความยาว
                                     ของ vecter in
q=f.*conj(f);                        เพื่อหาค่า จำนวนจริง
q=abs(q(1:5000));                    เป็น แอบโสรูด โดย ค่า 1-5000 เก็บในตัวแปร q
for i=1:50                           เป็นส่วนในการวนลูปรวมค่า
 t=(i-1)*100+1;
 p(i)=sum(q(t:t+99));
end
p=p/sum(p);                          p คือค่าเฉลี่ยของตัวมันเอง
```

**Source code การสร้างหน้าต่างการแสดงผลของโปรแกรม (GUI)**

```
function varargout = newgui(varargin)
%*************************************************************************
%* Program to pick and chooses fruit
%* Make By CHATTA TOSUANCHIT  #CPE NARESUAN UNIVERSITY
% * NEWGUI M-file for newgui.fig
% * Copyright 2002-2003 The MathWorks, Inc.
% * Edit the above text to modify the response to help newgui
% * Last Modified by GUIDE v2.5 04-Sep-2007 23:40:03
% * Begin initialization code - DO NOT EDIT
%*
%*************************************************************************
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @newgui_OpeningFcn, ...
                   'gui_OutputFcn',  @newgui_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before newgui is made visible.
function newgui_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to newgui (see VARARGIN)

% Choose default command line output for newgui
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);


% UIWAIT makes newgui wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = newgui_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```
% Get default command line output from handles structure
varargout{1} = handles.output;
set(handles.axes4);
[p1]=soundSig('sweettest4');
[p2]=soundSig('souar5');
[p3]=soundSig('exp1');

axis([0 30 0 0.04]);
hold on;
subplot(4,3,1),plot(p1,'c');
title('Sweet');
subplot(4,3,2),plot(p2,'r');
title('Sour');
subplot(4,3,3),plot(p3,'g');
title('Putrid');
% Show 1 Graph at last position when click button
function pushbutton1_Callback(hObject, eventdata, handles)
fileName = uigetfile('*.wav');
[qq]=soundSig(fileName);
%--------------------------------------------------------------------
%--------------------------------------------------------------------
%axis([0 30 0 0.04]);
%hold on;
subplot(4,3,4),plot(qq,'b');
title('Test Fruite');
%--------------------------------------------------
[p1]=soundSig('sweettest4');
[p2]=soundSig('souar5');
[p3]=soundSig('exp1');
%--------------------------------------------------
sweet=sum(p1.*p1);
sour=sum(p2.*p2);
```

```
putrid=sum(p3.*p3);

f1=sum(p1.*qq);
f2=sum(p2.*qq);
f3=sum(p3.*qq);
test=sum(qq.*qq);

   A(1,1)=sweet;
   A(1,2)=sour;
   A(1,3)=putrid;
sweetNtest=sum(p1.*qq);
souarNtest=sum(p2.*qq);
putidNtest=sum(p3.*qq);
B(1,1)=sweetNtest;
B(1,2)=souarNtest;
B(1,3)=putidNtest;

test=sum(qq.*qq);
   for i=1:3
      a(1,i) = abs(A(1,i)-B(1,i));
      b(1,i) = abs(B(1,i)-test);
   end
      a;
      b;
      for i=1:3
         if a(1,i) > b(1,i)
            p(i)=abs(a(1,i)-(b(1,i)*2));
         elseif a(1,i) == b(1,i)
            p(i)=abs(a(1,i)-b(1,i));
         else
            p(i)=abs((a(1,i)*2)-(b(1,i)));
         end
```

```
        end
        fprintf('%0.4f \n',p);
    %-----------------------------------------------
    set(handles.text7,'String',sweet);
    set(handles.text8,'String',sour);
    set(handles.text9,'String',putrid);
    %-----------------------------------------------
    S1=(p(1)*100)/A(1,1);
    S2=(p(2)*100)/A(1,2);
    S3=(p(3)*100)/A(1,3);
    set(handles.text2,'String',S1);
    set(handles.text16,'String',S2);
    set(handles.text17,'String',S3);
    %-----------------------------------------------
     if S1 < S2
            if S1< S3
               set(handles.text20,'String','ผลลัพย์ที่ได้คือ หวานresult sweet');

            else
               set(handles.text20,'String','ผลลัพย์ที่ได้คือ เน่า result putrid');

            end
        else
            set(handles.text20,'String','ผลลัพย์ที่ได้คือ เปรี้ยว result souar');

        end
    if ispc
       set(hObject,'BackgroundColor','white');
    else
       set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
    end
```

**เขียนโปรแกรมเพื่อทดสอบสับปะรดตัวอย่างเทียบกับสับปะรดต้นแบบ**

ในส่วนแรกนี้จะเป็นการนำ reference ที่เราได้มา มาตัดสัญญาณโดยใช้ตาเปล่า เพื่อให้ได้สัญญาณ reference ที่มีลักษณะคงที่ในทุก ๆ ครั้งที่มีการทดสอบผลสับปะรด ทำให้การวิเคราะห์เป็นไปอย่างถูกต้องมากที่สุด

```
[y,fs1]=wavread('sweettest1.wav');          เรียก file ที่ต้องการจะใช้เป็น reference ของลูกสุก
y1=y(1900:2500);                            ตัดช่วงของสัญญาณที่ใช้ เพื่อให้ได้สัญญาณเดาะ 1 ครั้ง
Z1=fft(y1);                                 ทำ FFT สัญญาณที่ตัดช่วงได้
V1=fft(y1,fs1);                             ทำ FFT สัญญาณที่ตัดช่วงได้ โดยทำเทียบกับความถี่ ที่ใช้ในการ sampling สับปะรด
Y1=abs(V1);                                 นำเฉพาะค่า absolute ของการทำFFTมาใช้
T1=Y1(1:400);                               ตัดช่วงของค่า absolute เพื่อใช้แสดงช่วงที่มีผลมาก
[F1,x1]=max(T1);                            หาค่าของ peak แรก ว่ามีความถี่เท่าไหร่ เก็บที่ค่า x1
[t,fs2]=wavread('souar3.wav');              เรียก file ที่ต้องการจะใช้เป็น reference ของลูกดิบ
t1=t(1900:2500);
Z2=fft(t1);
V2=fft(t1,fs2);
Y2=abs(V2);
T2=Y2(1:400);
[F2,x2]=max(T2);
[r,fs3]=wavread('exp2.wav');                เรียก file ที่ต้องการจะใช้เป็น reference ของลูกเน่า
r1=r(1900:2500);
Z3=fft(r1);
V3=fft(r1,fs3);
Y3=abs(V3);
T3=Y3(1:400);
[F3,x3]=max(T3);
```

ในส่วนที่ 2 นี้จะเป็นการนำสัญญาณเสียงของผลสับปะรดที่ต้องการทดสอบมาตัดสัญญาณและทำตามกระบวนการอื่น ๆ ต่อไป

```
beginp4=1;                          กำหนดค่าเริ่มต้นให้แก่ลูปเพื่อหาจุดเริ่มต้นในการตัด สัญญาณ
endp4=1;                            กำหนดค่าเริ่มต้นให้แก่ลูปเพื่อหาจุดสิ้นสุดในการตัด สัญญาณ
h=wavread('14_Aug 5.wav');          เรียก file ของการเคาะครั้งที่1 สับปะรดลูกที่เราต้องการทดสอบ รับค่า
for j=10000:100000                  เริ่มการวนลูปตั้งแต่ sample ตำแหน่งที่ 1000 ถึง 10000
if abs(h(j))-abs(h(j-50))>0.3       หาจุดเริ่มต้นของการตัดช่วงและจะหยุดหาเมื่อเจอจุดที่สัญญาณเริ่มมี Amplitude ที่สูงกว่าช่วงก่อนหน้า 50 ตำแหน่ง เกิน0.3
beginp4=j-500;                      เมื่อได้จุดที่สัญญาณเริ่มสูงขึ้น เพื่อให้ได้สัญญาณที่ครบถ้วนจึงถอยกลับ 500 ตำแหน่ง และให้จุดที่ถอยกลับมานี้เป็นตำแหน่งเริ่มต้นของการตัดช่วง
  break;                            เมื่อได้จุดเริ่มต้นแล้วจบการวนลูป
  end;
end;
endp4=beginp4+400;                  จุดสิ้นสุดของช่วงที่ตัดจากจุดเริ่มต้นเอามาอีก 4000 ตำแหน่งจะได้สัญญาณครบช่วงที่สนใจ
h1=h(beginp4:endp4);                ทำการตัดสัญญาณในช่วงที่ต้องการ
A1=fft(h1);                         ทำ FFT สัญญาณที่ตัดช่วงได้
V4=fft(h1,fs4);                     ทำ FFT สัญญาณที่ตัดช่วงได้ โดยทำเทียบกับความถี่ที่ใช้ในการ sampling
B1=abs(V4);                         นำเฉพาะค่า absolute ของการทำFFTมาใช้
C1=B1(1:400);                       ตัดช่วงของค่า absolute เพื่อใช้แสดงช่วงที่มีผลมาก
[F4,x4]=max(C1);                    หาค่าของ peak แรก ว่ามีความถี่เท่าไหร่ เก็บที่ค่า x4
beginp5=1;
```

```
endp5=1;
k=wavread('6_Sept 8.wav');                    เรียก file ของการเคาะครั้งที่2 สับปะรดลูกที่เรา
                                              ต้องการทดสอบ
for j=10000:100000
  if abs(k(j))-abs(k(j-50))>0.3
  beginp5=j-500;
  break;
  end;
end;

endp5=beginp5+4000;
k1=k(beginp5:endp5);
A2=fft(k1);
V5=fft(k1,fs5);
B2=abs(V5);
C2=B2(1:400);
[F5,x5]=max(C2);

beginp6=1;
endp6=1;
p=wavread('6_Sept 8.wav');                    เรียก file ของการเคาะครั้งที่3 สับปะรดลูกที่เรา
                                              ต้องการทดสอบ
for j=10000:100000
  if abs(p(j))-abs(p(j-50))>0.3
  beginp6=j-500;
  break;
  end;
end;

endp6=beginp6+4000;
p1=p(beginp6:endp6);
A3=fft(p1);
```

```
V6=fft(p1,fs6);
B3=abs(V6);
C3=B3(1:400);
[F6,x6]=max(C3);
subplot(231),plot(T1,'r');title('sweettest1');      plot กราฟสัญญาณ FFT ของ reference ผลสุก
subplot(232),plot(T2,'m');title('souar3');          plot กราฟสัญญาณ FFT ของ reference ผลดิบ
subplot(233),plot(T3,'g');title('exp2');            plot กราฟสัญญาณ FFT ของ reference ผลเน่า
subplot(234),plot(C1);title('sample');              plot กราฟสัญญาณ FFT ของสับปะรดที่
                                                    ต้องการทดสอบในการเคาะครั้งที่1
subplot(235),plot(C2);title('sample');              plot กราฟสัญญาณ FFT ของสับปะรดที่
                                                    ต้องการทดสอบในการเคาะครั้งที่2
subplot(236),plot(C3);title('sample');              plot กราฟสัญญาณ FFT ของสับปะรดที่
                                                    ต้องการทดสอบในการเคาะครั้งที่3
c1=xcorr(abs(A1),abs(Z1),'coeff');                  ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                                    ผลทดสอบเคาะครั้งที่1 และ reference ของผล
                                                    สุก
s1=max(c1);                                         หาค่าสูงสุดของค่า cross correlation
c2=xcorr(abs(A1),abs(Z2),'coeff');                  ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                                    ผลทดสอบเคาะครั้งที่1 และ reference ของผล
                                                    ดิบ
s2=max(c2);                                         หาค่าสูงสุดของค่า cross correlation
c3=xcorr(abs(A1),abs(Z3),'coeff');                  ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                                    ผลทดสอบเคาะครั้งที่1 และ reference ของผล
                                                    เน่า
s3=max(c3);                                         หาค่าสูงสุดของค่า cross correlation
c4=xcorr(abs(A2),abs(Z1),'coeff');                  ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                                    ผลทดสอบเคาะครั้งที่2 และ reference ของผล
                                                    สุก
s4=max(c4);                                         หาค่าสูงสุดของค่า cross correlation
c5=xcorr(abs(A2),abs(Z2),'coeff');                  ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                                    ผลทดสอบเคาะครั้งที่2 และ reference ของผล
                                                    ดิบ
```

```
s5=max(c5);                          หาค่าสูงสุดของค่า cross correlation
c6=xcorr(abs(A2),abs(Z3),'coeff');   ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                     ผลทดสอบเคาะครั้งที่2 และ reference ของผล
                                     เน่า
s6=max(c6);                          หาค่าสูงสุดของค่า cross correlation
c7=xcorr(abs(A3),abs(Z1),'coeff');   ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                     ผลทดสอบเคาะครั้งที่3 และ reference ของผล
                                     สุก
s7=max(c7);                          หาค่าสูงสุดของค่า cross correlation
c8=xcorr(abs(A3),abs(Z2),'coeff');   ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                     ผลทดสอบเคาะครั้งที่3 และ reference ของผล
                                     ดิบ
s8=max(c8);                          หาค่าสูงสุดของค่า cross correlation
c9=xcorr(abs(A3),abs(Z3),'coeff');   ทำการ cross correlation สัญญาณ FFT ระหว่าง
                                     ผลทดสอบเคาะครั้งที่3 และ reference ของผล
                                     เน่า
s9=max(c9);                          หาค่าสูงสุดของค่า cross correlation
if s1>0.9;                           นำค่าสูงสุดของการ cross correlation ของการ
                                     เคาะ ครั้งที่1 มาเปรียบเทียบกับค่าสูงสุดของ
                                     การ cross correlation ของผลสุก ถ้าเกิน 0.7
                                     แสดงว่าสุก
  fprintf('1suk');                   แสดงออกที่หน้าจอว่าเคาะครั้งที่1 สุก
  v=1; else v=0;                     แล้วเก็บค่าไว้ในตัวแปรเพื่อหาผลที่แท้จริง
end;
if s1<0.9 && s2>0.8                  หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                     แล้วน้อยกว่า 0.7 ให้เปรียบเทียบกับ reference
                                     ของผลดิบ ถ้าเกินค่า 0.6 แสดงว่าดิบ
  fprintf('1dip');                   แสดงออกที่หน้าจอว่าเคาะครั้งที่1 ดิบ
  w=1; else w=0;                     แล้วเก็บค่าไว้ในตัวแปรเพื่อหาผลที่แท้จริง
end;
```

```
if s1<0.8 && s3>0.7                    หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                       แล้วน้อย กว่า 0.7 ให้เปรียบเทียบกับ reference
                                       ของผลเน่า ถ้าเกินค่า 0.6 แสดงว่าเน่า
   fprintf('1nout');                   แสดงออกที่หน้าจอว่าเคาะครั้งที่1 เน่า
   x=1; else x=0;                      แล้วเก็บค่าไว้ในตัวแปรเพื่อหาผลที่แท้จริง
end;
if s1<0.9 && s2<0.8 && s3<0.7          หรือถ้าไม่เข้าเงื่อนไขใด ๆ ให้นำค่าเหล่านั้นมา
                                       เปรียบ
   if s1>s2 && s1>s3                   เทียบกันเอง ถ้าค่าใดมีค่ามากที่สุดให้ผลเป็น
                                       ตาม reference นั้น และแสดงออกทางหน้าจอ
      fprintf('1suk');                 ว่าการเคาะ ครั้งที่ 1 เป็นอย่างไร และเก็บค่าไว้
      v=1; else v=0;                   ในตัวแปรนั้น ๆ
   end;
   if s2>s1 && s2>s3
      fprintf('1dip');
      w=1; else w=0;
   end;
   if s3>s1 && s3>s2
      fprintf('1nout');
      x=1; else x=0;
   end;
end;
disp(' ');

if s4>0.9;                             นำค่าสูงสุดของการ cross correlation ของการ
                                       เคาะ ครั้งที่2 มาเปรียบเทียบกับค่าสูงสุดของ
                                       การ cross correlation ของผลสุก ถ้าเกิน 0.7
                                       แสดงว่าสุก
   fprintf('2suk');                    แสดงออกที่หน้าจอว่าเคาะครั้งที่2 สุก
   v=v+1;                              ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                       เก็บค่า ไว้เพื่อหาผลที่แท้จริง
end;
```

```
if s4<0.9 && s5>0.8                    หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                       แล้วน้อย กว่า 0.7 ให้เปรียบเทียบกับ reference
                                       ของผลดิบ ถ้าเกินค่า 0.6 แสดงว่าดิบ
   fprintf('2dip');                    แสดงออกที่หน้าจอว่าเคาะครั้งที่2 ดิบ
   w=w+1;                              ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                       เก็บค่า ไว้เพื่อหาผลที่แท้จริง
end;

if s4<0.8 && s6>0.7                    หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                       แล้วน้อย กว่า 0.7 ให้เปรียบเทียบกับ reference
                                       ของผลเน่า ถ้าเกินค่า 0.6 แสดงว่าเน่า
   fprintf('2nout');                   แสดงออกที่หน้าจอว่าเคาะครั้งที่2 เน่า
   x=x+1;                              ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                       เก็บค่า ไว้เพื่อหาผลที่แท้จริง
end;

if s4<0.9 && s5<0.8 && s6<0.7          หรือถ้าไม่เข้าเงื่อนไขใด ๆ ให้นำค่าเหล่านั้นมา
                                       เปรียบเทียบกันเอง ถ้าค่าใดมีค่ามากที่สุดให้ผล
   if s4>s5 && s4>s6                   เป็นตาม reference นั้น และแสดงออกทาง
                                       หน้าจอว่าการเคาะ
      fprintf('2suk');
      v=v+1;                           ครั้งที่ 2 เป็นอย่างไร และบวก1เข้าไปในตัว
                                       แปรนั้นๆ
   end;
   if s5>s4 && s5>s6
      fprintf('2dip');
      w=w+1;
   end;
   if s6>s4 && s6>s5
      fprintf('2nout');
      x=x+1;
   end;
```

```
end;
disp(' ');
if s7>0.9;                          นำค่าสูงสุดของการ cross correlation ของการ
                                    เคาะครั้ง ที่3 มาเปรียบเทียบกับค่าสูงสุดของ
                                    การ cross correlation ของผลสุก ถ้าเกิน 0.7
                                    แสดงว่าสุก
   fprintf('3suk');                 แสดงออกที่หน้าจอว่าเคาะครั้งที่3 สุก
   v=v+1;                           ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                    เก็บค่า ไว้เพื่อหาผลที่แท้จริง
end;
if s7<0.9 && s8>0.8                 หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                    แล้วน้อย กว่า 0.7 ให้เปรียบเทียบกับ reference
                                    ของผลดิบ ถ้าเกินค่า 0.6 แสดงว่าดิบ
   fprintf('3dip');                 แสดงออกที่หน้าจอว่าเคาะครั้งที่3 ดิบ
   w=w+1;                           ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                    เก็บค่าไว้เพื่อหาผลที่แท้จริง
end;
if s7<0.8 && s9>0.7                 หรือถ้าเปรียบเทียบกับ reference ของผลสุก
                                    แล้วน้อย กว่า 0.7 ให้เปรียบเทียบกับ reference
                                    ของผลเน่า ถ้าเกินค่า 0.6 แสดงว่าเน่า
   fprintf('3nout');                แสดงออกที่หน้าจอว่าเคาะครั้งที่3 เน่า
   x=x+1;                           ถ้าเป็นตามเงื่อนไขให้บวก1 เข้าไปในตัวแปรที่
                                    เก็บค่าไว้เพื่อหาผลที่แท้จริง
end;
if s7<0.9 && s8<0.8 && s9<0.7       หรือถ้าไม่เข้าเงื่อนไขใดๆ ให้นำค่าเหล่านั้นมา
                                    เปรียบเทียบกันเอง ถ้าค่าใดมีค่ามากที่สุดให้ผล
   if s7>s8 && s7>s9                เป็นตาม reference นั้นและแสดงออกทางบวก1
   fprintf('3suk');                 หน้าจอว่าการเคาะครั้งที่ 2 เป็นอย่างไร และ
   v=v+1;                           เข้าไปในตัวแปรนั้นๆ
   end;
   if s8>s7 && s8>s9
   fprintf('3dip');
```

```
        w=w+1;
    end;
        if s9>s7 && s9>s8
        fprintf('3nout');
        x=x+1;
    end;
end;
disp(' ');
 if v>=2;

    fprintf('suk')
end;
if w>=2;
    fprintf('dip')
end; %

if x>=2;
    fprintf('nout')
end;
```

จากค่าที่เก็บไว้ในตัวแปรแต่ละตัวของ reference ต่าง ๆ นำมาพิจารณา ในการเคาะ 3 ครั้ง ถ้าตัวแปรของ reference ใดมีค่ามากกว่าหรือเท่ากับ 2 ก็แสดงว่าสับปะรดผลที่นำมาทดสอบมีลักษณะเหมือนกับ reference นั้นๆ และแสดงผลสรุปนั้นๆ ออกมา

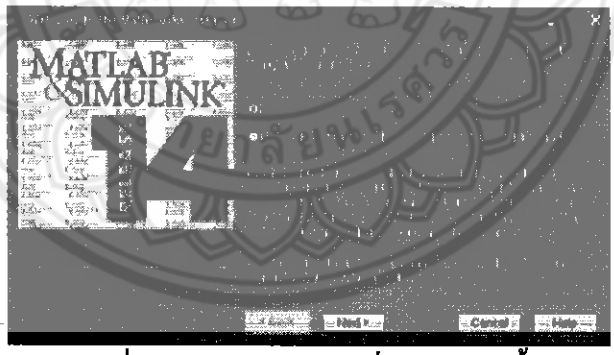
# ภาคผนวก ข.

## การติดตั้งโปรแกรม MATLAB

1. ดับเบิ้ลคลิกที่ Setup.exe เพื่อเริ่มต้นการติดตั้งโปรแกรม

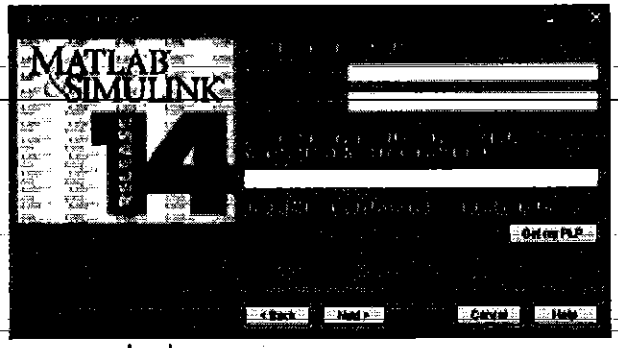
รูปที่ ข.- 1 ภาพหน้าแรกของการติดตั้ง

2. เลือก install เพื่อทำการติดตั้ง จากนั้นคลิก next



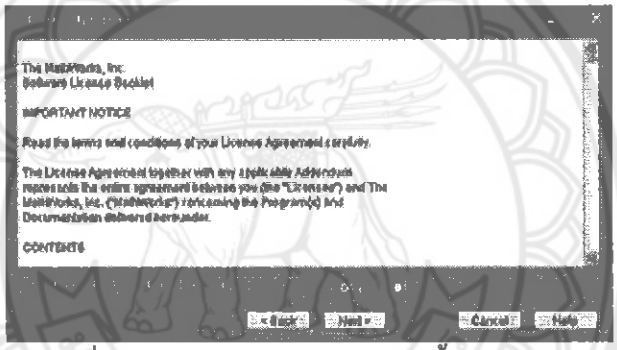

รูปที่ ข.- 2 ภาพเลือกชนิดที่ต้องการติดตั้ง

3. ใส่ชื่อ ชื่อองค์กร และ Personal License Password(PLP) จากนั้นคลิก next



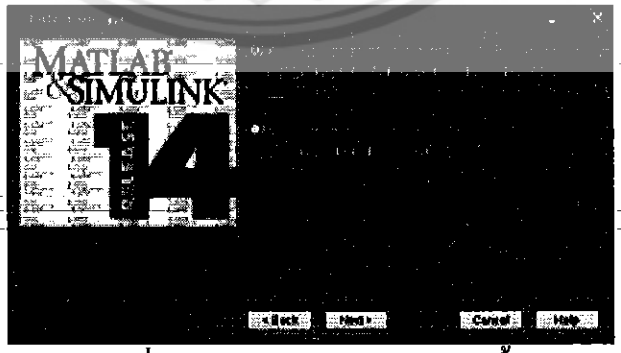

**รูปที่ ข.- 3** ภาพใส่ชื่อ ชื่อองค์กร และ Personal License Password(PLP)

4. ข้อตกลงในการติดตั้งโปรแกรม คลิกที่ yes จากนั้นต่อด้วย next



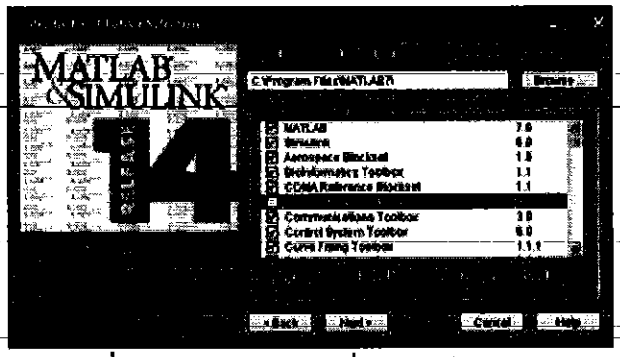
รูปที่ ข.- 4 ภาพข้อตกลงในการติดตั้งโปรแกรม

5. เลือกประเภทการติดตั้งโปรแกรม เลือก custom จากนั้นคลิก next



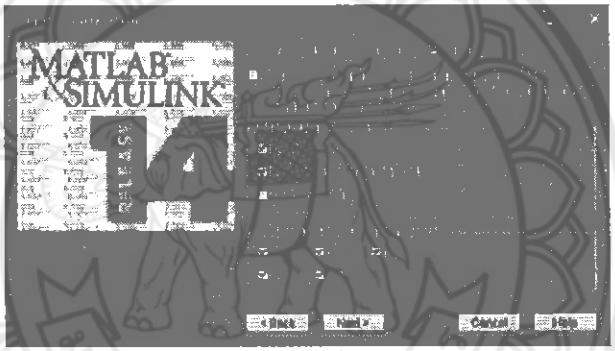

**รูปที่ ข.- 5** ภาพประเภทของการติดตั้ง

6. เลือกไดเรคทอรี่และองค์ประกอบที่ต้องการติดตั้งโปรแกรมลงในเครื่อง จากนั้นคลิก next



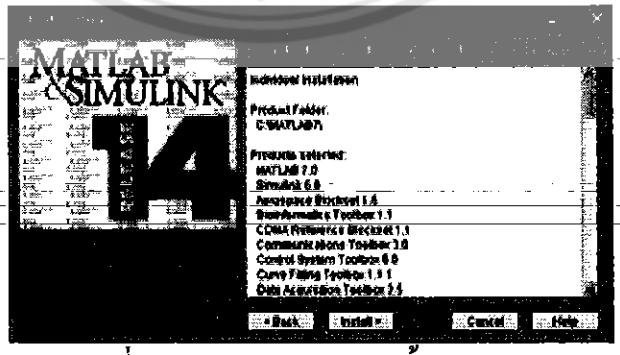

**รูปที่ ข.- 6** ภาพไดเรคทอรี่และองค์ประกอบ

7. เลือกตระกูลไฟล์และส่วนที่ต้องการให้แสดงการเข้าสู่โปรแกรม จากนั้นคลิก next



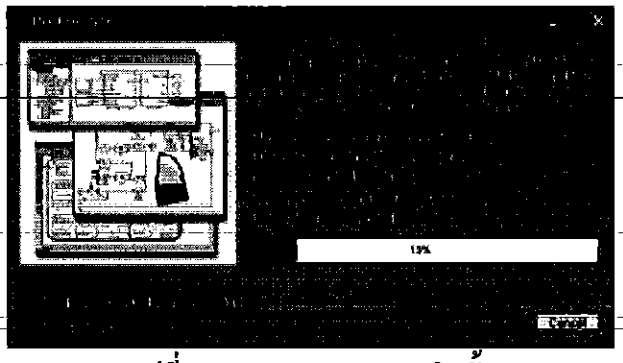

**รูปที่ ข.- 7** การเลือกตระกูลไฟล์และส่วนที่ต้องการให้แสดงการเข้าสู่โปรแกรม

8. ทำการยืนยันการติดตั้งโปรแกรม จากนั้นคลิก install



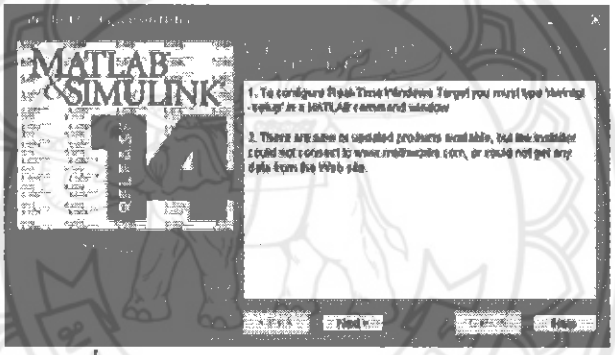

**รูปที่ ข.- 8** ภาพยืนยันการติดตั้งโปรแกรม

9. สถานะการติดตั้งที่แสดงให้ทราบว่าการดำเนินการติดตั้งเป็นไปกี่เปอร์เซนต์แล้ว



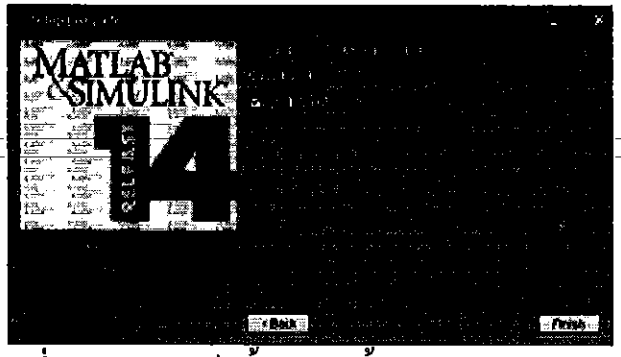
**รูปที่ ข.- 9** ภาพสถานะการติดตั้ง

10. แสดงค่าการติดตั้งโปรแกรม จากนั้นคลิก next




**รูปที่ ข.- 10** ภาพ Product Configuration Notes

12. เมื่อทำการติดตั้งสมบูรณ์แล้วจะเลือกว่าต้องการเริ่มทำงานโปรแกรมหรือไม่ จากนั้นคลิก Finish




**รูปที่ ข.- 11** ภาพเสร็จสิ้นการติดตั้งโปรแกรม MATLAB

# ประวัติผู้ทำโครงงาน

**ชื่อ :** ประเมษฐ ชำนาญยา

**ภูมิลำเนา :** 235/3 ม.11 ต.บ้านต๋อม อ.เมือง จ.พะเยา 56000

**ประวัติการศึกษา :**

- จบการศึกษาระดับมัธยมจาก
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

**ชื่อ :** เชษฐา โทสวนจิต

**ภูมิลำเนา :** 64 ถ.เปรมปรีดา ต.ธาตุเชิงชุม อ.เมือ จ.สกลนคร 47000

**ประวัติการศึกษา :**

- จบการศึกษาระดับมัธยมจากโรงเรียนสกลราชวิทยานุกูล
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

**ชื่อ :** ธนาภาค ธนาเอกภิญโญ

**ภูมิลำเนา :** 36 ถ.บุษบา ซอย 1 อ. เมือง จ. พิจิตร 66000

**ประวัติการศึกษา :**

- จบการศึกษาระดับมัธยมจากโรงเรียนพิจิตรพิทยาคม จังหวัดพิจิตร
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก

# ประวัติผู้ทำโครงงาน (ต่อ)

**ชื่อ :** กฤษณะ อัศวกุลเดช

**ภูมิลำเนา :** 112/37-38 ถ.เอกาทศรฐ อ.เมือง จ.พิษณุโลก 65000

**ประวัติการศึกษา :**

- จบการศึกษาระดับมัธยมจากโรงเรียนพิษณุโลกพิทยาคม
- ปัจจุบันกำลังศึกษาระดับปริญญาตรี ชั้นปีที่ 4

  สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

  คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร จังหวัดพิษณุโลก