



เครื่องตรวจจับการทุบและกรีดกระจก

GLASS-BREAK DETECTOR

โดย

นายเกรียงศักดิ์ ทองสกุล 34161101

นายเอกพล บุญรัตน์ 34161110

นายอมรเดช สุขเกษม 34161142

ปริญญานิพนธ์ฉบับนี้

เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะวิศวกรรมศาสตร์

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032569



ปริญญานิพนธ์ ปีการศึกษา 2/2535

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม
สาขาวิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตรวจจับการทุบและการกรีดกระจก
GLASS-BREAK-DETECTOR

ผู้จัดทำ

1. นายเกรียงศักดิ์ ทองสกุล 34161101
2. นายเอกพล บุณรัตน์ 34161110
3. นายอมรเดช สุขเกษม 34161142

..... อาจารย์ที่ปรึกษา
(อาจารย์ ทรงชัย วีระทวีมาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการน 032569

หัวข้อปริญญาโท เครื่องตรวจจับการทาบและการกริดกระจก

นักศึกษา นายเกรียงศักดิ์ ทองสกุล 34.161101

นายเอกพล บุญรัตน์ 34.161110

นายอมรเดช สุขเกษม 34.161142

อาจารย์ที่ปรึกษา อาจารย์ ทรงชัย วิระทวีมาศ

ระดับการศึกษา อุตสาหกรรมศาสตรบัณฑิตทางเทคโนโลยี

การวัดคุมอุตสาหกรรม

ปีการศึกษา พ.ศ. 2535

บทคัดย่อ

เครื่องตรวจจับการทาบและการกริดกระจกนี้ อาศัยหลักการวิเคราะห์ทางดิจิตอล โดยแปลงสัญญาณเสียงที่เข้ามาให้เป็นสัญญาณดิจิตอลนำข้อมูลที่ได้อไปวิเคราะห์หาสเปคตรัม ข้อมูลที่ได้นำไปเก็บเป็น pattern ต้นแบบและนำไปเปรียบเทียบกับข้อมูลใหม่ที่เข้ามาว่า มีความใกล้เคียงหรือไม่ วิธีการที่ใช้ในการแยกแยะความถี่เพื่อใช้เป็นข้อมูลสำหรับเปรียบเทียบ เป็นการนำวิธีการ Fast Fourier Transform มาใช้ ซึ่งจะประหยัดอุปกรณ์ด้าน hardware นอกจากการวิเคราะห์โดยใช้ Fast Fourier Transform แล้วอาจจะใช้วิธี LPC ซึ่งผู้จัดทำก็ได้ทดลองทั้งสองวิธี แต่วิธี LPC ยังมีปัญหาที่การ interrupt ของส่วน hardware ซึ่งใช้ IC Digital signal processing เบอร์ TMS 32010-10 แต่ยังไม่สมบูรณ์ จึงได้รวบรวมรายละเอียดไว้ในภาคผนวกด้วย

Thesis : GLASS-BREAK DETECTOR

Student : Mr.Kriengsak Thongsakul 34.161101

Mr.Akapol Bunrat 34.161110

Mr.Amorndet Sukkasem 34.161142

Advisor : Mr.Songchai Weerathaweemart

**Education Level : Bachelor of industrial instrument
technology**

Education year : 1992

ABSTRACT

The thesis is concerned with glass-break detector, in principle of analysis with digital by convert analog signal to digital signal. That bring parameter save to be template to separate with new signal. The stored data are analyzed by using "Fast Fourier Transform (FFT)" technique to transform input sound signal in time domain to spectrum in frequency domain. The other method in this thesis is use LPC to analy signal with CPU TMS32010 but it have problem in program then take detail program and circuit in last topic of thesis,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีการวิเคราะห์เสียง	3
2.1 หลักในการวิเคราะห์เสียงพูด	4
2.2 การแปลงฟูเรียร์	9
2.3 การแปลงฟาสต์ฟูเรียร์	13
บทที่ 3 วงจรและการทำงาน	39
3.1 วงจรและการทำงานของการ์ด	41
3.2 การทำงานของโปรแกรม	42
บทที่ 4 การทดลองและผลการทดลอง	51
บทที่ 5 สรุปผลการทดลอง อุปสรรค และข้อเสนอแนะ	66
ภาคผนวก ก. ทฤษฎี LINEAR PREDICTION CODING	68
ภาคผนวก ข. วงจรและโปรแกรมของ TMS 32010	80
ภาคผนวก ค. โปรแกรมที่ใช้วิเคราะห์ โดยวิธี FFT	103
กิตติกรรมประกาศ	
บรรณานุกรม	

บทที่ 1

1.1 บทนำ

ในปัจจุบันการพัฒนาเทคโนโลยีของไทย ยังน้อยมากเมื่อเทียบกับประเทศอื่น ๆ โครงงานนี้ก็เป็นการนำเอาเทคโนโลยีที่กำลังพัฒนาอยู่นำมาประยุกต์ใช้ให้เกิดผลขึ้นมา โดยอาศัยหลักการ วิเคราะห์เสียง นำเสียงที่ได้ไปหาค่าพารามิเตอร์ ต่าง ๆ นำไปเก็บไว้ เป็นต้นแบบ ไว้คอยเปรียบเทียบกับเสียงใหม่ที่เข้ามา เสียงที่นำไปเก็บก็แล้วแต่ว่าจะเป็นเสียงอะไร ซึ่งหลักการที่นำมาใช้ เป็นหลักการของ การวิเคราะห์ (speech recognition) ซึ่งมีการวิจัยในด้านนี้ไว้มากมาย จึงน่าจะสมควรถิ่น นำมาประยุกต์ใช้ในการจำเสียงแก้วแตกหรือการกรีดกระจก ซึ่งอาจจะมีการโจรกรรมเกิดขึ้น ก็ให้เครื่องรับรู้ และทำการ เตือน หรือแจ้งเหตุต่อไป

1.2 วัตถุประสงค์

เพื่อให้ในการตรวจจับการทุบหรือการกรีดกระจก ซึ่งอาศัยหลักการวิเคราะห์ความถี่เสียง โดยจะเก็บแถบความถี่เสียงที่มีค่าพลังงานสูง ๆ ไว้เป็นตัวเปรียบเทียบ เมื่อเกิดเสียงแบบเดียวกันก็ให้ตรวจเช็คได้ ในส่วนนี้อาจเพิ่มการทำงานด้านอื่นหากมีเวลาพอ เช่น เป็นรหัสผ่านโดยใช้เสียงตามความถี่ที่ตั้งไว้

1.3 ขอบเขตของโครงงาน

1. สามารถตรวจเช็คสัญญาณเสียง การทุบแก้ว หรือการกรีดกระจกได้
2. สามารถจำและแยกแยะสัญญาณอื่น ๆ อีกได้ในขนาดหนึ่ง

1.4 ทั้ตอนและวิธีดำเนินการ

1. ศึกษากระบวนการเกิดเสียงในลักษณะต่าง ๆ
2. ศึกษากระบวนการจดจำเสียงและวิเคราะห์เสียงโดยวิธีต่าง ๆ
3. คัดเลือกวิธีที่เหมาะสมในการรับรู้เสียง โดยคำนึงถึงเวลาในการตอบสนองและความถูกต้องในการรับรู้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. พัฒนาอุปกรณ์เพื่อใช้ในการเปลี่ยนสัญญาณเสียงไปเป็นสัญญาณดิจิทัล เพื่อให้เครื่องคอมพิวเตอร์สามารถนำไปใช้ประมวลผลในระยะแรกได้
5. พัฒนาโปรแกรมสำหรับการวิเคราะห์สัญญาณเสียง การสร้างแบบอ้างอิง และการทดสอบการรับรู้
6. พัฒนาส่วนประมวลผลข้างนอก ซึ่งเป็น Digital signal processing
7. พัฒนาโปรแกรมในส่วนของ Digital signal processing
8. พัฒนาส่วนติดต่อกับอุปกรณ์ทั้งหมด
9. ทำการทดสอบและปรับปรุงคุณภาพในการรับรู้ รวบรวมข้อมูล และประเมินผล

1.5 ประโยชน์ที่จะได้จากโครงงานนี้

1. เป็นแนวทางในการพัฒนาระบบการจดจำเสียงในลักษณะอื่น ๆ อีกต่อไป
2. สามารถนำไปประยุกต์ใช้กับงานอื่นได้
3. ในส่วนของระบบประมวลผล TMS32010 ที่จะพัฒนาขึ้น สามารถประยุกต์ในงานด้านอื่น ๆ อีกมากมาย ซึ่งจะเป็นประโยชน์มาก

ในปฏิญานพนธ์นี้จะกล่าวถึงทฤษฎีการวิเคราะห์เสียงและผลการทดลองที่จะสรุปได้ดังนี้

ในบทที่ 2 จะกล่าวถึงวิธีการวิเคราะห์เสียงโดยใช้การแปลงฟูเรียร์ และอัลกอริทึมฟาสฟูเรียร์ช่วยในการวิเคราะห์

ในบทที่ 3 เป็นรายละเอียดของวงจรและการทำงาน ในส่วนของการ์ด ที่ต่อกับเครื่องคอมพิวเตอร์

ในบทที่ 4 เป็นการทดลองและผลการทดลอง

ในบทที่ 5 เป็นการสรุปผลการทดลอง อุปสรรค และข้อเสนอแนะต่างๆ

ส่วนของโปรแกรมที่ใช้แสดงในภาคผนวก ซึ่งมีส่วนที่ทดลองโดยใช้วิธี LPC (Linear prediction coding) ด้วย โดยใช้ CPU เบอร์ TMS32010

2.1 หลักการในการวิเคราะห์เสียง

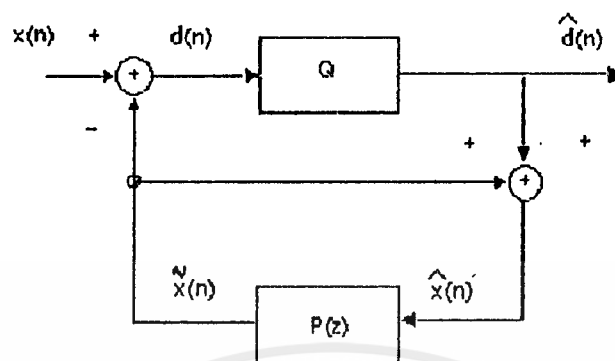
เราสามารถที่จะเก็บสัญญาณเสียงไว้ในรูปของสัญญาณดิจิทัลได้แล้วแต่เนื่องจากข้อมูลที่ได้ยังมีปริมาณมากเกินไป ไม่เหมาะสมที่จะนำมาวิเคราะห์โดยตรง จึงควรที่จะลดความซ้ำซ้อนของข้อมูลลงเก็บไว้เฉพาะที่จำเป็นเท่านั้นในรูปแบบของพารามิเตอร์ ซึ่งสามารถที่จะลดปริมาณของหน่วยความจำที่ใช้เก็บข้อมูลลง ทั้งนี้การวิเคราะห์สัญญาณเสียงในด้านต่าง ๆ อาจในด้านของโดเมนเวลา (time domain) คือพิจารณาจากรูปคลื่นของสัญญาณเสียงตามแกนของเวลาโดยตรงหรือในด้านของโดเมนความถี่ (frequency domain) คือพิจารณาจากสเปกตรัมของสัญญาณเสียง

ในการวิเคราะห์เสียงสามารถแบ่งออกเป็น 3 ลักษณะใหญ่ ๆ คือ ในลักษณะแรกจะกล่าวถึง วิธีการเปลี่ยนสัญญาณเสียง หรือสัญญาณอนาล็อกไปเป็นสัญญาณดิจิทัล ซึ่งจะนำไปใช้ในการวิเคราะห์ต่อไป ในลักษณะที่สองจะกล่าวถึงการวิเคราะห์ในโดเมนของเวลา ซึ่งเป็น การวิเคราะห์โดยตรงกับสัญญาณ และลักษณะสุดท้ายจะเป็นการวิเคราะห์ในโดเมนความถี่ ซึ่งจะกล่าวในรายละเอียดในหัวข้อหลักต่อไป.

2.1.1 การนำสัญญาณเข้ารหัสเชิงเลข (Digital Waveform Coding)

หลักการคือ แปลงสัญญาณเสียงให้เป็นข้อมูลเชิงเลขโดยตรง แบ่งได้เป็น 2 วิธีคือ

1. พัลส์โคดมอดดูเลชัน (Pulse Code Modulation) หรือพีซีเอ็ม (PCM) คือ การสุ่มตัวอย่างสัญญาณอนาล็อกในช่วงเวลาคงที่ ย่านความถี่ที่ได้จะเท่ากับความถี่ครึ่งหนึ่งของความถี่ในการสุ่ม วิธีนี้เป็นวิธีที่ง่ายที่สุด แต่ข้อมูลเสียงที่ได้มากไม่เหมาะสม สำหรับนำมาวิเคราะห์และเก็บเป็นต้นแบบเพราะเปลืองหน่วยความจำและต้องใช้เวลามาก
2. ดิฟเฟอเรนเชียลควอนไทเซชัน (Differential Quantization)
วิธีการนี้เกิดจากความจริงที่ว่า ในการสุ่มตัวอย่างเสียงพูดตาม Nyquist Rate ค่าสหสัมพันธ์ (Correlation) ของสัญญาณที่ทำการสุ่มต่อเนื่องกันยังคงมีค่าสูง นั่นคือเรามีโอกาสทำนายค่าของสัญญาณได้ไม่ยากนัก จากรูป 2.1.1



รูปที่ 2.1.1 แสดงแผนภาพหลักการทำงานของดีเฟอว์เรนเชียลควอนไทเซชัน

2.1.2 การวิเคราะห์ในโดเมนเวลา

การประมวลผลสัญญาณเสียงในโดเมนของเวลานี้ มีประโยชน์ในด้านการประมวลผลได้ไม่ยาก การคำนวณใช้เวลาไม่มากนักและง่ายในการวิเคราะห์พิจารณา การวิเคราะห์ โดยยึดหลักพิจารณาจากรูปคลื่นสัญญาณที่แปรตามแกนเวลาซึ่งมีอยู่หลายวิธีด้วยกัน เช่น เอ้าท์พุทที่ได้จากการเข้ารหัสพีซีเอ็มตั้งแต่ได้กล่าวมาแล้ว ผลของการวิเคราะห์เสียงในด้านของโดเมนเวลาก็มีเช่น ค่าพลังงานหรือแอมพลิจูดของเสียง การหาคาบของพิทช์

1. การวัดพลังงานเสียง

คือการนำค่า amplitude ไปคำนวณหาค่าพารามิเตอร์

2. Shot-Time Autocorrelation Analysis

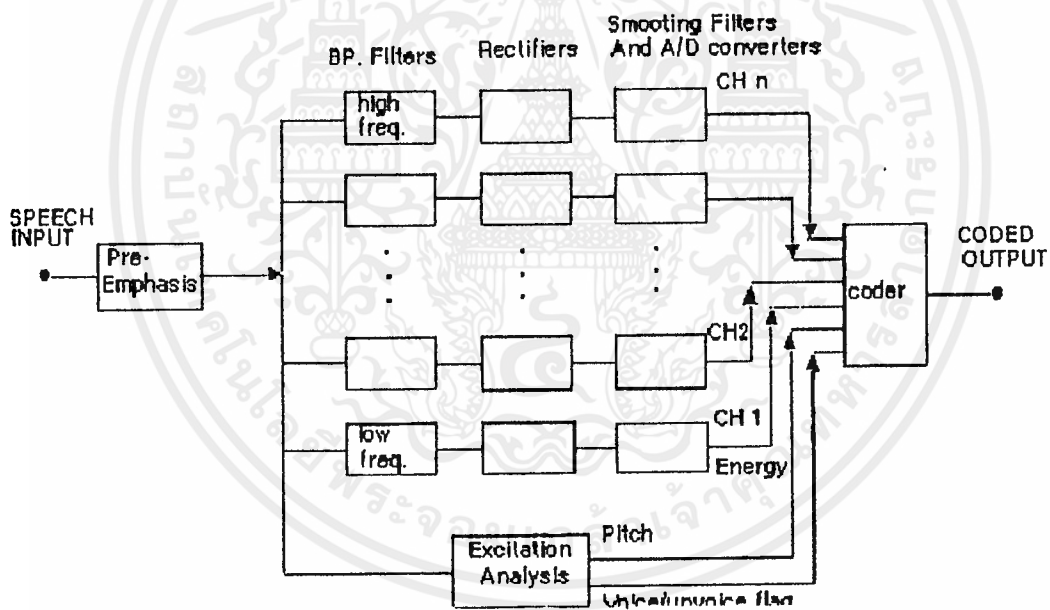
เป็นวิธีการวิเคราะห์ความเป็นคาบของสัญญาณเสียงซึ่งวิธีการในการวิเคราะห์เสียงส่วนใหญ่แล้ว จะสมมติให้สัญญาณเสียงมีคุณสมบัติที่เปลี่ยนแปลงสัมพันธ์กับเวลาอย่างช้า ๆ นั่นคือเราต้องแบ่งช่วงเวลาเป็นช่วงแคบ ๆ เหมือนมองผ่านห้องแคบ ๆ เมื่อเทียบตามเวลาที่มีการเปลี่ยนแปลง

2.1.3 การวิเคราะห์ทางโดเมนความถี่ (Frequency Domain Analysis)

หลักการนี้ จะอาศัยการเปลี่ยนรูปข้อมูลระหว่างโดเมนเวลา กับ โดเมนความถี่ เทคนิคนี้ จะช่วยในการระบุถึงส่วนประกอบทางความถี่ (spectrum) ของคลื่นเสียง ตัวอย่างของการวิเคราะห์ทางสเปกตรัมได้แก่

1. เทคนิคของ Channel Vocoder

เป็นเทคนิคทางด้านกรวิเคราะห์สัญญาณเสียง โดยใช้สเปกตรัมที่นิยมมากอันหนึ่ง เนื่องจากสามารถทำได้ง่ายแล้วได้ผลออกมาทันทีทันใดและไม่สิ้นเปลืองมากนัก โดยการใช้กลุ่มของตัวกรองแบบ ย่านความถี่ผ่าน ตัวกรองแต่ละอันจะทำการกรองเอาความถี่ที่สำคัญ ทั้งหมดของสัญญาณเสียงอินพุต



รูปที่ 2.1.2 แสดงหลักการของ Channel Vocoder

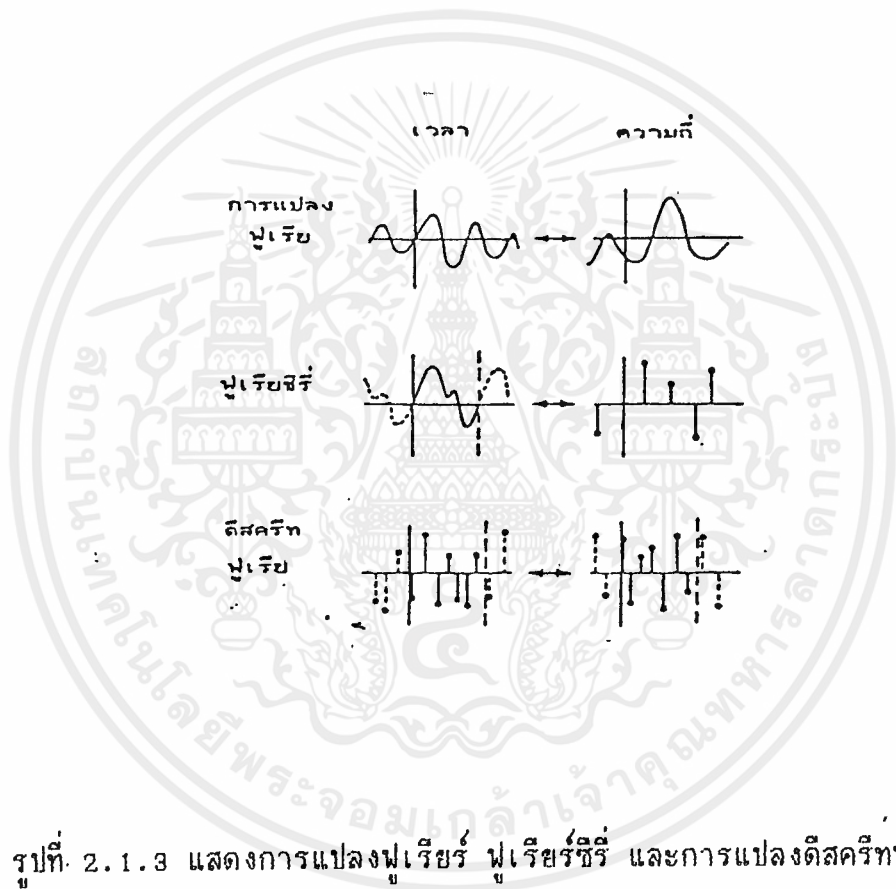
2. การวิเคราะห์ฟูเรียร์ในช่วงสั้น ๆ (Short-time Fourier Analysis)

เป็นวิธีการดั้งเดิมของการวิเคราะห์สัญญาณทางด้านสเปกตรัมและเฟสซึ่งก็เป็นฟังก์ชันอันหนึ่งของความถี่

ฟูเรียร์ซีรีส์เป็นการแปลงระหว่างฟังก์ชันที่เป็นคาบในโดเมนเวลากับฟังก์ชันแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไม่ต่อเนื่องในโดเมนความถี่ ถ้าเราทำการแปลงรูปในโดเมนเวลาที่มีลักษณะเป็นฟังก์ชันแบบไม่ต่อเนื่องและมีช่วงที่จำกัด เราจะได้รูปคลื่นในโดเมนของความถี่ ซึ่งมีช่วงที่จำกัด มีลักษณะเป็นคาบ และเป็นฟังก์ชันที่ไม่ต่อเนื่อง ลักษณะการแปลงแบบนี้ก็คือการแปลงแบบดิสครีตฟูเรีย (Discrete Fourier Transform) ซึ่งก็คือการทำฟังก์ชันที่เป็นคาบแบบไม่ต่อเนื่องในโดเมนเวลาไปเป็นฟังก์ชันที่เป็นคาบและไม่ต่อเนื่องในโดเมนความถี่นั่นเอง



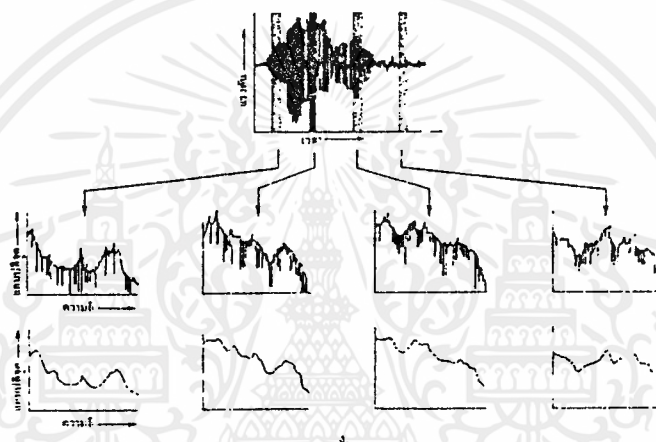
รูปที่ 2.1.3 แสดงการแปลงฟูเรียร์ ฟูเรียร์ซีรีส์ และการแปลงดิสครีตฟูเรียร์

3. การเข้ารหัสแบบลิเนียร์พรีดิคทีฟ (Linear Predictive Coding)

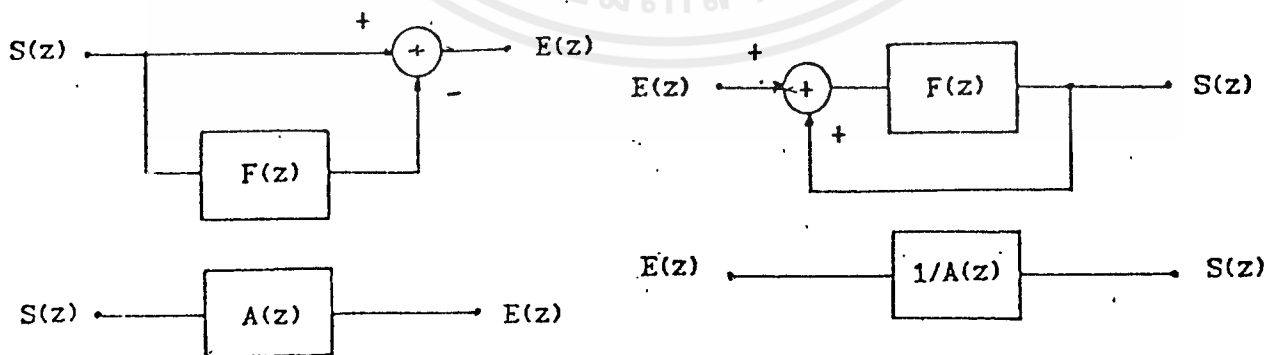
การเข้ารหัสแบบลิเนียร์พรีดิคทีฟ หรือ แอลพีซี เป็นวิธีการวิเคราะห์เสียงที่แพร่หลายมากที่สุดวิธีหนึ่งในปัจจุบัน หลักการเบื้องต้นของแอลพีซี คือ สัญญาณเสียงสามารถประมาณด้วยการนำมารวมกันแบบเชิงเส้นระหว่างสัญญาณที่ลุ่มมาในอดีต บางครั้งก็เรียกวิธีการนี้ว่า autoregressive analysis อาจจะถูกกล่าวได้ว่าวิธีการนี้เป็นวิธีการวิเคราะห์สเปกตรัมทางอ้อม (indirect

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

spectrum analysis) จะสามารถหาค่าแอมพลิจูดของสัญญาณที่เวลาใดๆ v_n ได้ หากรู้แอมพลิจูดของสัญญาณในอดีตก่อนหน้านั้นทั้งหมด v_{n-k} เมื่อ n คือ เลขจำนวนเต็มบวก จากการใช้หลักการของ LPC จะสามารถคำนวณตัวเลขกลุ่มหนึ่งเรียกว่า สัมประสิทธิ์ LPC และตัวเลขเหล่านี้ จะใช้สำหรับการกำหนดลักษณะเฉพาะของเทมเพลตต้นแบบและยังใช้สำหรับวิเคราะห์เปรียบเทียบ ค้นหาเสียงต้นแบบ



รูปที่ 2.1.4 การวิเคราะห์เสียงโดยใช้วงจรมุมลู่ทางคณิตศาสตร์ หรือ LPA



รูปที่ 2.1.5 แบบจำลองการวิเคราะห์และสังเคราะห์เสียงพูดด้วยวิธี LPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2.2 การแปลงฟูรีเยร์ (Fourier Transform)

การแปลงฟูรีเยร์เป็นเครื่องมือวิเคราะห์ในงานวิทยาศาสตร์ สาขาต่างๆ ในปัจจุบัน การประยุกต์ใช้งานที่นิยมใช้วิเคราะห์ระบบเชิงเส้นที่ไม่มีแปรผันตามเวลา (Linear time-invariant system) แต่การแปลงฟูรีเยร์ที่จะพูดถึงต่อไปนี้เป็นหลักหรือคุณสมบัติที่สามารถใช้วิเคราะห์งานได้ทั่วไป

2.2.1 ฟูรีเยร์อินทิกรัล

ฟูรีเยร์อินทิกรัล มีคำจำกัดความ เป็นสมการว่า

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{j2\pi f t} dt$$

$h(t)$ เป็นเทอมของฟังก์ชันของตัวแปรเวลา และ $H(f)$ เป็นเทอมของฟังก์ชันของตัวแปรความถี่ การแปลงฟูรีเยร์ของฟังก์ชันเวลา จะแทนด้วยตัวอักษรตัวใหญ่ ในรูปทั่วไป การแปลงฟูรีเยร์ มีรูปเป็นจำนวนเชิงซ้อน (complex number)

$$H(f) = R(f) + jI(f) = |H(f)| e^{j\theta(f)}$$

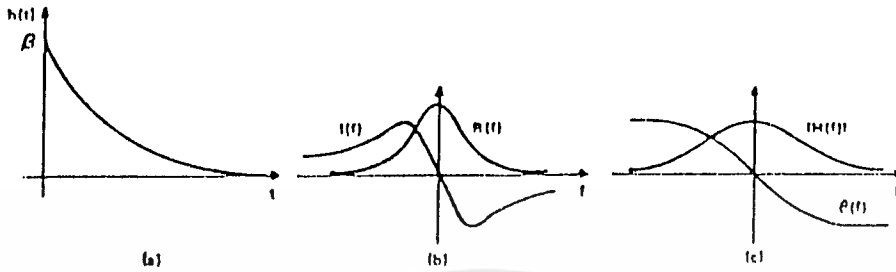
เมื่อ $R(f)$ เป็นส่วนจริงของการแปลงฟูรีเยร์

$I(f)$ เป็นส่วนจินตภาพของการแปลงฟูรีเยร์

$H(f)$ เป็นแอมพลิจูดหรือฟูรีเยร์สเปกตรัมของ $h(t)$ ซึ่งได้จาก

$$\sqrt{R(f)^2 + I(f)^2}$$

$\theta(f)$ เป็นมุมเฟสของการแปลงฟูรีเยร์ ซึ่งได้ $\tan^{-1} [I(f)/R(f)]$



รูปที่ 2.2.1 (a) ตัวอย่างของฟังก์ชันในโดเมนเวลา

(b) ส่วนจริงและส่วนจินตภาพของการแปลงฟูเรียร์

(c) แอมพลิจูดและเฟสของการแปลงฟูเรียร์

2.2.2 การแปลงอินเวอร์สฟูเรียร์ (Inverse fourier transform)

การแปลงอินเวอร์สฟูเรียร์ มีค่าจำกัดความ ดังนี้

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} df$$

2.2.3 การแปลงฟูเรียร์แบบไม่ต่อเนื่อง

(discrete fourier transform หรือ DFT)

DFT เป็นลำดับของความถี่ไม่ต่อเนื่อง ที่ได้จากการสุ่มในหนึ่งคาบเวลาของการแปลงฟูเรียร์

เมื่อกำหนดให้จำนวนการสุ่มเท่ากับ N สุ่มบนคาบเวลา $0 < \omega < 2\pi$

ดังนั้น

$$\omega_k = 2\pi k/N \quad \text{เมื่อ } 0 < k < N-1$$

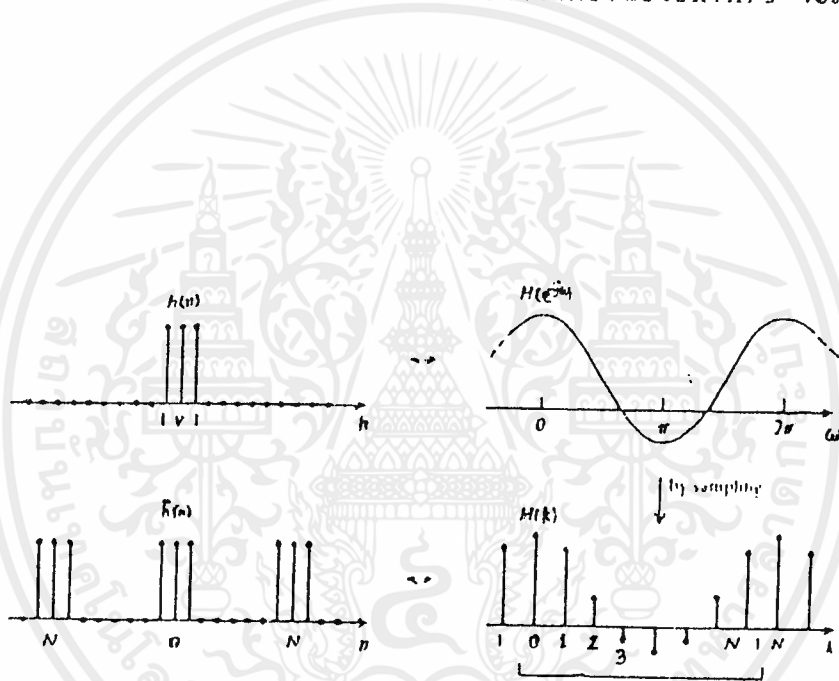
ถ้ากำหนดให้ $[h(n)]$ เป็นลำดับของ discrete time และลำดับของ Fourier Transform $H(e^{j\omega})$ ให้เท่ากับ $[H(k)]$ โดยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H(k) = H(e^{j\omega}) \text{ เมื่อ } \omega = \omega_k = 2\pi k/N \text{ และ } 0 < k < N-1$$

ลำดับของ DFT เริ่มต้นที่ $k=0$ หรือที่ $\omega=0$ แต่ไม่รวมจุด $k=N$ หรือ $\omega=2\pi$

เราจะเห็นว่า $H(e^{j\omega})$ มีคาบเวลาเท่ากับ ω หรือ 2π ซึ่งได้จากการกระจายอนุกรมฟูเรียร์และคำนวณค่าสัมประสิทธิ์จากลำดับ $[h(n)]$ แสดงว่าเมื่อสัญญาณเวลาต่อเนื่องถูกสุ่มด้วยคาบเวลาการสุ่ม T สเปกตรัมของผลลัพธ์จากการแปลงฟูเรียร์จะมีฟังก์ชันคาบเวลาของความถี่เท่ากับ $2\pi T$ จากรูปที่ 1 เมื่อ $H(e^{j\omega})$ จะถูกสุ่มด้วยคาบเวลาการสุ่ม $\omega_k = 2\pi/N$ เมื่อแปลงกลับเป็นลำดับของฟังก์ชันเวลาไม่ต่อเนื่อง $[h(n)]$ จะมีคาบเวลา $2\pi/\omega_k = N$



รูปที่ 2.2.2 แสดงผลลัพธ์ในลำดับของเวลา จาก sampled spectrum

ในกรณีนี้ $N=8$ และอยู่ในช่วง $0 < \omega < 2\pi$

ดังนั้นลำดับ discrete time ที่อยู่ในฟังก์ชันคาบสามารถหาในเทอมของ $[h(n)]$

$$h'(n) = \sum_{m=-\infty}^{\infty} h(n+mN)$$

ลำดับ $[h(n)]$ เรียกว่า periodic extention ของ $[h(n)]$ จำนวนจุดสุ่มในหนึ่งคาบของ spectrum N จะมีค่าเท่ากับคาบของ $[h(n)]$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

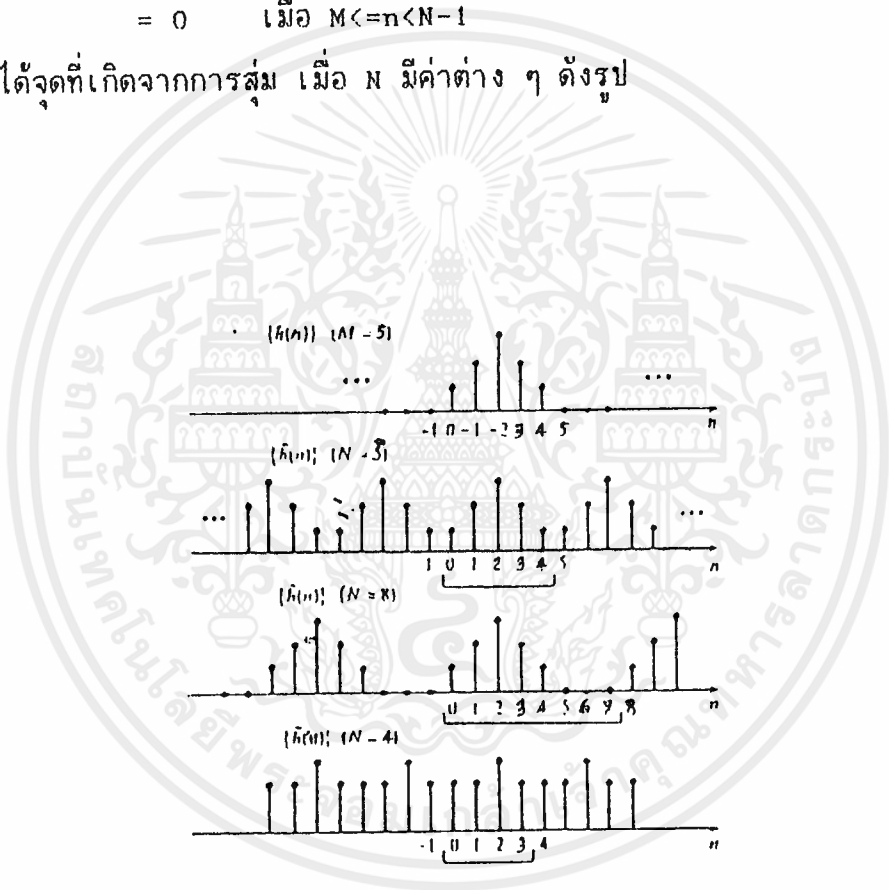
เราสามารถเลือกค่า N เท่าใดก็ได้ ซึ่งเป็นจำนวนการสุ่มของ $H(e^{j\omega})$ ภายใต $0 \leq \omega < 2\pi$ เมื่อ N เป็นคาบของ $[h(n)]$ เราต้องเลือกค่าที่ไม่น้อยเกินไป ในลักษณะที่ $[h(n)]$ เป็นฟังก์ชันคาบเวลาแบบไม่จำกัดทำให้เกิดลักษณะสัญญาณที่ถูกสุ่มทับกันสัญญาณที่ได้ จะผิดพลาดไป สามารถแสดงจากตัวอย่างข้างล่าง

กำหนด

$$h(n) = h(n) \text{ เมื่อ } 0 \leq n < M-1$$

$$= 0 \text{ เมื่อ } M \leq n < N-1$$

จะได้จุดที่เกิดจากการสุ่ม เมื่อ N มีค่าต่าง ๆ ดังรูป



รูปที่ 2.2.3 แสดงความสัมพันธ์ของช่วงเวลาของลำดับ M และจำนวนจุดในการสุ่มสเปคตรัม N

จากรูป จะเห็นว่า ถ้า ค่า $M > N$ รูปสัญญาณจะทับกัน (overlap) หรือเกิด time-aliasing ซึ่งสามารถป้องกันได้โดยเลือกค่า $N \geq M$

2.3 การแปลงฟูรีเยร์เร็ว (FFT)

FFT (fast fourier transform) เป็นอัลกอริทึม (algorithm) ที่ช่วยในการคำนวณ DFT มีประสิทธิภาพมากขึ้นลดจำนวนการกระทำทางคณิตศาสตร์และขนาดของหน่วยความจำที่ต้องใช้ในการคำนวณ DFT อัลกอริทึมของ FFT จึงเป็นเทคนิคที่ถูกเลือกใช้ในการประยุกต์ใช้งานด้านต่างๆ

2.3.1 ลักษณะของปัญหา

ให้ $[X(m)]$ เป็นลำดับของข้อมูล $X(m)$, $m = 0, 1, \dots, N-1$ ที่ได้จากการสุ่มในช่วงจำกัดของสัญญาณ $x(t)$ เรามีอัลกอริทึมที่ใช้คำนวณค่าสัมประสิทธิ์ $C_x(k)$ ดังสมการนี้

$$C_x(k) = 1/N \sum_{m=0}^{N-1} X(m) W^{km}, \quad k = 0, 1, \dots, N-1$$

เมื่อ $W = e^{-j2\pi/N}$, $i = \sqrt{-1}$ เราเรียกสมการนี้ว่า DFT ของ $[X(m)]$ อัลกอริทึมซึ่งเราจะได้กล่าวถึงต่อไปเรียกว่าการแปลงฟูรีเยร์เร็ว (FFT) เป็นอัลกอริทึมที่พัฒนาขึ้นโดย Cooley และ Tukey บางทีอาจจะเรียกว่าอัลกอริทึมของ Cooley และ Tukey ก็ได้ ตั้งแต่บัดนี้ไป เราจะตั้งข้อสมมุติว่า

$$N = 2^n, \quad n = 1, 2, \dots, n_{max}$$

จากที่เราได้เข้ากันแล้ว ถ้าไม่ต้องการให้ข้อมูลสูญหาย ต้องมีขนาด N ที่เหมาะสมตามทฤษฎีการสุ่ม (Sampling theorem) นั่นคือ $N \geq 2BL$ เมื่อ B เป็นแบนวิดธ์ของ $x(t)$ ในหน่วย Hz และ L คือช่วงเวลาของมัน สำหรับในกรณี N ไม่เท่ากับ 2^n เราจะได้พูดถึงกันต่อไป

เราจะพิจารณาในกรณีที่ $[X(m)]$ เป็นค่าจำนวนจริง และ $N = 8$ จากคุณสมบัติคอมเพล็กซ์คอนจูเกต (complex conjugate property) เรามี

$$C_x(4+i) = C_x^*(4-i), \quad i = 1, 2, 3$$

ดังนั้นเมื่อเทียบกับสมการที่ ทำให้การคำนวณเหลือเพียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$C_x(k) = 1/8 \sum_{m=0}^7 X(m)W^{km}, \quad k=0,1,\dots,4$$

เมื่อ $w = e^{-j2\pi/4}$

จาก สมการ (1) จะได้

$$8C_x(k) = \sum_{m=0}^7 X(m)\cos(mk\pi/4) - j\sum_{m=0}^7 X(m)\sin(mk\pi/4) = A(k) - B(k)$$

เมื่อ

$$A(k) = \sum_{m=0}^7 X(m)\cos(mk\pi/4),$$

และ

$$B(k) = \sum_{m=0}^7 X(m)\sin(mk\pi/4), \quad k = 0,1,\dots,4$$

ใช้สัญลัษณ์เมทริกซ์ สมการที่ (2) จะเป็น

$$A = CX$$

$$B = SX$$

เมื่อ A และ B มีมิติ (5×1) , X มีมิติ (8×1) และ C และ S มีมิติ (5×8)

ดังนี้

$$S = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1/\sqrt{2} & 1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} & -1 & -1/\sqrt{2} \\ 0 & 1 & 0 & -1 & 0 & 1 & 0 & -1 \\ 0 & 1/\sqrt{2} & -1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} & 1 & -1/\sqrt{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$$C = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} & -1 & -1/\sqrt{2} & 0 & 1/\sqrt{2} \\ 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\ 1 & -1/\sqrt{2} & 0 & 1/\sqrt{2} & -1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

เมื่อพิจารณาคู จะเห็นว่า สมาชิกของเมทริกซ์ C และ S มีค่าซ้ำกัน ซึ่งเป็นเหตุผล 2 ประการที่พัฒนาอัลกอริทึมฟาสต์ฟูเรียร์ ดังนี้

1. ความเป็นซายน์ (sinusoidal) จัดกลุ่มด้วยลักษณะเด่น ซึ่งจะเกิดขึ้นได้เมื่อเลือกจุดสุ่มได้อย่างถูกต้อง

2. ถ้า N เป็นจำนวนของจุดสุ่มของ $x(t)$ ที่มีรูปแบบไม่แน่นอนแล้วการเกิดซ้ำกันของเมทริกซ์ C และ S จะเพิ่มขึ้นตามจำนวนตัวประกอบของ $N = 2, n = 2 \times 2 \times \dots \times 2$ ดังนั้น เมื่อจำนวนของข้อมูลมาก การใช้อัลกอริทึมฟาสต์ฟูเรียร์จะช่วยประหยัดเวลาและหน่วยความจำอย่างมาก

หนทางที่จะพัฒนาอัลกอริทึมที่ต้องการคือ การหาความสัมพันธ์กับการซ้ำกันรูปแบบทั่วไปของเมทริกซ์ C และ S กับ กำลังของ W หรือรากที่ N ของหนึ่ง (principle N -th root of unity) เพื่อหาสัญลักษณ์ของรูปทั่วไป

สัญลักษณ์ แสดงค่าในฐานสิบแต่ละตัวของ $m, 0 \leq m < N-1$ ให้อยู่ในรูปของเลขฐานสองหรือไบนารี (binary) ดังนี้

$$m = m_{n-1} 2^{n-1} + m_{n-2} 2^{n-2} + \dots + m_1 2^1 + m_0 2^0$$

เมื่อ

$$m_v = 0 \text{ หรือ } 1, v = 0, 1, \dots, n-1, n = \log_2 N$$

ในทำนองเดียวกัน ค่าในฐานสิบแต่ละตัวของ $k, 0 \leq k < N-1$ เปลี่ยนรูปเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$k = k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \dots + k_12^1 + k_02^0$$

เมื่อ

$$kv = 0 \text{ หรือ } 1, v = 0, 1, \dots, n-1$$

ให้รูปแบบไบนารีของ $x(m)$ แทนด้วย $X(m)$ จะได้ว่า

$$\begin{aligned} X(m) &= X(m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_12^1 + m_02^0) \\ &= X(m_{n-1}, m_{n-2}, \dots, m_1, m_0) \end{aligned}$$

เปลี่ยนรูปของสมการตามความสัมพันธ์ที่ได้ ดังนี้

$$\begin{aligned} &\sum_{m=0}^{n-1} X(m)W^{km} \\ &= \sum_{m_0=0}^{m_0=1} \sum_{m_1=0}^{m_1=1} \dots \sum_{m_{n-1}=0}^{m_{n-1}=1} X(m_{n-1}, m_{n-2}, \dots, m_1, m_0) W^{k(m_{n-1}2^{n-1} + \dots + m_02^0)} \end{aligned}$$

ขั้นตอนการพัฒนา algorithm จะอธิบายในกรณี $N = 8 = 2^3$ ดังนั้นจะได้

$$C_x(k) = (1/8) \sum_{m=0}^7 X(m)W^{km}, \quad k=0, 1, \dots, 7$$

เมื่อ $w = e^{-j\pi/4}$ เราแสดงค่าของ m ในรูปไบนารีดังนี้

$$m = m_22^2 + m_12^1 + m_02^0$$

จากสมการ จะได้ว่า

$$8C_x(k) = \sum_{m_0=0}^{m_0=1} \sum_{m_1=0}^{m_1=1} \sum_{m_2=0}^{m_2=1} X'(m_2, m_1, m_0) W^{k(4m_2 + 2m_1 + m_0)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \sum_{m_0} \sum_{m_1} \sum_{m_2} X'(m_2, m_1, m_0) W^{4km_2} W^{2km_1} W^{km_0}$$

ผลบวกในเขตของ m_2 แทนด้วยสัญลักษณ์ M_2 จะได้ว่า

$$M_2 = \sum_{m_2} X'(m_2, m_1, m_0) W^{4km_2}$$

แทนค่าของ k ในรูปของไบนารี และ $W^4 = -1$ เราจะได้ว่า

$$M_2 = \sum_{m_2} X'(m_2, m_1, m_0) (-1)^{m_2(4k_2 + 2k_1 + k_0)}$$

เพราะว่า $(-1)^{m_2(4k_2 + 2k_1 + k_0)} = 1$ M_2 สามารถแสดงในรูปของ

$$M_2 = \sum_{m_2} X'(m_2, m_1, m_0) (-1)^{k_0 m_2}$$

จากสมการผลบวกบน m_2 แสดงค่าที่รวมได้ในรูปฟังก์ชันของ k_0, m_1, m_0

$$M_2 = \sum_{m_2} X'(m_2, m_1, m_0) (-1)^{k_0 m_2} = X'(k_0, m_1, m_0)$$

แทนค่าสมการจะได้

$$BC_x(k) = \sum_{m_0} \sum_{m_1} X'(k_0, m_1, m_0) W^{2km_1} W^{km_0}$$

ทำอีกครั้งและแทนด้วยสัญลักษณ์ M_1

$$M_1 = \sum_{m_1} X'(k_0, m_1, m_0) W^{2km_1}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \sum_{m_1} X'(k_0, m_1, m_0) (-1)^{4k_2 + 2k_1 + k_0} m_2$$

ในสมการ จำนวน $(-1)^{4k_2 + 2k_1 + k_0} m_1 = 1$ จะได้รูปแบบที่ง่ายขึ้นดังนี้

$$M_1 = \sum_{m_1} X'(k_0, m_1, m_0) (-i)^{2k_1 + k_0} m_1$$

ผลบวก บน m_1 จะได้ผลลัพธ์ M_1 ที่เป็นฟังก์ชันของ k_0, m_1, m_0

$$M_1 = X_2'(k_0, k_1, m_0)$$

แทน m_1 ที่ได้ ในสมการจะได้

$$8C_x(k) = \sum_{m_0} X_2'(k_0, k_1, m_0) W^{m_0 k}$$

เหมือนกับ M_2 และ M_1 เราแทนสัญลักษณ์ผลบวกบน m_0 ด้วย M_0

$$M_0 = \sum_{m_0} X_2'(k_0, k_1, m_0) [(1-i)/\sqrt{2}]^{4k_2 + 2k_1 + k_0} m_0$$

การทำผลบวกบน m_0 จะได้ M_0 ที่เป็นฟังก์ชันของ k_0, k_1, k_2 ดังนี้

$$M_0 = X_3'(k_0, k_1, k_2)$$

$$8C_x(k) = 8C_x'(k_2, k_1, k_0) = X_3'(k_0, k_1, k_2)$$

ขั้นแรกในการหาคำตอบพิจารณา

$$1. N = 8 \text{ ทำให้ผล } \log_2 N = 3$$

2. coefficient $(-1), (-i), [(1-i)/\sqrt{2}]$ มีความสัมพันธ์กับรากที่

2, 4, 8 ตามลำดับ ของ unity $e^{-12\pi}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจึงกำหนดสัญลักษณ์ใหม่ ให้

$$A_2 = e^{-j2\pi/2^r}, r = 1, 2, \dots, \log_2 N$$

เมื่อ A_2 เป็นรากที่ 2^r ของ unity โดยมีคุณสมบัติดังนี้

$$1. A_2 = w^{N/2}, W = e^{-j2\pi/n}$$

$$2. (A_2)^{+1} = -(A_2)^1$$

$$; r = 1, 2, \dots, \log_2 N, l = 0, 1, \dots, 2^{r-1}$$

$$3. (A_2)^{N/2} = -1$$

จากสมการข้างต้นสามารถเขียนในเทอม A_2 จะได้

$$X_L(k_0, m_1, m_0) = \sum_{m_2} X'(m_2, m_1, m_0) A_2^{k_0 m_1} O^{m_2}$$

ตัวแปร k_0 มีค่า 0 หรือ 1 ทำให้ได้สมการ 4 สมการในแต่ละค่า k_0

กรณี 1 : $k_0 = 0$

$$X_L(0, 0, 0) = X(0, 0, 0) + X(1, 0, 0) \implies X_L(0) = X(0) + X(4)$$

$$X_L(0, 0, 1) = X(0, 0, 1) + X(1, 0, 1) \implies X_L(1) = X(1) + X(5)$$

$$X_L(0, 1, 0) = X(0, 1, 0) + X(1, 1, 0) \implies X_L(2) = X(2) + X(6)$$

$$X_L(0, 1, 1) = X(0, 1, 1) + X(1, 1, 1) \implies X_L(3) = X(3) + X(7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณี 2 : $k_o = 1$

$$X_1(1,0,0) = X(0,0,0)+X(1,0,0)\implies X_1(4) = X(0)-X(4)$$

$$X_1(1,0,1) = X(0,0,1)+X(1,0,1)\implies X_1(5) = X(1)-X(5)$$

$$X_1(1,1,0) = X(0,1,0)+X(1,1,0)\implies X_1(6) = X(2)-X(6)$$

$$X_1(1,1,1) = X(0,1,1)+X(1,1,1)\implies X_1(7) = X(3)-X(7)$$

ในเทอมของ A_4

$$X_2'(k_o, k_1, k_2) = X_1'(k_o, 0, m_o) + X_1'(k_o, 1, m_o) A_4^{(2k+k)}$$

กรณี 1 : $(k_1, k_o) = (0, 0)$

$$X_2'(0, 0, 0) = X_1'(0, 0, 0) + X_1'(0, 1, 0) \implies X_2(0) = X_1(0) + X_1(2)$$

$$X_2(0, 0, 1) = X_1(0, 0, 1) + X_1(0, 1, 1) \implies X_2(0) = X_1(1) + X_1(3)$$

กรณี 2 : $(k_1, k_o) = (0, 1)$

$$X_2(1, 0, 0) = X_1(1, 0, 0) + A_4 X_1(1, 1, 0) \implies X_2(4) = X_1(4) + A_4 X_1(6)$$

$$X_2(1, 0, 1) = X_1(1, 0, 1) + A_4 X_1(1, 1, 1) \implies X_2(1) = X_1(5) + A_4 X_1(7)$$

กรณี 3 : $(k_1, k_o) = (1, 0)$

$$X_2(0, 1, 0) = X_1(0, 0, 0) + A_4 X_1(0, 1, 0) \implies X_2(2) = X_1(0) - X_1(2)$$

$$X_2(0, 1, 1) = X_1(0, 0, 1) + A_4 X_1(0, 1, 1) \implies X_2(3) = X_1(1) - X_1(3)$$

กรณี 4 : $(k_1, k_o) = (1, 1)$

$$X_2(1, 1, 0) = X_1(1, 0, 0) + A_4 X_1(1, 1, 0) \implies X_2(6) = X_1(4) - A_4 X_1(6)$$

$$X_2(1, 1, 1) = X_1(1, 0, 1) + A_4 X_1(1, 1, 1) \implies X_2(7) = X_1(5) - A_4 X_1(7)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เขียนให้อยู่ในรูป $A_{\text{๒}}$ จะได้

$$X_2(k_o, k_t, k_z) = X_2(k_o, k_t, 0) + (k_o, k_t, 1)A_{\text{๒}}^{(4k+2k+k)}$$

กรณี 1 : $(k_2, k_1, k_0) = (0, 0, 0)$

$$X_3(0, 0, 0) = X_2(0, 0, 0) + X_2(0, 0, 1) \implies X_3(0) = X_2(0) + X_2(1)$$

กรณี 2 : $(k_2, k_1, k_0) = (0, 0, 1)$

$$X_3(1, 0, 0) = X_2(1, 0, 0) + A_{\text{๒}} X_2(1, 0, 1) \implies X_3(4) = X_2(4) + A_{\text{๒}} X_2(5)$$

กรณี 3 : $(k_2, k_1, k_0) = (0, 1, 0)$

$$X_3(0, 1, 0) = X_2(0, 1, 0) + A_{\text{๒}} X_2(0, 1, 1) \implies X_3(2) = X_2(2) + A_{\text{๒}} X_2(3)$$

กรณี 4 : $(k_2, k_1, k_0) = (0, 1, 1)$

$$X_3(1, 1, 0) = X_2(1, 1, 0) + A_{\text{๒}} X_2(1, 1, 1) \implies X_3(6) = X_2(6) + A_{\text{๒}} X_2(7)$$

กรณี 5 : $(k_2, k_1, k_0) = (1, 0, 0)$

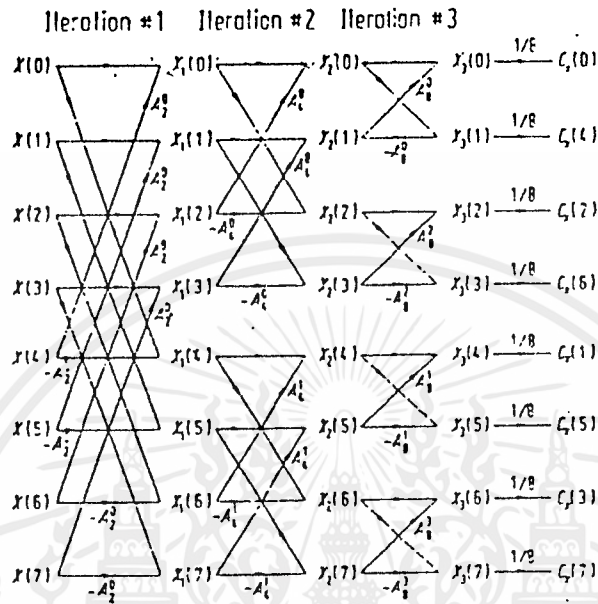
$$X_3(0, 0, 1) = X_2(0, 0, 0) + A_{\text{๒}} X_2(0, 0, 1) \implies X_3(1) = X_2(0) - X_2(1)$$

ลำดับการกระทำทางคณิตศาสตร์ตั้งแต่สมการที่เขียนในเทอมของ $A_{\text{๒}}$ ได้เขียนแสดงไว้ดังรูป 2.3.1 และการทำซ้ำ (iteration) ครั้งที่ 3 ก็คือ $n=3$ ในสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อให้ได้ค่า $C_x(k)$ ตามที่ต้องการ ใช้สมการ

$$8C_x(k_2, k_1, k_0) = X_3(k_0, k_1, k_2)$$



รูปที่ 2.3.1 แสดงแผนภาพการไหล (signal flow-graph) ของ FFT เมื่อ $N=8$ จะเห็นว่ามีการสลับตำแหน่งของสัมประสิทธิ์ไบนารี k_0, k_1 และ k_2 ใน X_3 เมื่อเทียบกับตำแหน่งจริง $C_x(k)$ เราเรียกลักษณะเช่นนี้ ว่าการกลับบิต (bit reversal) X_3 มีความสัมพันธ์กับ $C_x(k)$ ตามสมการข้างล่างนี้

$$X_3(0) = 8C_x(0)$$

$$X_3(4) = 8C_x(1)$$

$$X_3(2) = 8C_x(2)$$

$$X_3(6) = 8C_x(3)$$

$$X_3(1) = 8C_x(4)$$

ผลลัพธ์ของสัมประสิทธิ์ $C_x(k)$, $k = 5, 6, 7$ เราหาได้จาก

$$C_x(4+i) = C_x(4-i), \quad i = 1, 2, 3$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้

$$C_x(5) = C_x(3)$$

$$C_x(6) = C_x(2)$$

$$C_x(7) = C_x(1)$$

$C_x(5), C_x(6), C_x(7)$ สามารถหาได้จากสมการ และ

$$8C_x(5) = X_3(5) = X_2(4) - A_{32}X_2(5)$$

$$8C_x(6) = X_3(3) = X_2(2) - A_{32}X_2(3)$$

$$8C_x(7) = X_3(7) = X_2(6) - A_{32}X_2(7)$$

จากขั้นตอนข้างต้นเป็นขั้นตอนและแนวทางในการเขียน โปรแกรม FFT

2.3.2 การประยุกต์ใช้งาน FFT

ความละเอียด(resolution)ของ FFT

ผลลัพธ์ของ FFT ในรูป 2.3.2 (a) และ (b) แต่ละความถี่ มีระยะห่าง

$f_0 = 1/NT$ ดังนั้นจุดสุ่มในโดเมนความถี่จึงมีค่าตั้ง $0/NT, 1/NT, 2/NT, \dots, (N/2)/NT$

สำหรับด้าน ความถี่บวก ระยะห่างของความถี่ $f_0 = 1/NT$ เป็นเทอมที่แสดงถึงความ

ละเอียดของFFTแต่ละสมาชิกของความถี่ที่ได้เรียกว่า สมาชิกความละเอียด(resolution

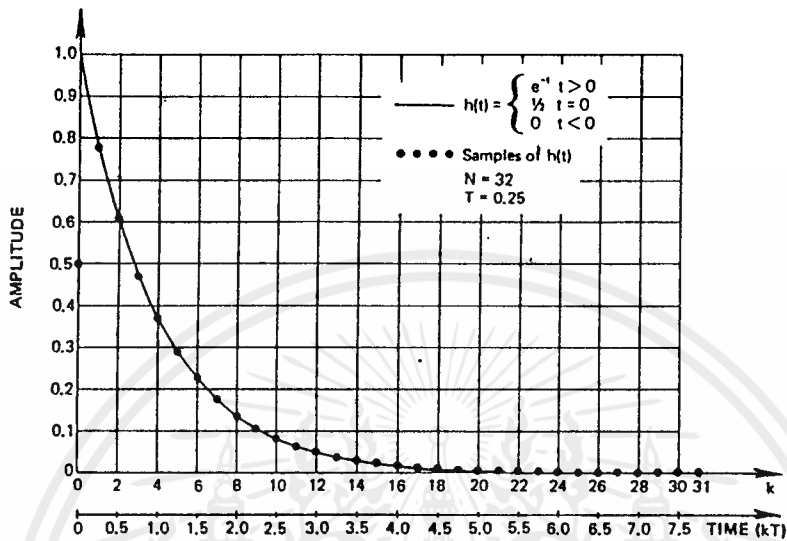
element)หรือ เซลความละเอียด (resolution cell)เราอาจจะคิดว่าความละเอียด

เป็นเทอมที่ทำให้เราสามารถ แยกความแตกต่างของแต่ละความถี่ โดยประมาณค่าความถี่ที่

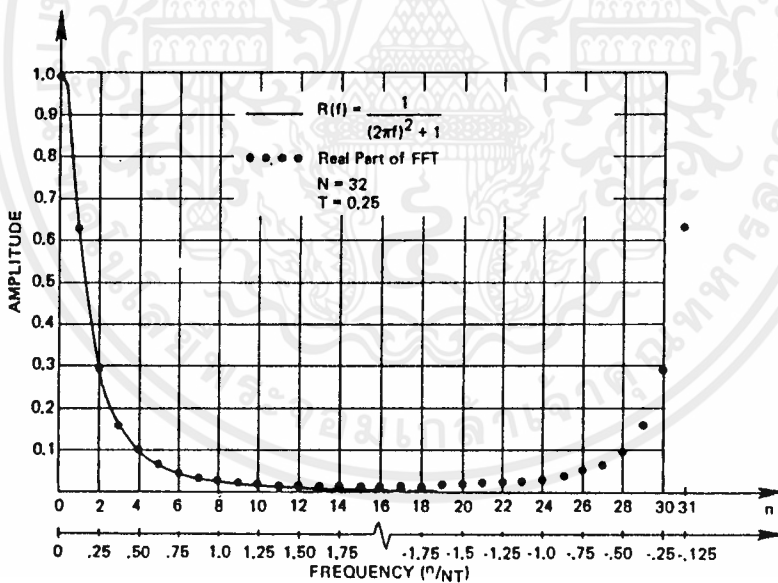
ได้จากการแปลง เป็นความถี่ตั้งแต่ $0/NT, 1/NT, 2/NT, \dots, (N/2)/NT$ ซึ่งเราอาจจะ

ลดระยะห่างระหว่างแต่ละความถี่ได้ (truncation) ของฟังก์ชันที่ต้องการแปลง ถ้าค่า

ของ N เพิ่มขึ้นเป็นสองเท่า ระยะห่างของความถี่ลดลงสองเท่าเช่นกัน



(a)

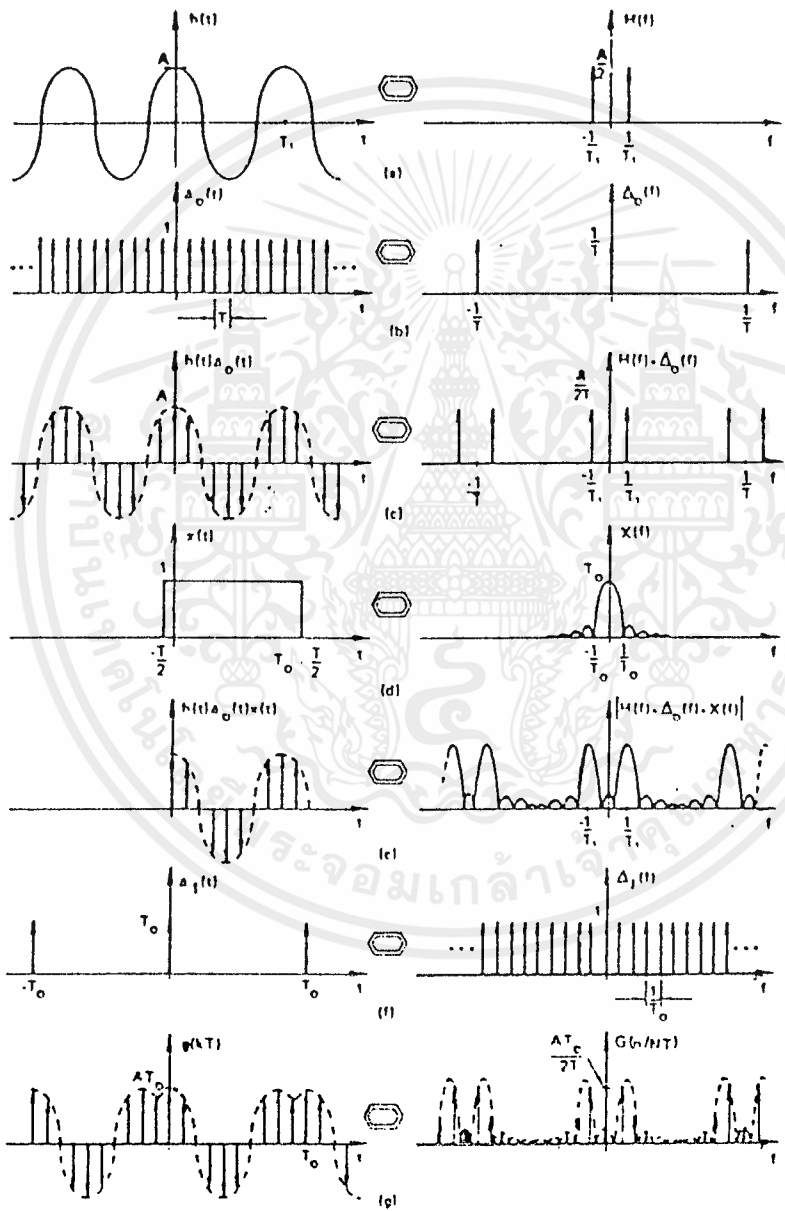


(b)

รูปที่ 2.3.2 ตัวอย่างการแปลงฟูเรียร์โดยวิธี FFT

จากรูปที่ 2.3.3 ระยะห่างของความถี่ (ความละเอียด) ของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ถูกกำหนดจากความกว้างของสี่เหลี่ยมที่ไปคนหรือตัดทอนฟังก์ชันก่อนทำการแปลงการตัดทอนในโดเมนเวลาก็คือการทำคอนโวลูชันของฟังก์ชัน เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$[\sin(f)/f]$ กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้น การทำคอนโวลูชันกับฟังก์ชัน $[\sin(f)/f]$ ทำให้ผลจากการแปลงคลุ่มเคลื่อนและเลื่อนลง ถ้าฟังก์ชันที่ตัดทอนในโดเมนเวลา มีความกว้างมากขึ้น จะทำให้ผลของฟังก์ชัน $[\sin(f)/f]$ แคบลงและความคลุ่มเคลื่อนของความถี่ลดลงด้วย ความคลุ่มเคลื่อนของความถี่ยิ่งน้อยเท่าใด ก็ยิ่งทำให้ประสิทธิภาพของการใช้ FFT ในการแก้ปัญหาในแต่ละงาน เป็นไปได้มากยิ่งขึ้น

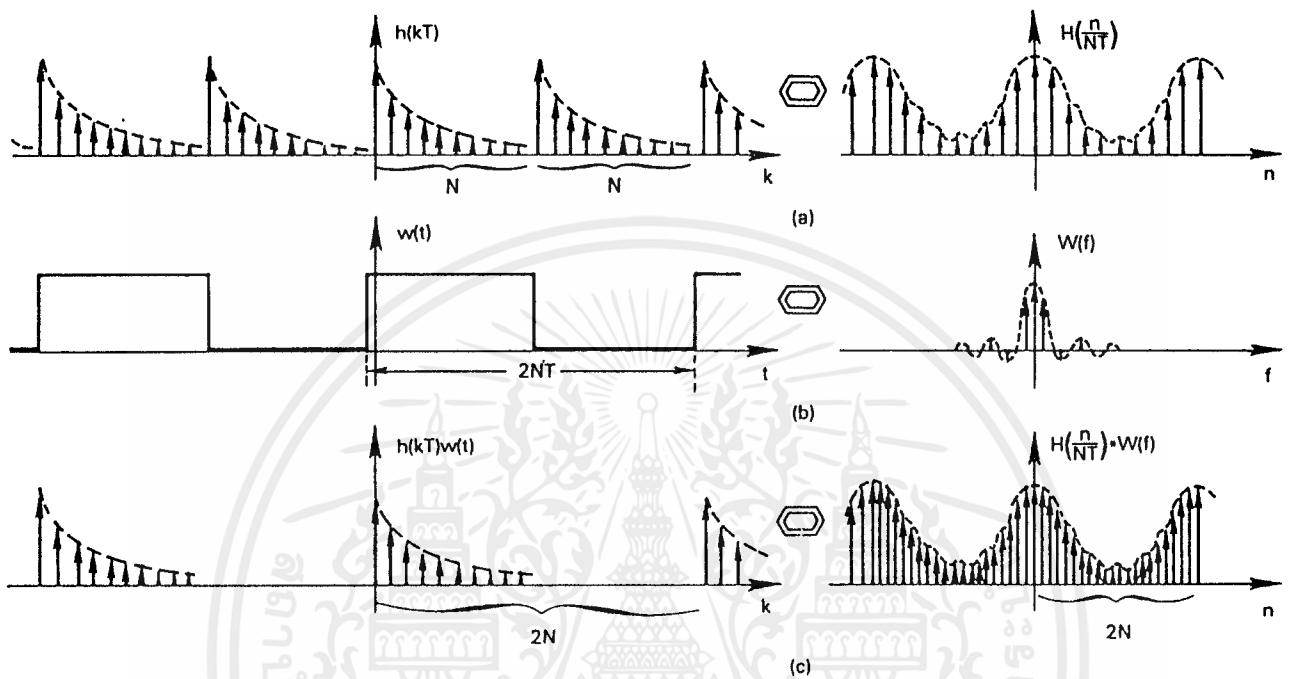


รูปที่ 2.3.3 แสดง DFT ของสัญญาณคาบเวลา โดยมีช่วงการตัดทอนน้อยกว่า 1 คาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเข้าใจผิดโดยทั่วไปของผู้ใช้ FFT คือการเพิ่มจำนวนข้อมูล N โดยการต่อเพิ่ม ศูนย์และตัดทอนฟังก์ชัน และตีความหมายของฟังก์ชันความถี่ที่ได้เป็นฟังก์ชันความถี่ที่มีความละเอียดสูง ตอนนี้นำมาเปรียบเทียบกับรูปที่ 2.3.3 กับรูปที่ 2.3.4 ในรูปที่ 2.3.4(a) แสดงผลจากการแปลงดิสครีตฟูเรียร์ เช่นเดียวกับรูปที่ 2.3.3 (๕) เราต้องการตรวจสอบผลของการเพิ่มศูนย์ให้กับฟังก์ชันเวลาของรูปที่ 2.3.4 (a) สมมติว่าจำนวนของศูนย์ที่เพิ่มเข้าไปนั้นเท่ากับ M ซึ่งทำได้โดยการทำด้วยฟังก์ชันคาบเวลา ในโดเมนเวลาในรูปที่ 2.3.4(b) ผลของความถี่ที่ได้ แสดงในรูปถัดมาการคูณทำให้ฟังก์ชันมีคาบเวลาเท่ากับ $2M$ โดยที่จุดที่ไม่ใช่ศูนย์เท่ากับจุดสุ่ม N จำนวนในรูปที่(a) ผลของฟังก์ชันความถี่ที่ได้คือ การทำคอนโวลูชันของฟังก์ชันความถี่ของ รูปที่ (a) และรูปที่ (b) แต่ความละเอียดของความถี่ได้ ถูกกำหนดไว้ใน รูปที่ (a) เรียบร้อยแล้วการทำคอนโวลูชันเป็นการเพิ่มจำนวนจุดสุ่มในโดเมนความถี่โดยการเฉลี่ยค่า (interpolation) ด้วยฟังก์ชัน $\text{sinc}(f)/f$ กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้น ดังนั้น ถึงแม้ว่าระยะห่างของผลลัพธ์จาก FFT จะมีระยะใกล้ขึ้น โดยการเติมศูนย์ก็ตาม แต่ความละเอียดก็ไม่ได้เปลี่ยนแปลง ความละเอียดของ FFT ไม่สามารถ ทำให้ความละเอียดของข้อมูลเพิ่มขึ้น โดยการต่อเติมศูนย์ นอกเสียจากว่าฟังก์ชันที่จะทำการต่อเติมศูนย์มีค่าศูนย์ครอบคลุมช่วงนั้นอยู่แล้ว

จากหัวข้อนี้ เราอาจจะสรุปได้ว่า ความละเอียดนั้นต้องพิจารณาจากช่วงเวลาในโดเมนเวลาของสัญญาณ และในการประยุกต์ใช้ FFT ช่วงเวลาของสัญญาณถูกตั้งใหม่เป็นช่วงระยะห่างของการตัดทอนข้อมูล



รูปที่ 2.3.4 ตัวอย่างการเพิ่มความละเอียดของ FFT โดยการเติมศูนย์

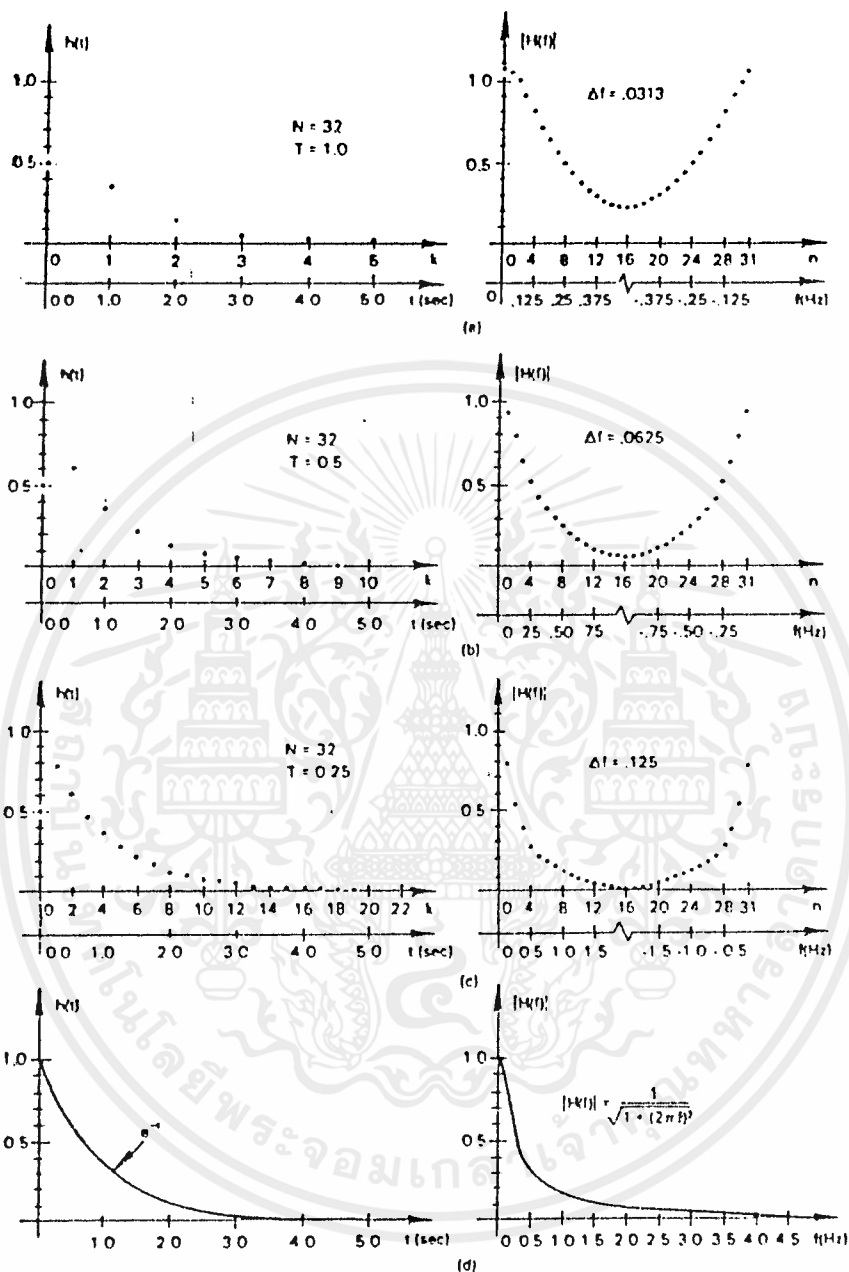
2.3.3 ผลความผิดพลาดจาก FFT (FFT aliasing)

ปัญหาหนึ่งที่เราพบ ในการคำนวณการแปลงฟูเรียร์จาก FFT คือ ความผิดเพี้ยน (aliasing) จากที่กล่าวมาแล้ว ในการสุ่มข้อมูล ความผิดเพี้ยนจะเกิดขึ้นเมื่อจุดสุ่มสัญญาณของฟังก์ชันเวลา มีระยะห่างของการเก็บมากเกินไป ผลของฟังก์ชันความถี่ที่ผิดเพี้ยนนี้เรียกว่า การพับ (fold) หรือการซ้อน (overlap) ในตัวมันเอง ซึ่งได้แสดงตัวอย่างไว้ใน

รูปที่ 2.3.5

ในรูปที่ 2.3.5 เรามีจุดสุ่มของฟังก์ชัน $h(t) = e^{-t}, t > 0$ โดยที่ช่วงเวลาการสุ่ม $T = 1.0, 0.5$, และ 0.25 ตามลำดับ จำนวนข้อมูลตั้งให้เท่ากับ 32 สำหรับแต่ละกรณีขนาดของการแปลงฟูเรียร์ที่ได้ คำนวณโดยใช้ FFT ซึ่งได้แสดงรวมไว้ในรูปที่ 2.3.5 (a) ถึง (c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

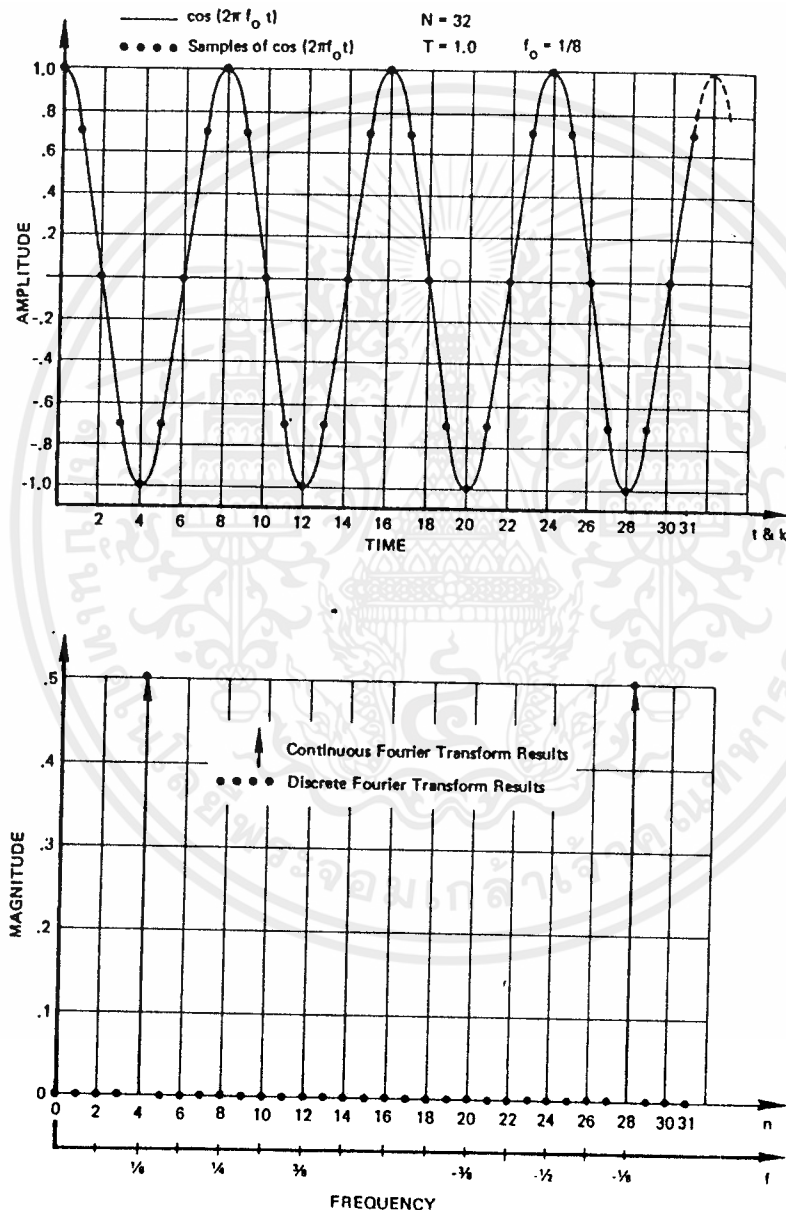


รูปที่ 2.3.5 แสดงตัวอย่างของความผิดเพี้ยนในโดเมนความถี่จากอัตราการสุ่ม

แล้ว จากผลที่ได้จาก FFT จะเห็นว่าเมื่อ $T = 1.0$ ผลที่ได้ความผิดเพี้ยนมากที่สุด (ขนาดของการแปลงฟูเรียร์แบบต่อเนื่องได้แสดงไว้ในรูปที่ (d)) จะเห็นว่าความผิดเพี้ยนมีค่าลดลงเมื่อมีคาบเวลาเท่ากับ 0.5 และเมื่อคาบเวลาเท่ากับ 0.25 ผลที่ได้จะมีความคล้ายคลึงกับผลที่ได้จากการคำนวณแบบต่อเนื่องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.3.5 ได้แสดงหลักการลดความผิดเพี้ยน โดยการลดช่วงการสุ่ม ในกรณีนี้จะเห็นว่าไม่มีผลของการตัดทอน เนื่องจาก T มีค่าต่ำจึงทำให้ $N T$ มีค่ามากกว่า ความกว้างของช่วงไม่เป็นศูนย์ (nonzero interval) ของฟังก์ชันมาก



รูปที่ 2.3.6 แสดงการตัดทอนในโดเมนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.4 การตัดทอนของ FFT ในโดเมนเวลา (FFT Time-domain Truncation)

ปัญหาความผิดพลาดอื่นที่มักพบจากการประยุกต์ใช้ FFT ในการแปลงฟูรีเยร์คือความผิดพลาดเนื่องจากการตัดทอนข้อมูลในโดเมนเวลา (time-domain truncation) ความผิดพลาดนี้เกิดขึ้นจากการสุ่มข้อมูลเพื่อหาลักษณะของฟังก์ชันเวลาไปตัดทอน ลักษณะบางส่วนของรูปคลื่นเริ่มต้น รูปที่ 2.3.6 (a) และ รูป (b) แสดงตัวอย่างของจุดนี้ เมื่อเราตัดทอนฟังก์ชัน $h(t)$ ที่ $NT = 1, 2, 5$ s. ตามลำดับ ขนาดของการแปลงฟูรีเยร์ที่ผ่าน FFT ได้แสดงในรูป (c)

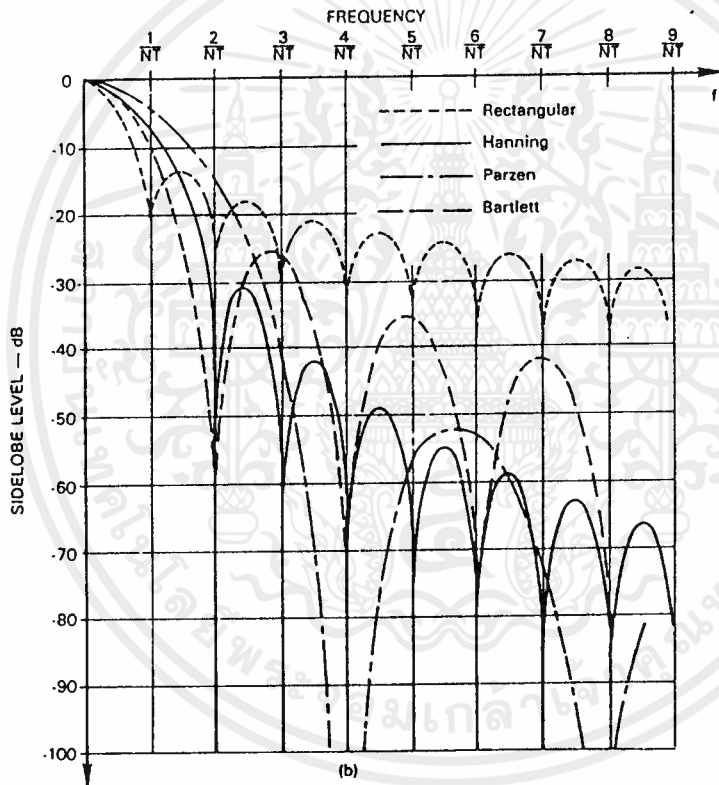
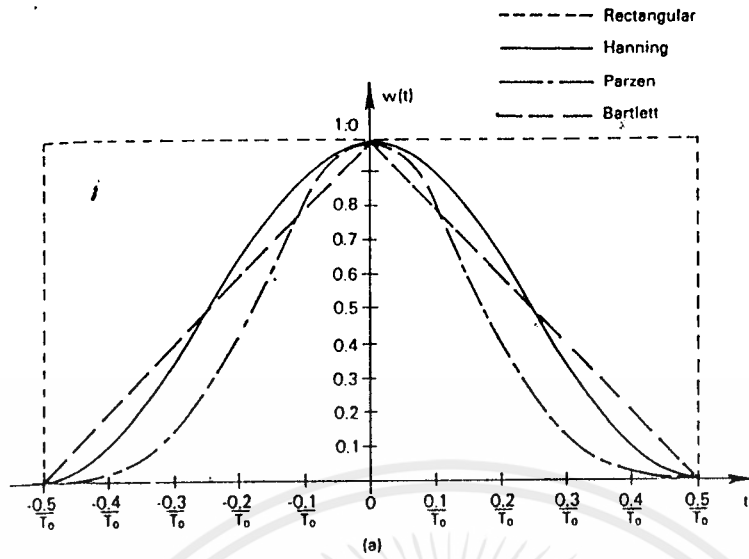
การตัดทอนที่ 1 วินาที ทำให้ผลที่ได้จาก FFT มีการกระเพื่อม (rippling) เมื่อใช้ช่วงเวลาของการตัดทอนนานขึ้น เท่ากับ 2 วินาทีผลของ FFT ที่ได้มีการกระเพื่อมน้อยลง และเมื่อเพิ่มช่วงเวลาการตัดทอนให้เป็น 5 วินาที ผลของ FFT ที่ได้ไม่ปรากฏผลของการเกิดการกระเพื่อม เหมือนกับรูปที่ถูกต้องในรูปที่

รูปที่ 2.3.6 แสดงผลของการทดลอง โดยการพิจารณาหาช่วงเวลาการตัดทอนที่ดีที่สุดโดยการเพิ่มเวลาการตัดทอน ซึ่งเราก็ได้เห็นผลของการทดลองของการกระเพื่อม

2.3.5 FFT ของฟังก์ชันคาบเวลา

การคำนวณหา FFT ของฟังก์ชันคาบเวลา เราต้องเลือกช่วงการสุ่มและช่วงการตัดทอน จากที่กล่าวมาแล้ว การเลือก T ต้องเลือกให้เหมาะสมเพื่อไม่ให้เกิดความผิดพลาดเนื่องจากการตัดทอนของฟังก์ชันเวลาคาบเป็นปัญหาใหม่ที่ไม่ได้อธิบายไว้ในหัวข้ออื่นมาก่อน

ถ้าเราเลือกจำนวนจุดสุ่ม N ของการแปลงฟูรีเยร์ เท่ากับ หนึ่งคาบเวลาพอดี หรือ เท่ากับจำนวนเท่าๆกับจำนวนเท่าของคาบเวลา ผลของความถี่ที่ได้จะตรงกับสัญญาณเริ่มต้น



รูปที่ 2.3.7 แสดงผลจาก FFT ของฟังก์ชันคาบเวลาโดยมีช่วงเวลากการตัดทอน

เท่ากับจำนวนเท่ากับจำนวนเท่าของคาบเวลา

เพื่ออธิบายจุดนี้ เราทำการคำนวณ FFT ของฟังก์ชันโคไซน์ในรูปที่ 2.3.7 (a) โดยมีช่วงการสุ่ม $T = 1.0$ s และจำนวนจุดสุ่มเท่ากับ 32 และแสดงผลที่ได้ในรูปที่ (b) โดยแสดงเป็นขนาดที่ได้จาก FFT ของจุดเหล่านี้ จากรูปจะเห็นว่า จุดอื่นเป็นศูนย์หมด นอกจากจุดที่เป็นความจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.6 ฟังก์ชันถ่วงน้ำหนักข้อมูล (data weighting function)

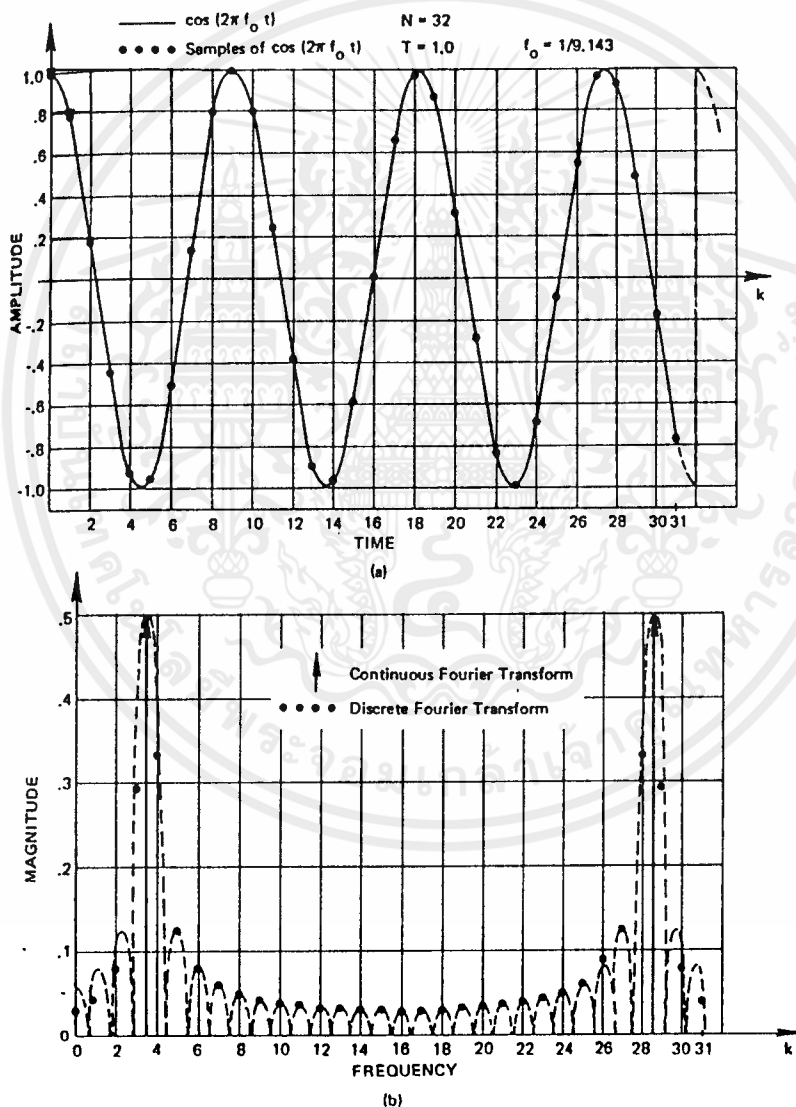
จากหัวข้อที่แล้ว การตัดทอน (truncation) อาจจะทำให้ผลจากการแปลงฟูเรียร์คลาดเคลื่อน การประมวลผลข้อมูลจำนวนจำกัด N ของฟังก์ชันคาบเวลาที่ไม่รู้คาบเวลานั้น จำเป็นต้องใช้วินโดว์ของข้อมูลหรือที่เรียกว่าฟังก์ชันถ่วงน้ำหนักข้อมูล ในหัวข้อนี้ เราจะได้อธิบายเทคนิค การใช้ฟังก์ชันถ่วงน้ำหนักเพื่อลดผลเสีย ที่เกิดจากการตัดทอนให้เหลือน้อยที่สุด

2.3.7 ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม (rectangular weighting function)

จากรูปที่ 2.3.3 เราสรุปสัญญาณชายน โดยการคูณด้วย อิมพัลส์ฟังก์ชัน (impulse function) ซึ่งได้แสดงไว้ในรูปที่ (a) นำผลที่ได้ (รูปที่ (c)) ไปคูณด้วยฟังก์ชันตัดทอนสี่เหลี่ยม ซึ่งแสดงไว้ในรูปที่ (d) เพื่อจำกัดจำนวนของจุดสุ่มให้เหลือเท่ากับ N ในที่นี้การตัดทอนของโดเมนเวลา (time domain) ก็คือการถ่วงน้ำหนักข้อมูลโดยคูณด้วยฟังก์ชันสี่เหลี่ยม ผลของการตัดทอนโดเมนเวลา ได้แสดงไว้ในรูปที่ (e) เราเห็นว่าฟังก์ชันโดเมนความถี่ (frequency domain) ของฟังก์ชันสี่เหลี่ยม คือฟังก์ชัน $[\sin(f)/f]$ ดังนั้นการแปลง เป็นโดเมนความถี่ ของฟังก์ชันเวลาที่ตัดทอนแล้ว จึงเป็นการทำ คอนโวลูชัน (convolution) ของฟังก์ชันเวลากับฟังก์ชันสี่เหลี่ยมทำให้ผลที่ได้มีองค์ประกอบของความถี่เพิ่มขึ้นมา ซึ่งก็คือลอนข้าง (side lobe) ของฟังก์ชันสี่เหลี่ยม องค์ประกอบที่เพิ่มขึ้นมานี้เรียกว่าส่วนรั่วไหล (leakage) ทำให้อิมพัลส์ของความถี่ของฟังก์ชันรั่วออกไปที่ลอนข้างของฟังก์ชัน $[\sin(f)/f]$

ถึงแม้ว่าฟังก์ชันในโดเมนเวลาเริ่มต้นเป็นสัญญาณชายน แต่เมื่อผ่านการสุ่มแล้ว รูปคลื่นของสัญญาณที่ถูกสุ่มในโดเมนเวลา จะไม่เป็นสัญญาณชายนอีก เพราะว่าช่วงของการตัดทอน (truncation interval) ไม่เท่ากับคาบเวลาหรือจำนวนเท่าของคาบเวลา (เป็นจำนวนเต็ม) ทำให้คอนโวลูชันในโดเมนเวลาในรูปที่ (e) และ (f) ไม่ตรงกับฟังก์ชันคาบเวลาเริ่มต้น เนื่องจากการทำคอนโวลูชันของฟังก์ชันเวลาเป็นแบบไม่ต่อเนื่อง (discontinuous) จึงได้แสดงผลของการเกิดการกระเพื่อม (rippling effect) ไว้ในแบบไม่ต่อเนื่องในรูปที่ (g) ด้วย

ต่อไป เราสาธิตตัวอย่างของผลที่เกิดจากการใช้ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมดังนี้ สมมติว่า เราต้องคำนวณ FFT ของฟังก์ชันโคไซน์ (cosine function) ซึ่งแสดงไว้ใน รูปที่ 2.3.8 (a) โดยมีคาบเวลา $T = 1.0$ และ $N = 32$ ในรูปที่ 2.3.8 (b) เรา ขนาดของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ของจุดสุ่มในรูปที่ 2.3.8 (a) FFT จะทำให้ผลเป็นความถี่ไม่ต่อเนื่องทางด้านขั้วทั้งหมด จากที่ได้อธิบาย ในหัวข้อที่แล้ว องค์ประกอบของความถี่เพิ่มขึ้นมาที่เรียกว่าส่วนรัวไหลเป็นผลมาจากลักษณะ การมีลอนข้างของฟังก์ชัน $[\sin(f)/f]$



รูปที่ 2.3.8 แสดงผลจาก FFT ของฟังก์ชันคาบเวลาโดยมีช่วงการตัดทอน ไม่เท่ากับจำนวนเท่าของคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อลดผลจากการรั่วไหล เราจำเป็นต้องใช้วิธีการตัดทอนโดเมนเวลาหรือฟังก์ชันด่วงน้ำหนักรที่มีลักษณะของลอนข้างในโดเมนความถี่เล็กกว่าของฟังก์ชันสี่เหลี่ยม ลอนข้างที่เล็กกว่านี้ทำให้การรั่วไหลของผลลัพธ์จาก FFT ลดน้อยลง เพื่อให้เข้าใจจุดนี้ได้ชัดเจนยิ่งขึ้น ขอให้พิจารณารูปที่ 2.3.2 อีกครั้งหนึ่ง ถ้าเปลี่ยนฟังก์ชันตัดทอนในรูปที่ (d) เป็นฟังก์ชันที่มีลอนข้างต่ำ จะทำให้การประมาณค่าของการแปลงฟูเรียร์ดีขึ้น การใช้ฟังก์ชันด่วงน้ำหนักรข้อมูลเพื่อตัดทอนและด่วงน้ำหนักรข้อมูลต้องทำกับจุดสุ่ม N ก่อนคำนวณ FFT

2.3.8 ลักษณะของฟังก์ชันด่วงน้ำหนักรชนิดต่าง ๆ

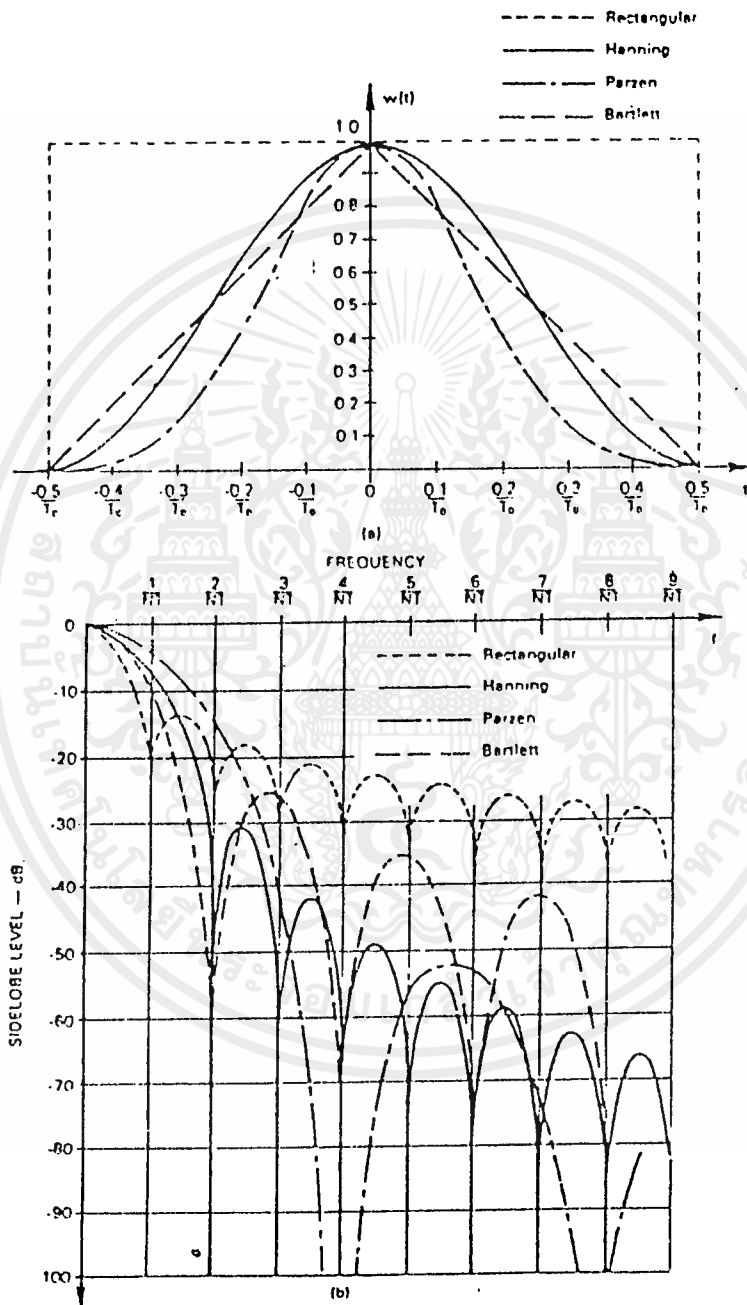
ในรูปที่ 2.3.9 (a) ได้แสดงฟังก์ชันตัดทอนหรือด่วงน้ำหนักรที่นิยมใช้กับ FFT ผลของการตอบสนองความถี่ของฟังก์ชัน ได้แสดงไว้ในรูปที่ (b) และในตารางที่ 2.3.1 แสดงข้อมูลการเปรียบเทียบลักษณะของฟังก์ชันด่วงน้ำหนักรแต่ละตัว ทั้งในโดเมนเวลา และโดเมนความถี่โดยมีจุดศูนย์กลางอยู่ที่จุดศูนย์เพื่อให้ง่ายต่อการสังเกต

จากรูปที่ 2.3.9 (a) จะเห็นว่า ฟังก์ชันด่วงน้ำหนักรทั้งหมด มีลอนข้างในโดเมนความถี่ที่มีขนาดเล็กกว่าฟังก์ชันด่วงน้ำหนักรสี่เหลี่ยมทั้งสิ้น ซึ่งจะช่วยให้ผลของการรั่วไหลลดน้อยลงแต่อย่างไรก็ตาม ฟังก์ชันด่วงน้ำหนักรทั้งหมดนี้มีลักษณะของลอนหลัก (main lobe) ที่กว้างกว่าลองดูรูปที่ 2.3.3 (d) และ (e) อีกครั้ง ลองคิดเปรียบเทียบผลที่ได้ เมื่อใช้ฟังก์ชันด่วงน้ำหนักรตามรูปที่ 2.3.9 (b) จะเห็นว่า ยิ่งลอนหลักมีความกว้างมากขึ้นเท่าใด ผลที่ได้จาก FFT ยิ่งมีความคลุมเคลือมากยิ่งขึ้น นั่นก็หมายความว่าถ้าลอนหลักมีขนาดกว้าง จะทำให้แยกความแตกต่างของแต่ละความถี่ได้ยาก

ข้อได้เปรียบเสียเปรียบระหว่างส่วนรั่วไหล (ระดับการรั่วไหล) และความละเอียด (แบนด์วิดธ์ของลอนหลัก) เป็นที่รู้จักกันดีในงานวิทยาศาสตร์สาขาต่าง ๆ ในตารางที่ 2.3.1 แสดงระดับสูงสุดของลอนข้างและแบนด์วิดธ์ (bandwidth) 3-dB สำหรับแต่ละฟังก์ชันด่วงน้ำหนักร สำหรับงานการทดลองทั่วไป นิยมใช้ฟังก์ชันแฮนนิ่ง (Hanning function) เพราะว่าการโครงสร้างของมันง่ายกว่าแบบอื่น การเลือกใช้ฟังก์ชันด่วงน้ำหนักรที่ให้ผลดีที่สุดต้องขึ้นกับลักษณะของงานที่เราจะไปใช้งานด้วยเป็นสำคัญ

ผลที่ได้จาก FFT จะมีช่วงความถี่ $f_0 = 1/NT$ ซึ่งอาจทำให้เกิดความสับสนกับค่า $1/NT$ ของฟังก์ชันด่วงน้ำหนักรที่เลือกใช้ผลของความละเอียดของความถี่ของผลลัพธ์ที่ได้จาก FFT เป็นเพียงฟังก์ชัน ของแบนด์วิดธ์ของฟังก์ชันด่วงน้ำหนักร แต่ละตัวเท่านั้น (ดูรูปที่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.9 (b)) ดังนั้นการใช้ค่านำมของความละเอียด $1/NT$ ต้องทำอย่างระมัดระวังและรำลึกเสมอว่ามันเป็นเพียงช่องว่าง ระหว่างความถี่ที่เกิดจากผลที่ได้จาก FFT เท่านั้น และไม่เกี่ยวกับ $1/NT$ ของวินโดว์ที่ใช้



รูปที่ 2.3.9 แสดงฟังก์ชันถ่วงน้ำหนักหรือวินโดว์ของ FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Weighting Function Nomenclature	Time Domain	Frequency Domain	Highest Side-Lobe Level (db)	3-dB Bandwidth	Asymptotic Roll-off (dB/Octave)
Rectangular	$w_r(t) = 1 \quad t \leq \frac{T_0}{2}$ $= 0 \quad t > \frac{T_0}{2}$	$W_r(f) = \frac{T_0 \sin(\pi f T_0)}{\pi f T_0}$	-13	$\frac{0.85}{T_0}$	6
Bartlett (triangle)	$w_b(t) = \left[1 - \frac{2 t }{T_0} \right] \quad t < \frac{T_0}{2}$ $= 0 \quad t > \frac{T_0}{2}$	$W_b(f) = \frac{T_0}{2} \left[\frac{\sin\left(\frac{\pi}{2} f T_0\right)}{\frac{\pi}{2} f T_0} \right]^2$	-26	$\frac{1.25}{T_0}$	12
Hanning (cosine)	$w_h(t) = \cos^2\left(\frac{\pi t}{T_0}\right)$ $= \frac{1}{2} \left[1 + \cos\left(\frac{2\pi t}{T_0}\right) \right] \quad t \leq \frac{T_0}{2}$ $= 0 \quad t > \frac{T_0}{2}$	$W_h(f) = \frac{T_0}{2} \frac{\sin^2(\pi f T_0)}{\pi^2 f^2 T_0^2 - (f T_0)^2}$	-32	$\frac{1.4}{T_0}$	18
Parzen	$w_p(t) = 1 - 24\left(\frac{t}{T_0}\right)^2 + 48\left \frac{t}{T_0}\right \quad t < \frac{T_0}{4}$ $= 2 \left[1 - \frac{2 t }{T_0} \right]^3 \quad \frac{T_0}{4} < t < \frac{T_0}{2}$ $= 0 \quad t \geq \frac{T_0}{2}$	$W_p(f) = \frac{3T_0}{8} \left[\frac{\sin^2(\pi f T_0/4)}{\pi^2 f^2 T_0^2/4} \right]^2$	-52	$\frac{1.82}{T_0}$	24

ตารางที่ 2.3.1 ฟังก์ชันถ่วงน้ำหนักข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไป เราจะแสดงผลจากการใช้ฟังก์ชันถ่วงน้ำหนักที่มีลอนข้างต่ำเพื่อให้เห็นลักษณะการนำไปใช้ลดการรั่วไหลของผลจากการตัดทอนในโดเมนเวลา ในรูปที่ 2.3.10 (a) เราได้แสดงรูปคลื่นโคไซน์ ที่แสดงไว้ในรูปที่ 2.3.8 (a) ที่คูณด้วยฟังก์ชันถ่วงน้ำหนักแชนนิ่งที่แสดงไว้ในรูปที่ 2.3.9 (a) แล้ว

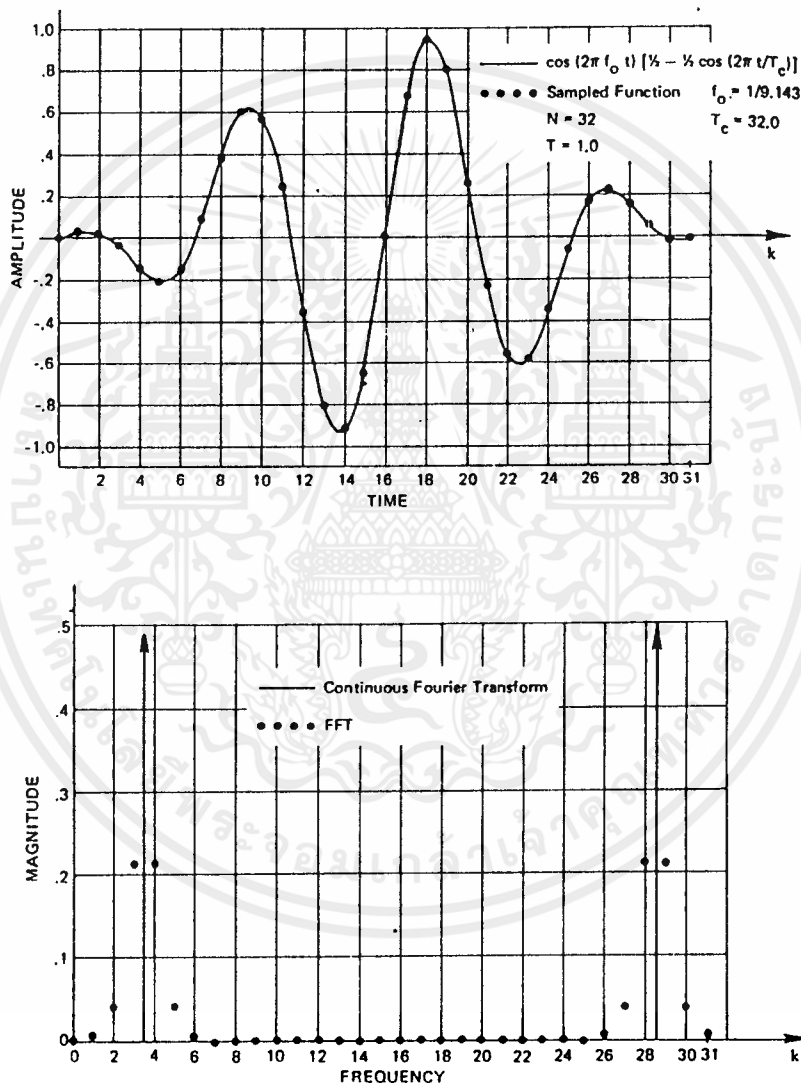
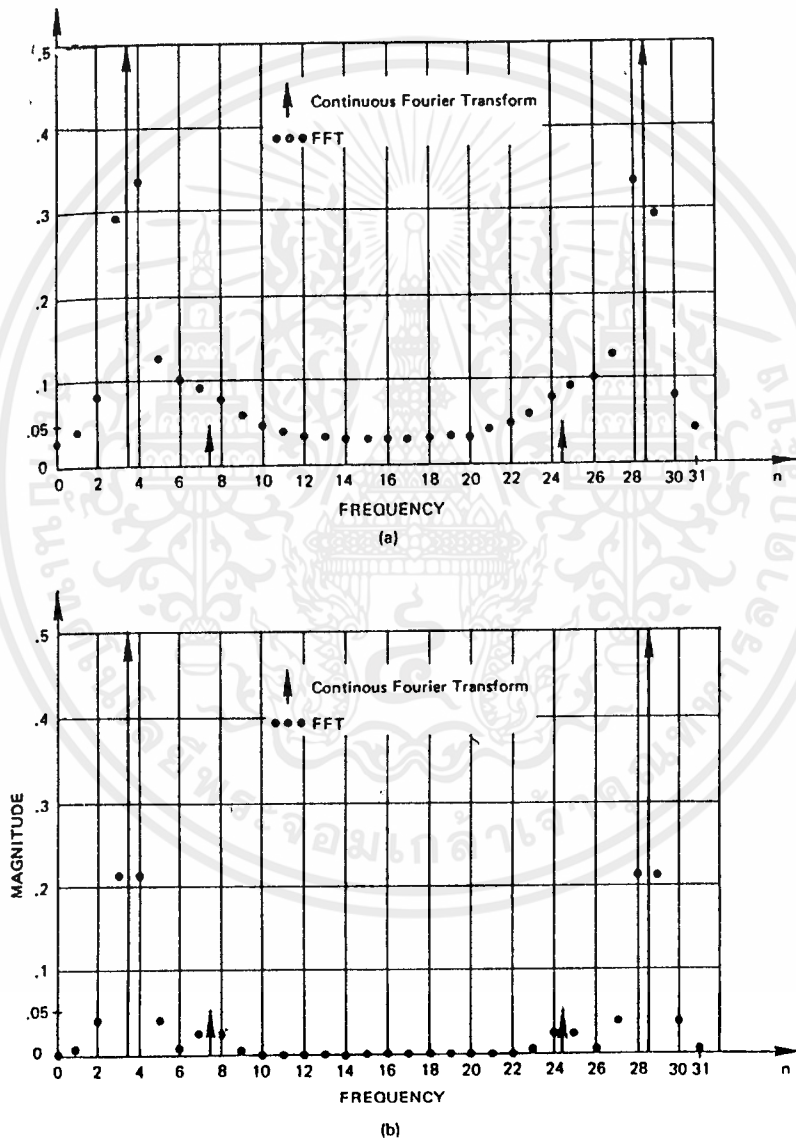


Figure 9.9 Example of applying the Hanning function to reduce leakage in computation of the FFT.

รูปที่ 2.3.10 แสดงตัวอย่างการใช้แชนนิ่งฟังก์ชันเพื่อลดส่วนรั่วไหลของการคำนวณด้วย FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.3.10 (b) แสดงจุดสุ่มจาก FFT ของรูปที่ 2.3.10 (a) จากรูปจะเห็นว่า ส่วนรบกวนลดลงไปอย่างมาก แต่องค์ประกอบหลักของความถี่มีระยะกว้างขึ้น หรือคลุมเคลือเมื่อเปรียบเทียบกับจุดของอิมพัลส์ฟังก์ชันเพราะผลจากการคอนโวลูชันความถี่ของอิมพัลส์ฟังก์ชันกับผลการแปลงฟูเรียร์ของฟังก์ชันถ่วงน้ำหนัก



รูปที่ 2.3.11 (a) แสดงตัวอย่างที่เลือนลางเนื่องจากผลจากส่วนรบกวนของลอนข้าง
 (b) สัญญาณที่ตรวจจ้งได้หลังจากผ่านแฮมมิงฟังก์ชันแล้ว

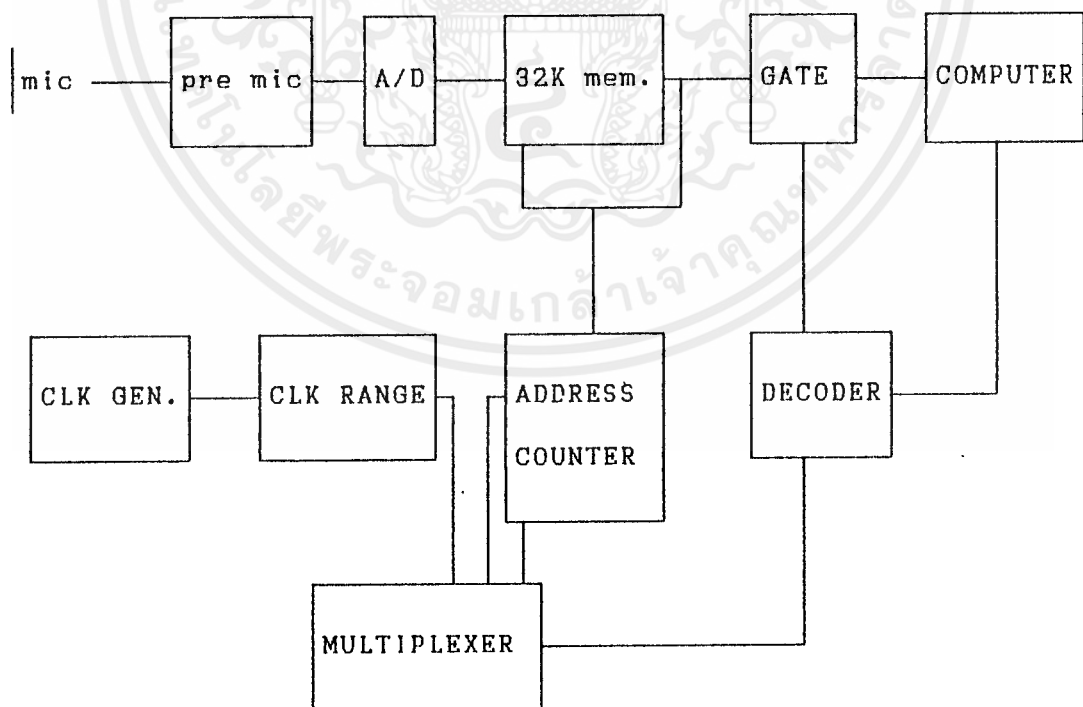
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 วงจรและการทำงาน

ส่วนของวงจรและการทำงานแบ่งเป็น 2 ส่วน คือ ส่วนวงจรอินเตอร์เฟสซึ่งแปลงสัญญาณเสียงเป็นสัญญาณดิจิทัล และส่วนประมวลผล

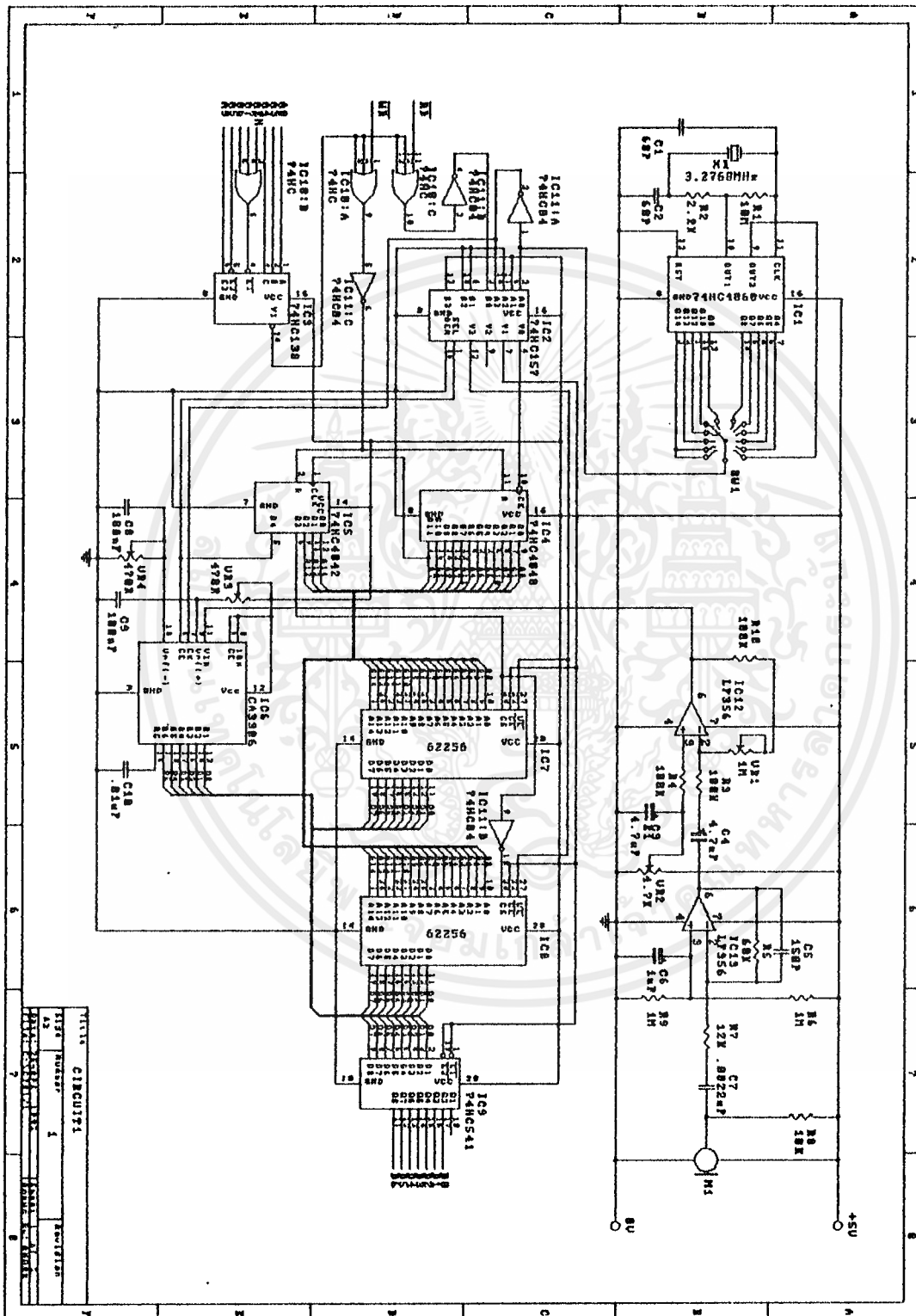
โครงงานในส่วนนี้จะกล่าวถึงส่วนวงจรอินเตอร์เฟสกับคอมพิวเตอร์ ซึ่งทำหน้าที่ส่งสัญญาณที่ได้เป็นสัญญาณดิจิทัลให้กับคอมพิวเตอร์ทำการประมวลผล และทำการพัฒนาโปรแกรมในส่วนนี้ก่อน ซึ่งต่อมาจะได้พัฒนาต่อไปกับส่วนประมวลผลที่สร้างขึ้น

ส่วนของวงจรอินเตอร์เฟสกับคอมพิวเตอร์นี้จะทำการแปลงสัญญาณเสียงที่เข้ามาเป็นสัญญาณดิจิทัลเก็บในหน่วยความจำ 64กิโลไบต์เพื่อให้การทำงานในส่วนนี้ของคอมพิวเตอร์ไม่ยุ่งยากในการเขียนโปรแกรมเมื่อเก็บสัญญาณเสียงเต็ม หน่วยความจำ คอมพิวเตอร์จะมาถึงข้อมูลไปทำการวิเคราะห์ โดยความถี่ในการแซมปลิ่งสามารถกำหนดทางฮาร์ดแวร์การทำงานดังบล็อกไดอะแกรม



รูปที่ 3.1 BLOCK DIAGRAM การทำงานวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรของการ์ดที่ใช้ร่วมกับเครื่อง PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 วงจรและการทำงานของการ์ด

การทำงานของการ์ดต้องใช้กับคอมพิวเตอร์ XT, AT เพราะเป็นการดึงข้อมูลหรือ DATA เพียง 7 bit การทำงานเป็นการสังรีเซต จากคอมพิวเตอร์ เพื่อรีเซตวงจร counter เริ่มรับเสียงเข้ามาเปลี่ยนเป็นสัญญาณ ดิจิตอล นำไปเก็บใน RAM เมื่อวงจร counter นับข้อมูลจนเต็มหน่วยความจำ ก็จะส่งสัญญาณไปที่ bit 7 ให้คอมพิวเตอร์มาสั่งอ่านว่า มีสัญญาณบอกกว่าเต็มหรือยังถ้าเต็มก็ให้อ่านข้อมูลออกไป

การทำงานอย่างละเอียด สัญญาณเสียงเข้ามาจากไมโครโฟน ผ่านวงจรมอนิเตอร์ที่ IC13 ผ่านวงจรรีบระดับแรงดันและปรับอัตราขยายสัญญาณ ที่ IC12 เพื่อให้ได้สัญญาณ 0.5-4.5V ในช่วง REF(+) 4.5-5V และ REF(-) 0-0.5 เพื่อให้ในช่วงการทำงานของ IC12 และ IC6 ADC ซึ่งการทำงานอาจคลาดเคลื่อนได้และเป็นการป้องกันสัญญาณรบกวนด้วย สัญญาณเสียงจาก IC12 ก็จะส่งต่อไปยัง IC6ADC ขนาด 6 bit แบบ flash ซึ่งทำให้การทำงานของวงจรง่ายขึ้น โดยไม่ต้องคำนึงถึง conversion time ซึ่งความเร็วสูงสุดที่ใช้งานได้ คือ 15 MHz ระดับ REF(+) ปรับที่ VR9 ระดับ REF(-) ปรับที่ VR4 วงจรในภาคอื่น IC1 เป็นวงจรถ่ายซึ่งจะได้สัญญาณ clock เป็นสัญญาณ sampling ให้กับวงจรถ่าย ในวงจรถ่ายสัญญาณจากขา 14 เป็นการหารความถี่จาก crystal 3.2768MHz ด้วย 128 จะได้สัญญาณ sampling 25.6 KHz ซึ่งเพียงพอในการ sampling เสียงแก้วแตก ครอบคลุมความถี่ได้ดีพอสมควร ซึ่งหน่วยความจำที่ใช้ เป็น 32K สองตัว คือ IC7,8 หน่วยความจำทั้งหมด 64 K จะเก็บสัญญาณได้ 2.56 วินาที ซึ่งก็นานพอสมควรที่จะเก็บสัญญาณเสียงในส่วนสำคัญ ๆ ไว้ได้

เมื่อเกิดการรีเซตจาก computer ซึ่งผ่านทาง port 300H-31FH คือในวงจรถ่าย 300H คือ A9-A0 เป็น 1100000000 เมื่อส่ง out port ที่สัญญาณ IOWR เป็น 0 สัญญาณจาก IC 9 จะได้สัญญาณ Y เป็น 0 IC 10A เป็น 0 ผ่านอินเวอร์เตอร์เป็น 1 ไปรีเซต IC4,5 ซึ่งเป็น counter ทั้งสองตัวซึ่งต่อกัน เมื่อส่งรีเซตมา เข้าพุทของ counter ทั้ง 2 ตัว เป็น 0 IC 2 ได้รับสัญญาณ 0 ที่ขา 1 SEL ทำให้เป็นการต่อสัญญาณชุด A กับเข้าพุท Y เพราะฉะนั้น สัญญาณ clock ของ IC4จะได้จาก A0 ซึ่งเป็นสัญญาณ sampling จาก IC1 ขา Y1 ต่อ A1 ซึ่งเป็นสัญญาณ OE(output enable) ของ IC7,8 ซึ่งให้อ่านพุทลอยตัวไว้ Y3 เป็นการต่อขา WE เพื่อเขียนข้อมูลจาก IC6ADC เข้าไปเก็บใน RAM ในแต่ละสัญญาณ clock แต่ละลูก IC 5 เป็น IC counter อีกตัวซึ่งต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับ IC 4 เมื่อวงจรนับนับจนถึง 64k เต็มหน่วยความจำแล้ว สัญญาณที่ IC5 ขา Q4 จะเป็น 1 ก็จะเป็นการส่งสัญญาณไปหยุด ADC ให้ CE เป็น 1 และส่งสัญญาณไปควบคุม IC2 ที่ขา 1 SEL เป็นการต่อสัญญาณเอาพุท Y กับสัญญาณอินพุท B และส่งสัญญาณไปที่ IC9 ที่ขา D7 เพื่อให้ computer สั่งมาอ่านคอยดูว่าเก็บข้อมูลเต็มหน่วยความจำหรือยัง ถ้าเต็มก็จะมีสัญญาณมาที่ขา 1 เมื่อเก็บข้อมูลเต็มหน่วยความจำ ที่ IC2 จะเป็นการต่อสัญญาณเอาพุท Y กับอินพุท B คือสัญญาณ clock ของวงจรจะได้รับการอ่านข้อมูลได้การสั่งอิน port ที่ 300H ในแต่ละครั้งก็เป็นการเลื่อนแอดเดรสของ RAM ไปหนึ่งแอดเดรส สัญญาณอินจากขา IORD ได้ที่ IC10C ผ่านอินเวอร์เตอร์ IC11B เข้าขา B0 เป็นการ count แอดเดรสของ RAM การอ่านข้อมูลจาก computer เป็นการตั้ง loop ว่าจะอ่านข้อมูลจาก หน่วยความจำในการ์ดนี้ไปมากน้อยเท่าใด แล้วจึงนำข้อมูลไปประมวลผลต่อไป

3.2 การทำงานของโปรแกรมวิเคราะห์และเปรียบเทียบเสียงกระจกแตก

เมื่อเข้าสู่โปรแกรมจะขึ้นเมนูหลักดังนี้

- 0 -> Exit
- 1 -> Sound record from ADC
- 2 -> Plotgraph
- 3 -> Files compare

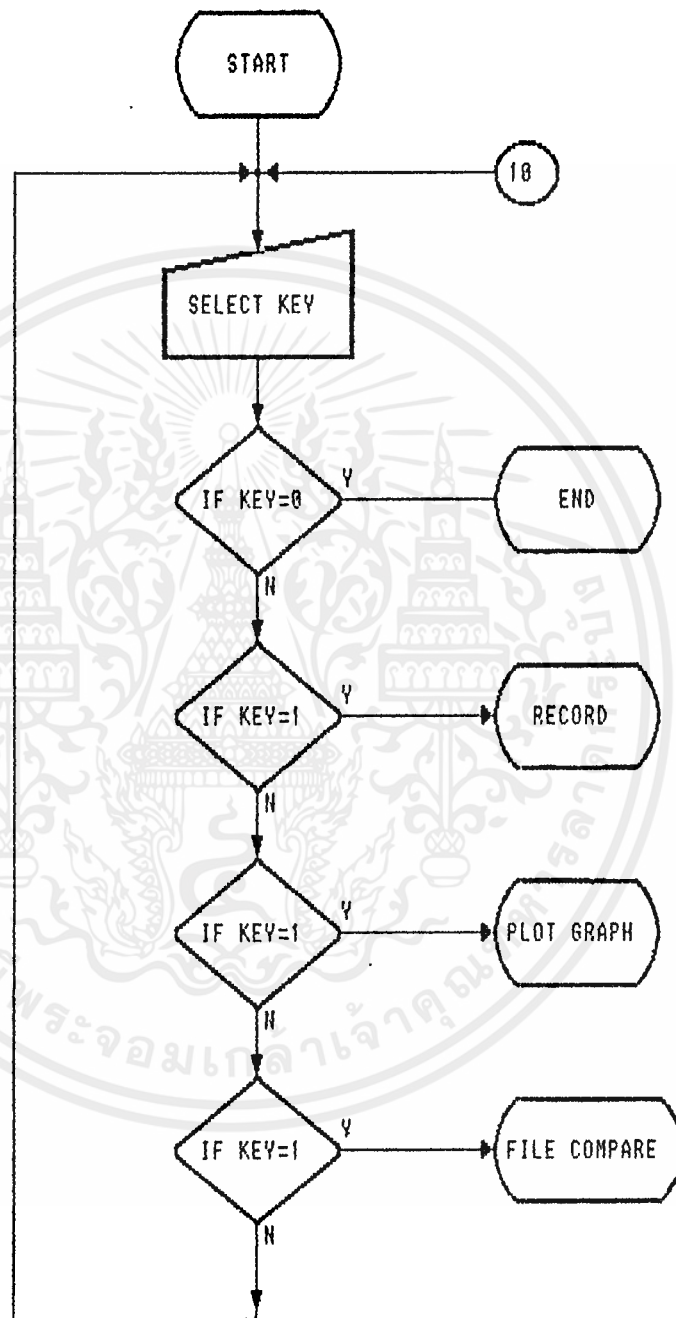
Exit ทำหน้าที่ออกจากโปรแกรม กลับไปที่ดอส เมื่อกดคีย์ "0"

Sound record from ADC ทำหน้าที่บันทึกเสียงจากการ์ด เก็บไว้เป็นไฟล์ ในดิสค์ เมื่อกดคีย์ "1" โปรแกรมจะหน่วง 3 นาที นับเวลาถอยหลังจนถึง 0 จะเริ่มบันทึกเสียงจากไมโครโฟน เก็บไว้ที่ แรม ของการ์ด ต่อจากนั้นจึงจะเริ่มดึงข้อมูลจาก แรมเก็บลง ดิสค์ ความจุของไฟล์ที่เก็บ เท่ากับ 4096 ไบท์

Plot graph ทำหน้าที่แสดงรูปคลื่นสัญญาณเสียงที่บันทึกไว้ตามแกนเวลา, แกนความถี่ และเก็บค่าความถี่ 200 ค่าแรกที่มี แมกนิจูดสูงที่สุดเก็บไว้เป็นไฟล์ในดิสค์ การทำงาน กดคีย์ "2" โปรแกรมจะถามชื่อไฟล์เสียงที่ต้องการแสดง เมื่อพิมพ์ชื่อไฟล์ที่ต้องการแสดง ผลเรียบร้อยแล้ว กดคีย์ เอนเตอร์ โปรแกรมจะเริ่มพล็อตรูปคลื่นตามแกนเวลาในหน้าต่าง ที่ 1 เราสามารถเลื่อนตำแหน่งข้อมูลเพื่อหา จุดเริ่มต้น และจุดปลายของสัญญาณเสียงได้ โดยใช้คีย์ "Tab", "\b" ทำการเลื่อนข้อมูลดอยหลังและไปข้างหน้าทีละ 50 ตำแหน่งตาม ลำดับ และคีย์ "<-", "->" เลื่อนข้อมูลไปข้างหน้าและดอยหลังทีละ 1 ตำแหน่งตามลำดับ หลังจากดูรูปคลื่นตามแกนเวลาเรียบร้อยแล้วต้องการออกไปที่ เมนเมนูให้กดคีย์ "Esc" แต่ถ้าต้องการดู สเปคตรัมตามแกนความถี่ต้องทำการกำหนดจุดเริ่มและจุดสิ้นสุดของสัญญาณ การกำหนดจุดเริ่มต้นของสัญญาณทำได้โดยเลื่อนจุดเริ่มต้นของสัญญาณให้ชิดขอบซ้ายของ หน้าต่างแล้วกดคีย์ "s" ขึ้นต่อไปคือหาจุดปลายของสัญญาณให้ตรงกับขอบขวาของหน้าต่าง เสร็จแล้วกดคีย์ "E" จากนั้นกด "d" เครื่องจะเริ่มคำนวณ ฟาสต์ฟูเรียร์ และพล็อต แมกนิจูดตามแกนความถี่ ในหน้าต่างที่ 2 และสามารถเลื่อนดูข้อมูลที่ตำแหน่งต่าง ๆ ได้ เมื่อเลิกดูให้กด "Esc" เครื่องจะเริ่มเรียงข้อมูลที่มีแมกนิจูดสูงสุด 400 ค่าแรกมาเก็บไว้ ที่อาร์เรย์ mag[0]-mag[400] และใช้อาร์เรย์นี้ไปที่ค่าความถี่ เสร็จแล้วเก็บไว้ใน อาร์เรย์ freq[0]-freq[400] และแสดงค่าแมกนิจูดสูงสุด 200 ค่าแรก พร้อมทั้งค่าความถี่ที่ชี้ได้ เมื่อแสดงค่าเสร็จจะขึ้นข้อความให้ ตั้งชื่อไฟล์ เพื่อเก็บเป็นข้อมูลความถี่เสียงไว้เปรียบ- เทียบกับไฟล์ข้อมูลความถี่เสียงอื่นต่อไป

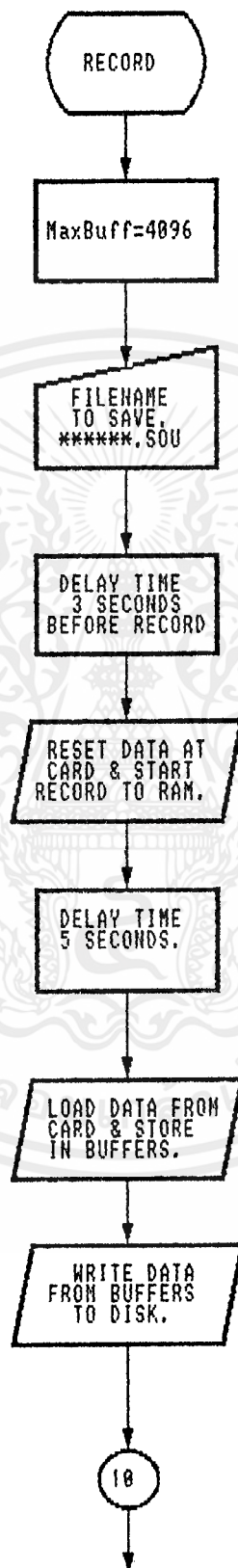
Files compare ทำหน้าที่เปรียบเทียบข้อมูล ของไฟล์ความถี่เสียง

การทำงาน จะขึ้นข้อความให้ใส่ชื่อไฟล์ที่ต้องการเปรียบเทียบไฟล์แรกและไฟล์ที่สอง เสร็จแล้วจะขึ้นข้อความว่าต้องการเปรียบเทียบข้อมูลที่จุด (ในที่นี้กำหนดให้เปรียบเทียบได้ สูงสุด 200 ชุด) ถ้าข้อมูลความถี่ชุดใดชุดหนึ่งตรงกันก็จะให้บวกสะสมเพิ่มขึ้น เมื่อเปรียบเทียบ เสร็จจะนำค่าสะสมที่ได้มาแสดงผลความถูกต้อง เป็น เปอร์ เซนต์



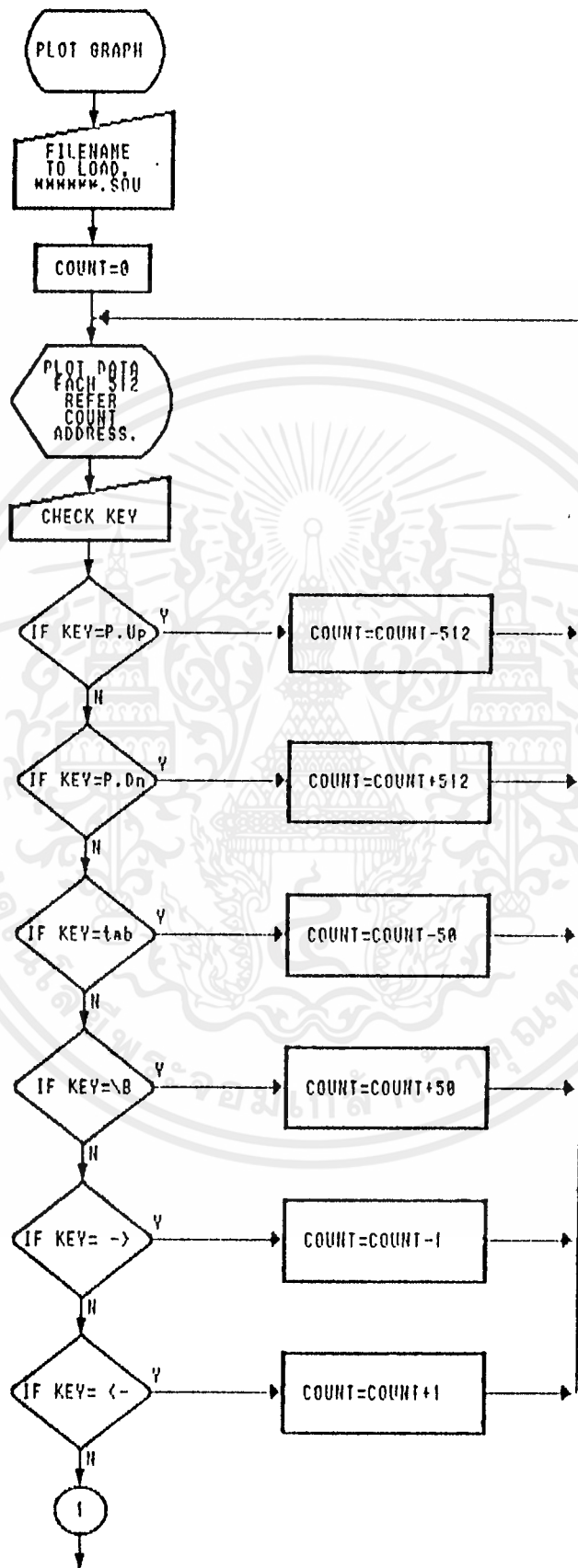
รูปที่ 3.3 แสดง Flow chart ของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

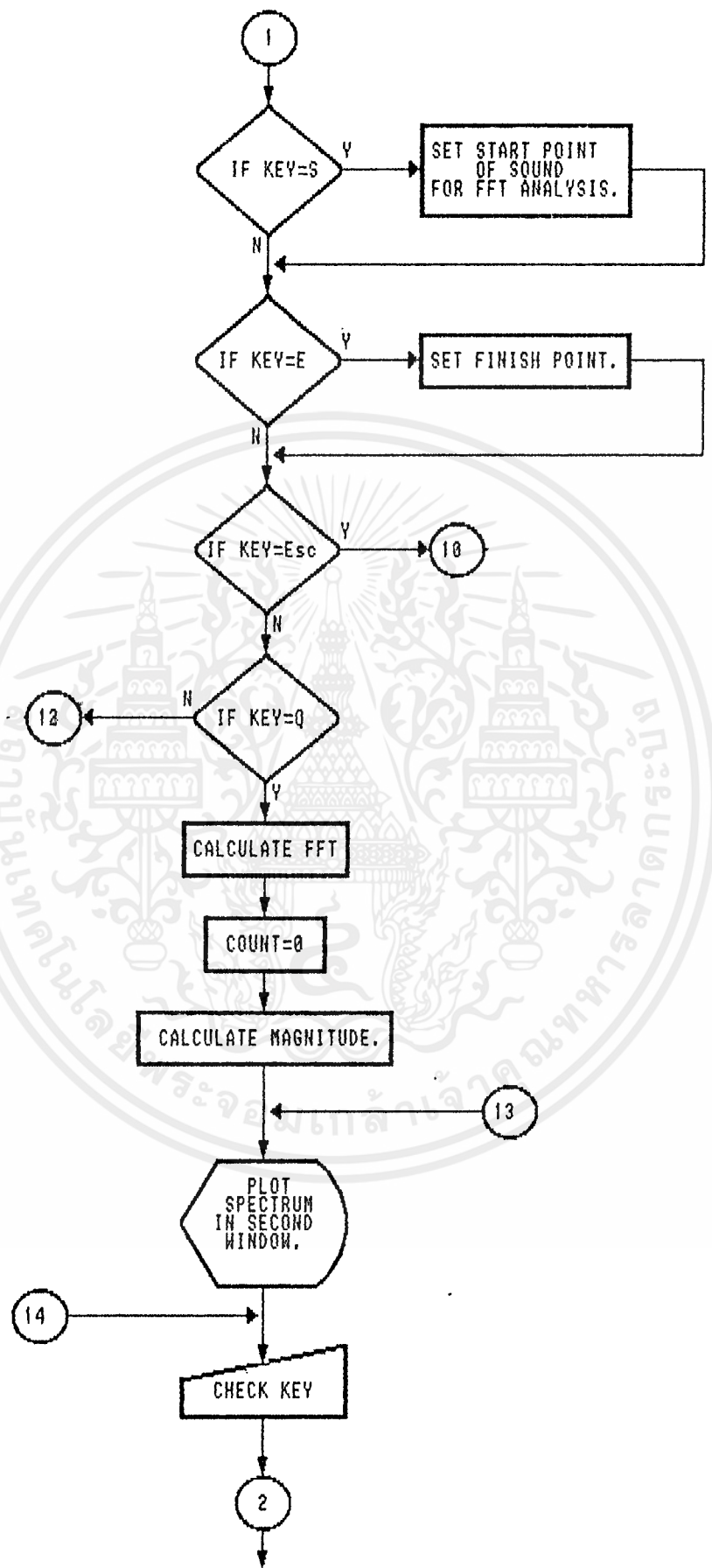


รูปที่ 3.4 แสดง Flow chart ของโปรแกรม ทำหน้าที่บันทึกเสียง

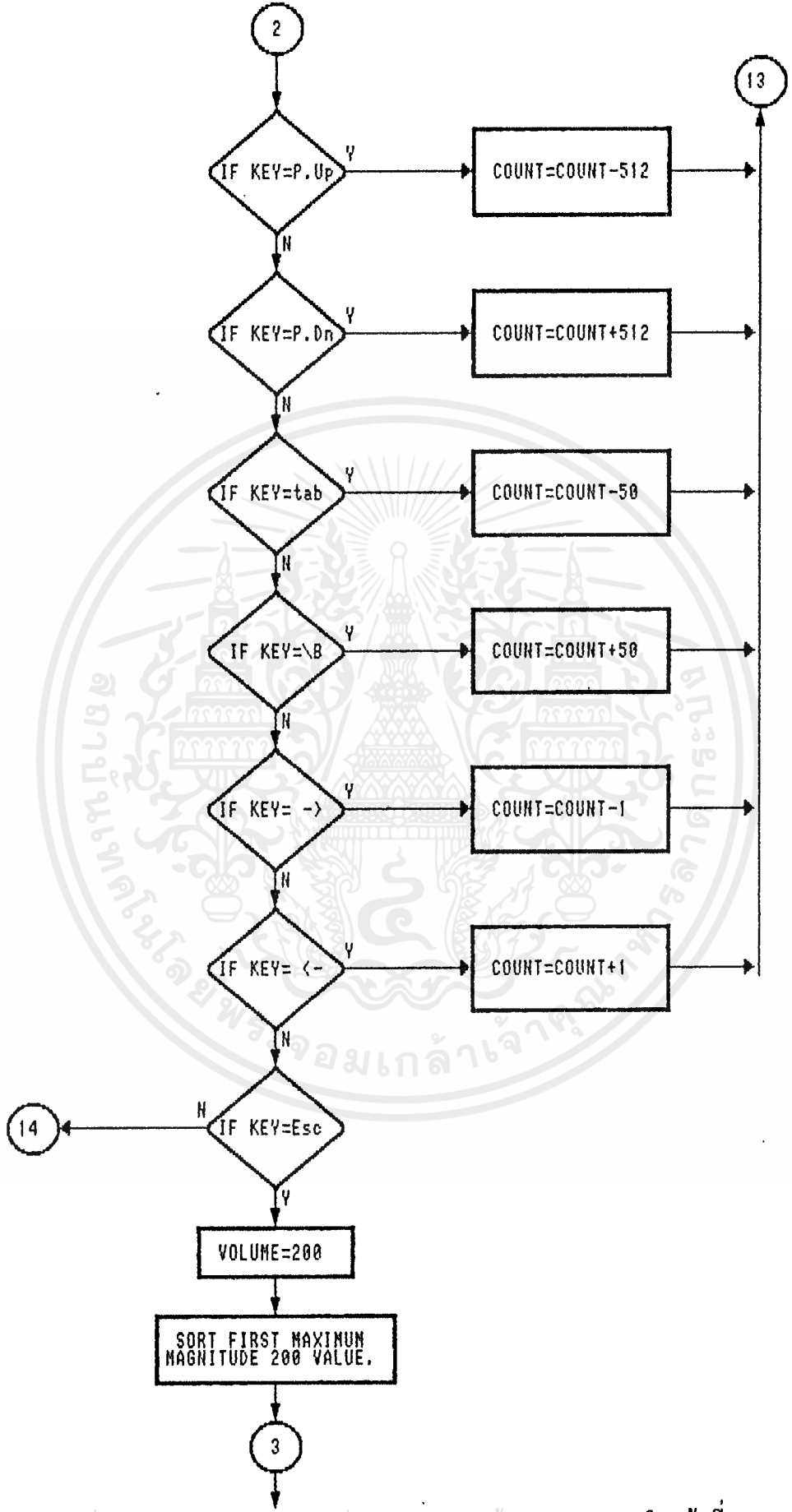
เอกสารนี้เป็นเอกสารที่สงวนเวลาสำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนักผู้จัดทำเห็นว่าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



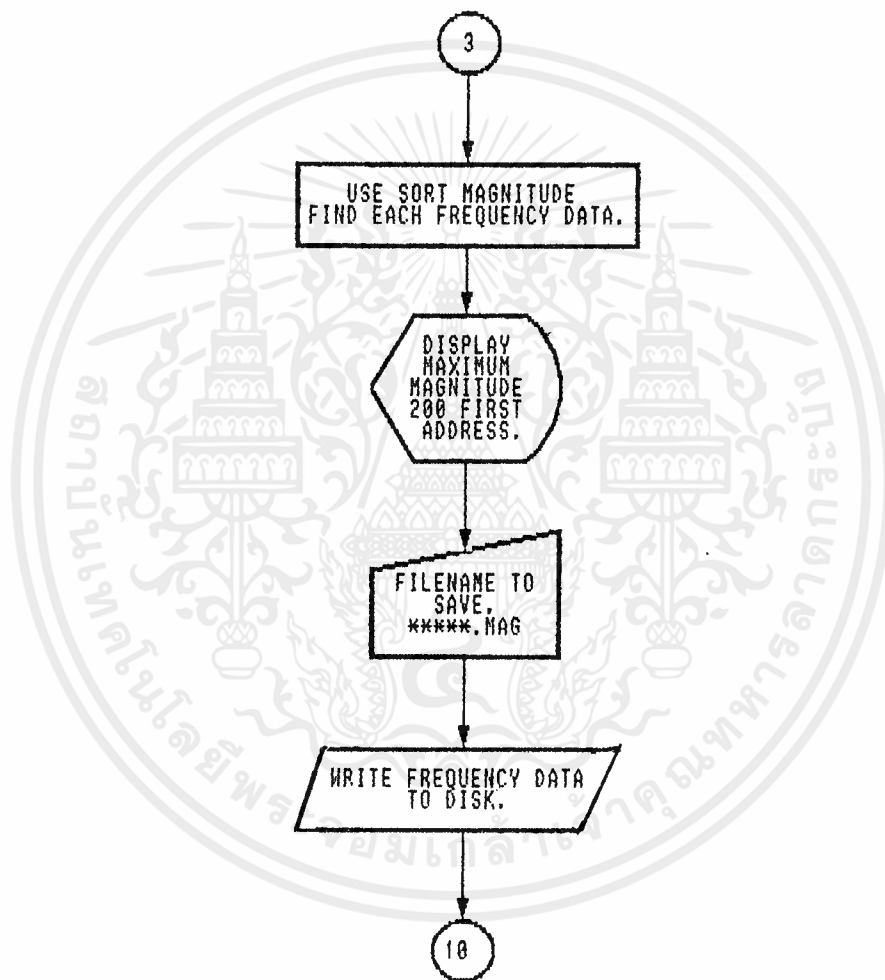
เอกสารที่ 3.5(a) ที่แสดง Flowchart ของโปรแกรม plot graph ทำหน้าที่แสดงขั้นตอนการคำนวณว่ากรณีใดๆ ทั้งสิ้น ผลลัพธ์ในโตเมทเวลล์ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5(b) แสดง Flow chart ของโปรแกรม plot graph ทำหน้าที่
 เอกสารประกอบเอกสารหนังสือพิมพ์หรือการอื่นที่ออกให้ประชาชน เมื่อผู้จัดทำไปใช้จะระเขยนด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกวีเคราะห์ความถี่และแสดงผลอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

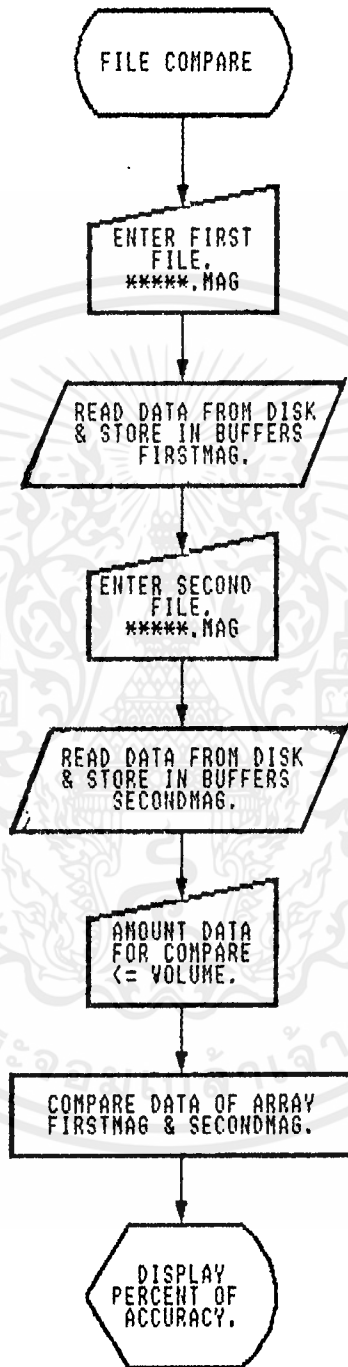


เอกซรูปที่ 3.5(c) แสดง Flow chart ของโปรแกรม plot graph ทำหน้าที่แสดง
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น ผลลัพธ์ใดไม่มีความถี่และเรียงข้อมูล
 ไปถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5(d) แสดง Flow chart ของโปรแกรม plot graph ทำหน้าแสดง
ผลข้อมูลความถี่และเก็บบันทึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 แสดง Flow chart ของโปรแกรม File compare ทำหน้าที่เปรียบเทียบไฟล์ข้อมูลความถี่และแสดงผลความถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

การทดลองช่วงแรกเป็นการดึงข้อมูลจาก RAM บนการ์ดซึ่งต่อร่วมกับเครื่อง PC มา และใช้ข้อมูลที่ได้นำไปคำนวณซึ่งก็ใช้โปรแกรมภาษา C คำนวณที่เครื่อง PC เนื่องจากการทำงานของเครื่อง PC เร็วกว่าความเร็วของการ์ด ที่ต่อเข้าไปมากทำให้ข้อมูลที่ดึงมาได้เป็นศูนย์ตลอด ต่อมาจึงได้ทดลองหาความเร็วของโปรแกรมในการดึงข้อมูล ทำให้ได้ข้อมูลที่ถูกต้อง

ในการทดลองที่ใช้ความถี่ในการ sampling 25.6 KHz ทำให้ครอบคลุมความถี่ที่ความถี่ที่ต้องการเก็บ และดูช่วงการเริ่มเกิดของเสียง และจุดสิ้นสุดของเสียง เพื่อความในการเก็บรูปแบบให้คล้ายๆ กันทั้งในการทดลองให้ผลของการแยกแยะเสียงได้ดีพอสมควร

ส่วนผลการเปรียบเทียบเสียงในการทดลองเช่น เคาะกระเบื้อง กับทุบกระจก ผลการเปรียบเทียบในโดเมนความถี่จะเห็นว่าแถบความถี่ที่มีแมกนิจูดสูงๆ มีช่วงที่ตรงกันเพียงเล็กน้อย แต่ถ้าเป็นเสียงเช่นทุบกระจกเหมือนกัน ความถี่แต่ละช่วงจะมีสเปคตรัมแสดงลักษณะแมกนิจูดช่วงนั้นคล้ายๆ กันมากดังนั้นการเปรียบเทียบเราจึงใช้ความสูงของแมกนิจูดไปหาค่าความถี่แต่เนื่องจากในโดเมนความถี่ lower, upper sideband ซึ่งมีค่าแมกนิจูดเท่ากันทั้งสองด้าน จึงตัดนำมาคำนวณเพื่อใช้หาค่าความถี่ในการเปรียบเทียบเพียงด้านเดียว

ส่วนเสียงคนพูด ลักษณะสเปคตรัมจะแตกต่างไปจากเสียงแก้วแตก สเปคตรัมของเสียงคนพูดจะมีค่าแมกนิจูดสูงๆ อยู่ในแถบความถี่ต่ำๆ ดังนั้น จึงสามารถแยกเสียง คนพูดกับเสียงแก้วแตกได้อย่างชัดเจน

ผลการทดลองเปรียบเทียบเสียงทั้งที่เป็นเสียงเดียวกัน และเสียงต่างชนิดกันโดยใช้ไฟล์ข้อมูลความถี่เสียงซึ่งได้จากการคำนวณ FFT แสดงในตารางที่ 4.1

ผลการเปรียบเทียบความเหมือนกันของ PATTERN เสียง

เสียงที่ 1	เสียงที่ 2	จำนวนข้อมูลของเสียงที่นำมาเปรียบเทียบ		
		80	150	200
GLASS1	GLASS2	42%	48%	49%
GLASS1	GLASS3	62%	56%	54%
GLASS2	GLASS3	60%	53%	52%
CUP1	CUP2	32%	52%	48%
CUP1	CUP3	40%	45%	41%
CUP2	CUP3	62%	53%	51%
COIL21	COIL22	40%	48%	47%
COIL21	COIL23	27%	48%	52%
COIL22	COIL23	20%	34%	38%
GLASS1	CUP1	0%	4%	5%
GLASS1	CUP2	0%	1%	3%
GLASS1	CUP3	0%	2%	6%
GLASS2	CUP1	0%	0%	3%
GLASS2	CUP2	0%	0%	3%
GLASS2	CUP3	0%	2%	8%
GLASS3	CUP1	0%	4%	6%
GLASS3	CUP2	0%	5%	6%
GLASS3	CUP3	0%	8%	12%

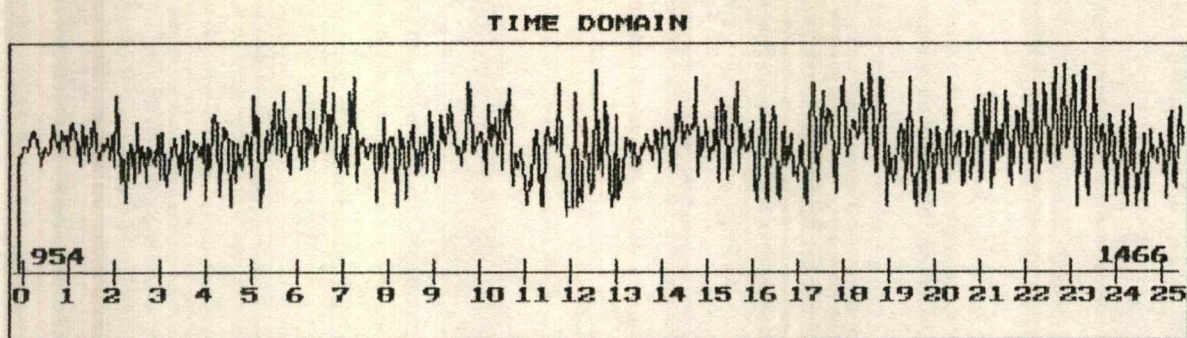
ตารางที่ 4.1 แสดงผลการเปรียบเทียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

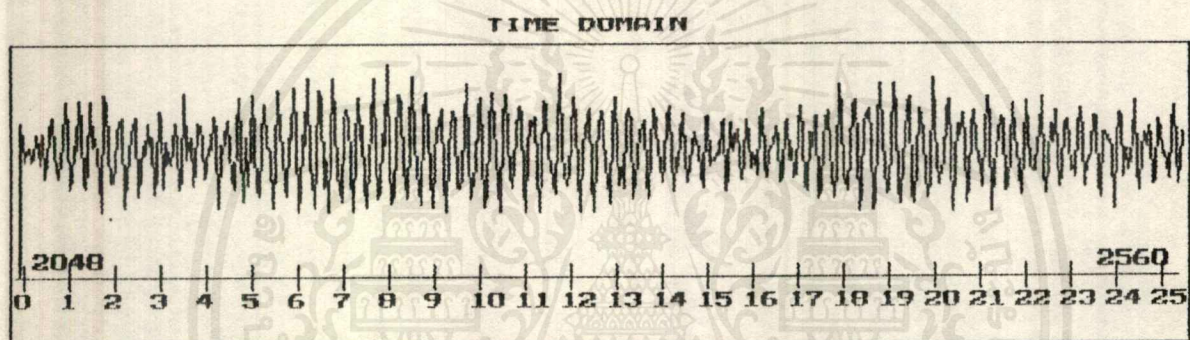
เสียงที่ 1	เสียงที่ 2	จำนวนข้อมูลของเสียงที่นำมาเปรียบเทียบ		
		80	150	200
CUP1	COIL21	10%	10%	11%
CUP1	COIL22	0%	2%	3%
CUP1	COIL23	0%	2%	3%
CUP2	COIL21	2%	9%	9%
CUP2	COIL22	0%	1%	3%
CUP2	COIL23	0%	1%	2%
CUP3	COIL21	0%	4%	7%
CUP3	COIL22	0%	1%	6%
CUP3	COIL23	0%	0%	6%

ตารางที่ 4.1 แสดงผลการเปรียบเทียบ (ต่อ)

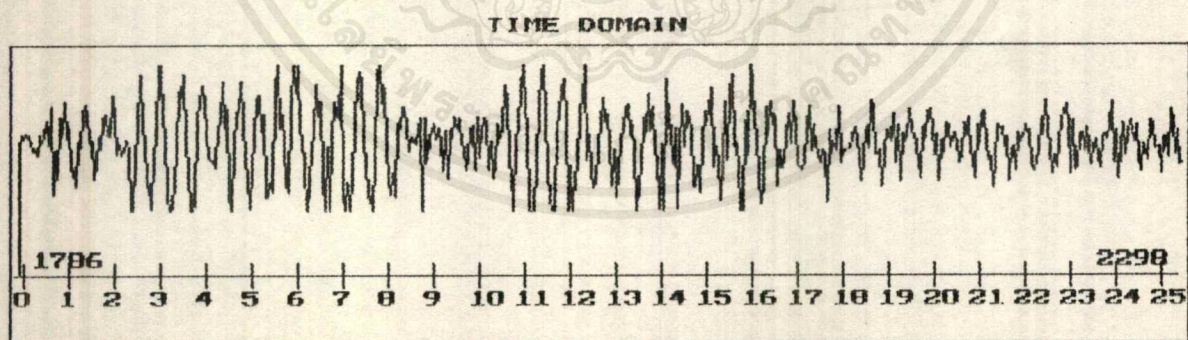
GLASS1-3 = GLASS1-3.MAG เป็นไฟล์เก็บข้อมูลของเสียงการทบทวนกระจก
 CUP1-3 = CUP1-3.MAG เป็นไฟล์เก็บเสียงการเคาะแก้ว
 COIL21-23=COIL21-23.MAG เป็นไฟล์เก็บเสียงการหล่นของเหรียญกระทบพื้น



รูปที่ 4.1 (a) แสดงรูปคลื่นสัญญาณเสียงแก้วแตก

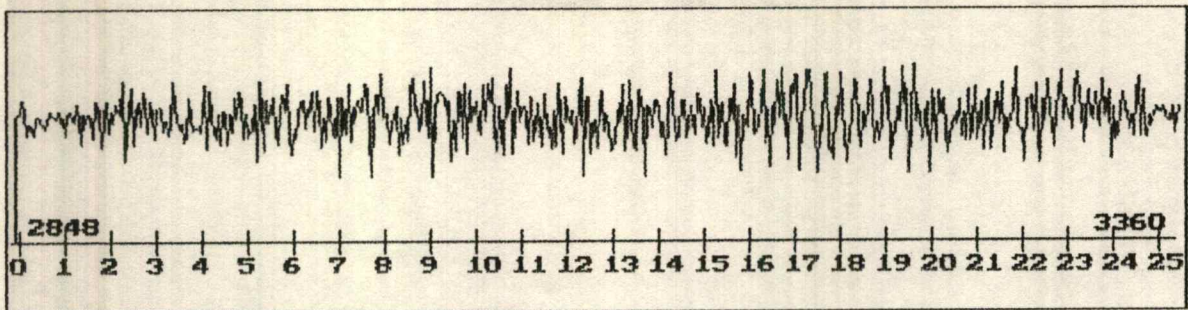


รูปที่ 4.1 (b) แสดงรูปคลื่นสัญญาณเสียงเคาะแก้ว



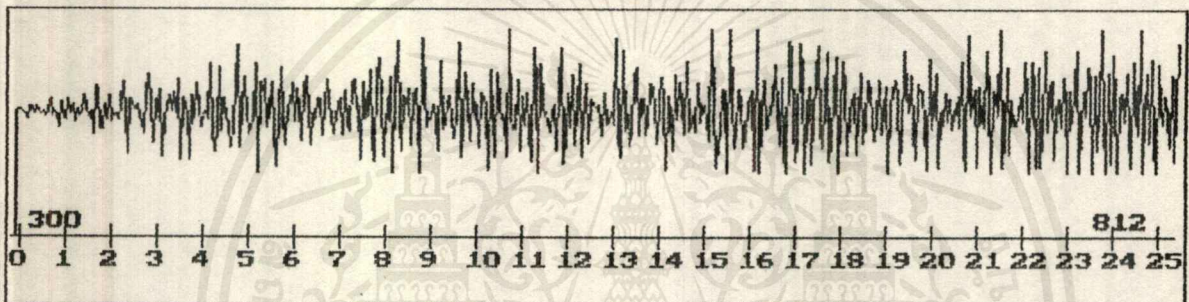
รูปที่ 4.1 (c) แสดงรูปคลื่นสัญญาณเสียงเหรียญตกกระทบพื้น

TIME DOMAIN



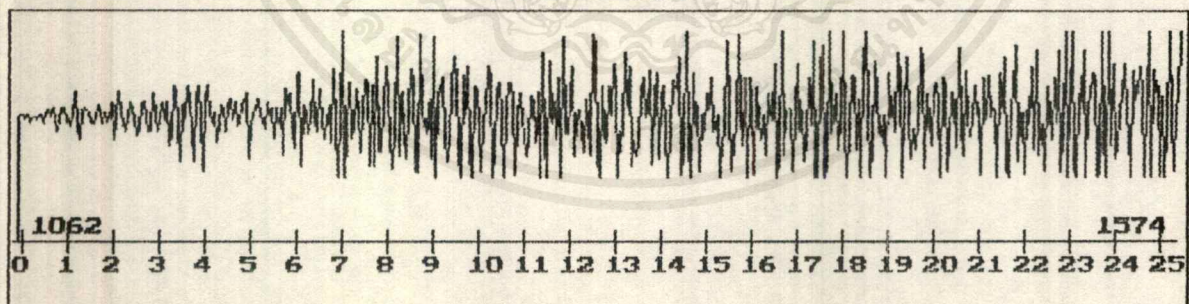
รูปที่ 4.1 (d) แสดงรูปคลื่นสัญญาณเสียงเคหะระจกตกกระทบผนัง

TIME DOMAIN



รูปที่ 4.1 (e) แสดงรูปคลื่นสัญญาณเสียงกระเบื้องแตก

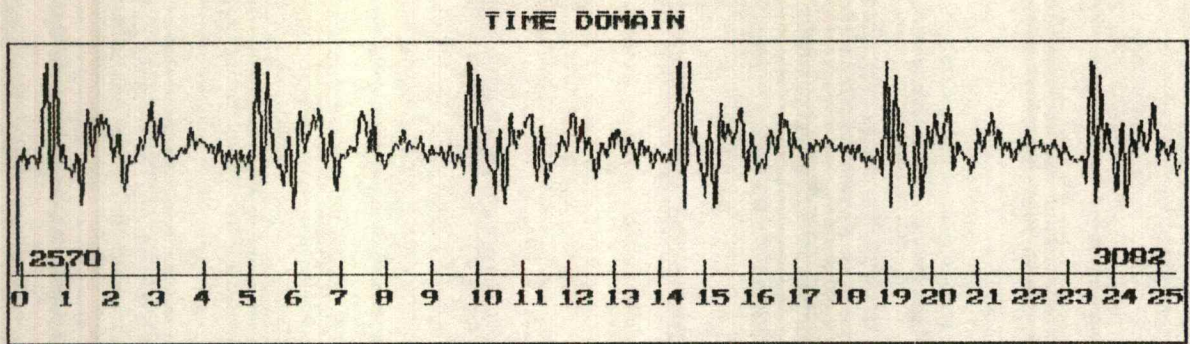
TIME DOMAIN



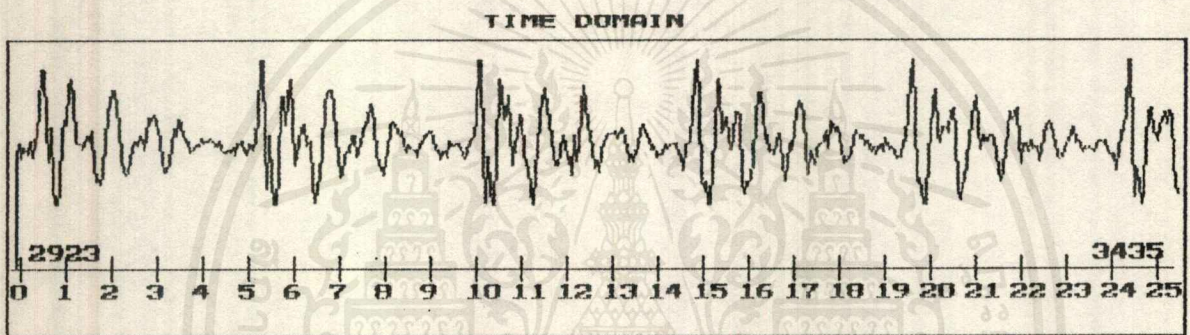
รูปที่ 4.1 (f) แสดงรูปคลื่นสัญญาณเสียงกระจกแตก

รูปที่ 4.1 แสดงการเปรียบเทียบรูปคลื่นเสียงที่เกิดจากการกระทบของวัตถุในโดเมนเวลา เห็นได้ว่ารูป (e) กับ (f) คล้ายกันมากแต่เมื่อเปรียบเทียบสเปกตรัมดังรูปที่ 4.10 กับ 4.11 จะเห็นความแตกต่างได้ชัดเจน

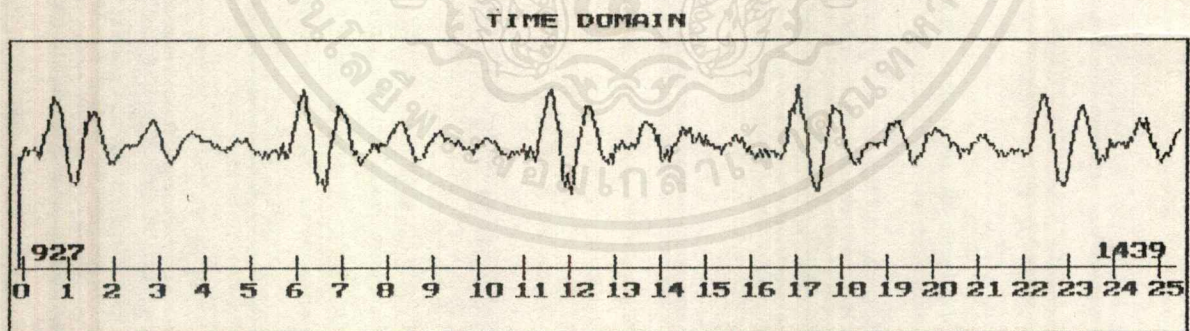
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 (a) แสดงรูปคลื่นสัญญาณเสียงพูดคำว่า "Test"



รูปที่ 4.2 (b) แสดงรูปคลื่นสัญญาณเสียงพูดคำว่า "One"

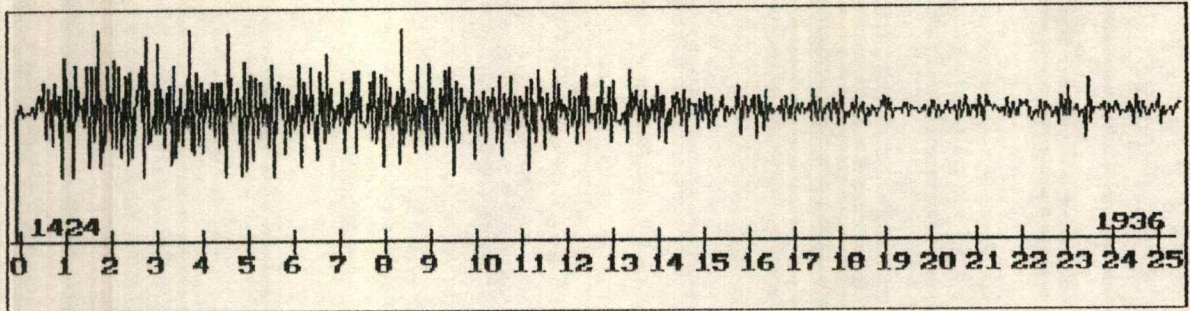


รูปที่ 4.2 (c) แสดงรูปคลื่นสัญญาณเสียงพูดคำว่า "Mode"

รูปที่ 4.2 แสดงการเปรียบเทียบรูปคลื่นสัญญาณเสียงพูดของมนุษย์ในโดเมนเวลาซึ่งจะแตกต่างจากลักษณะรูปคลื่นของเสียงที่เกิดจากการกระทบของวัตถุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

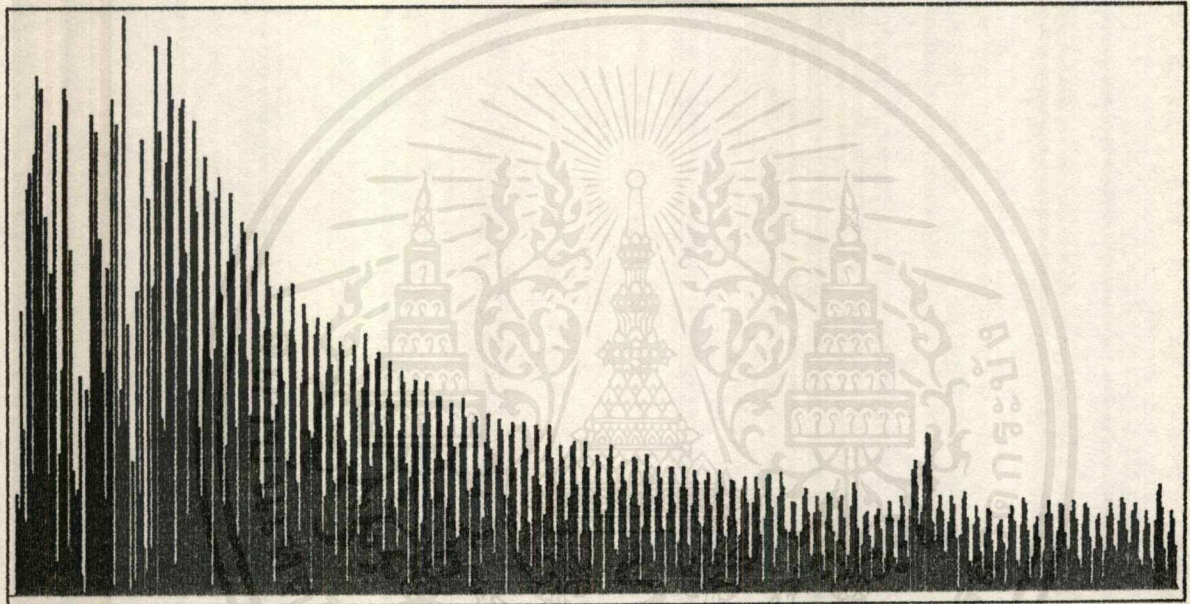
TIME DOMAIN



FFT solution.

Magnitude solution.

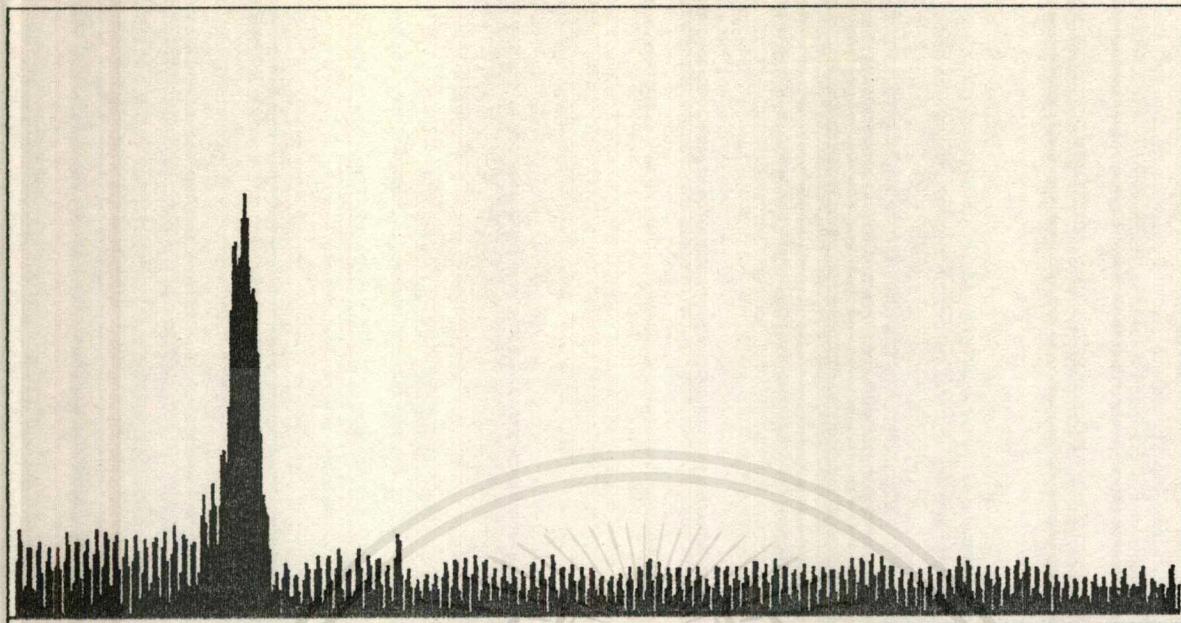
MAGNITUDE SPECTRUM



รูปที่ 4.3 แสดงรูปคลื่นสัญญาณเสียงเคาะแก้วในโดเมนเวลา และโดเมนความถี่เฟรมแรก ในส่วนของโดเมนความถี่ในช่วงเฟรมแรกนี้จะมีลักษณะคล้ายกัน เราจึงตัดทิ้งไม่นำข้อมูลในช่วงนี้ไปใช้เพื่อการเปรียบเทียบความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

MAGNITUDE SPECTRUM

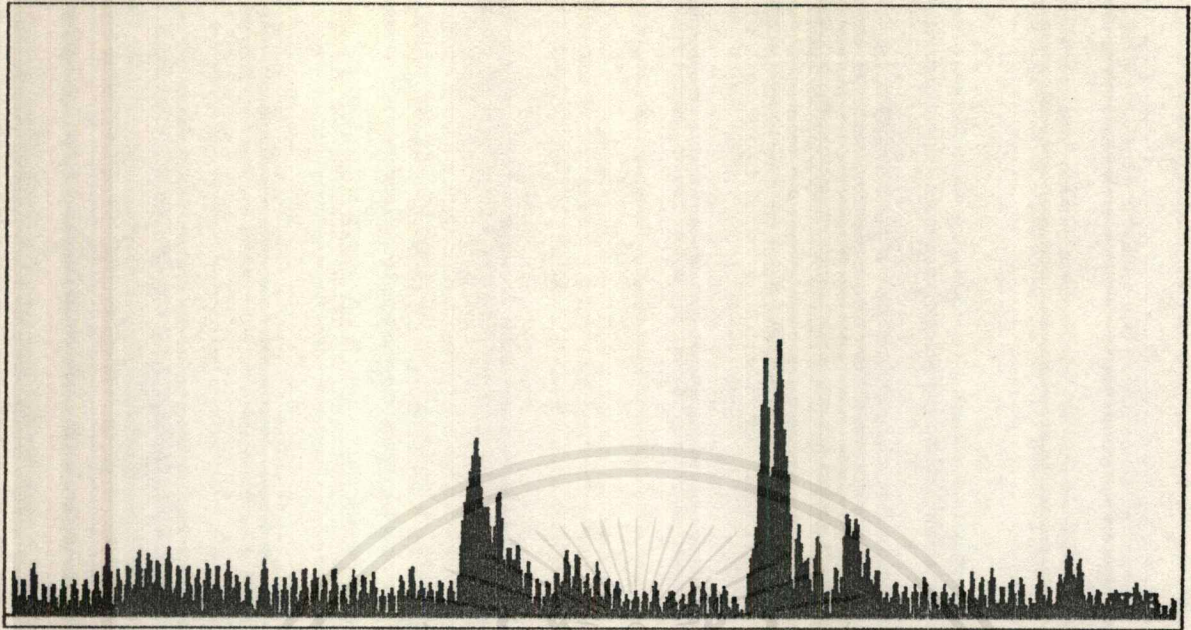


(b)

รูปที่ 4.4 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วใบเดียวกันครั้งแรกกับครั้งที่สอง ในเฟรมที่สองของโดเมนความถี่

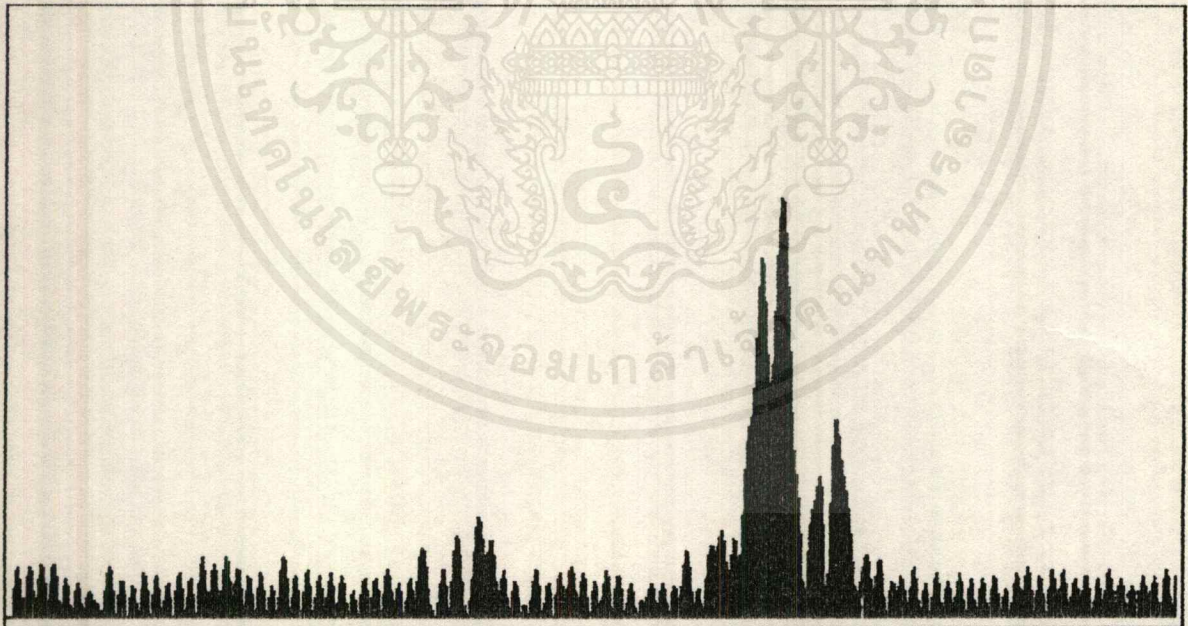
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

MAGNITUDE SPECTRUM

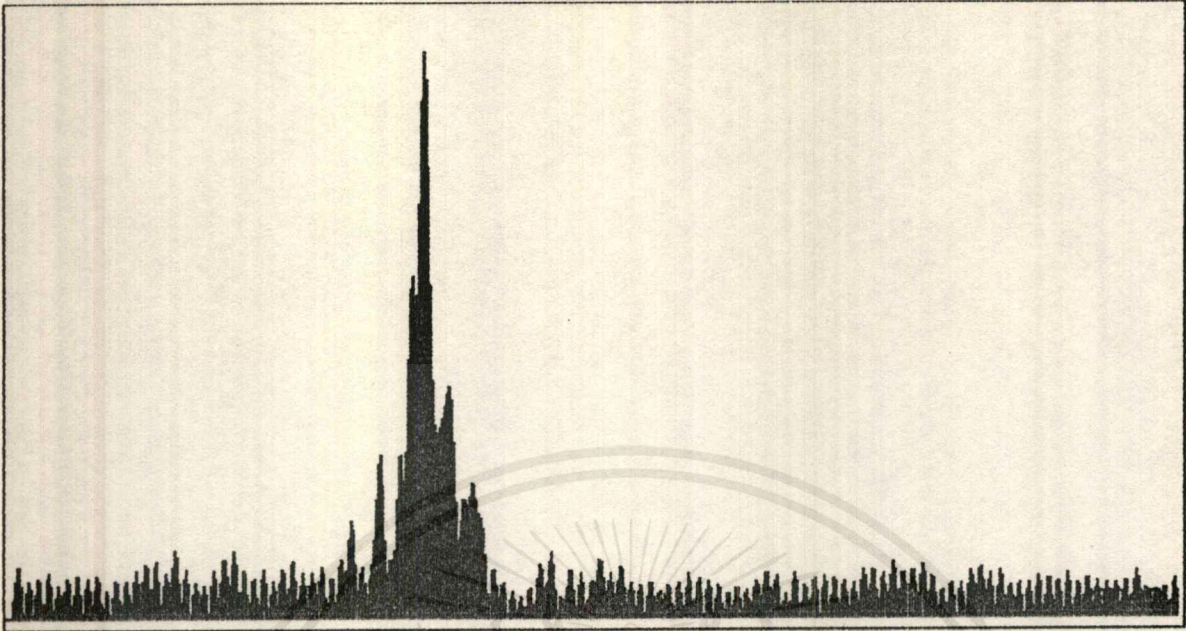


(b)

รูปที่ 4.5 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วใบเดียวกันครั้งแรกกับครั้งที่สอง
ในเฟรมที่สามของโดเมนความถี่

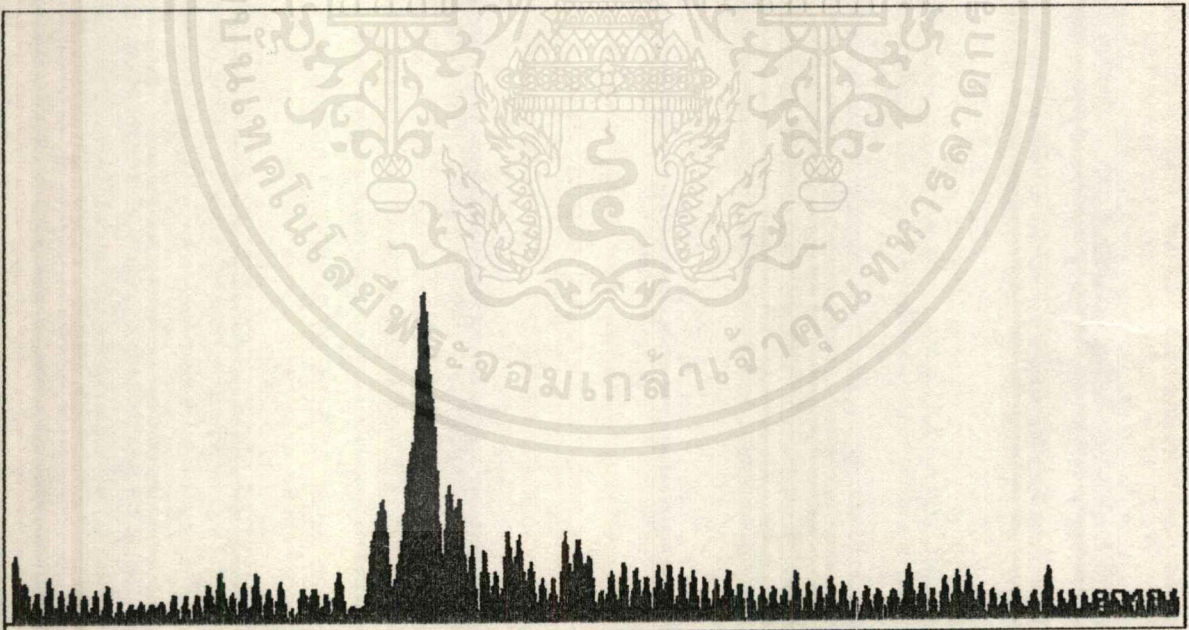
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

MAGNITUDE SPECTRUM

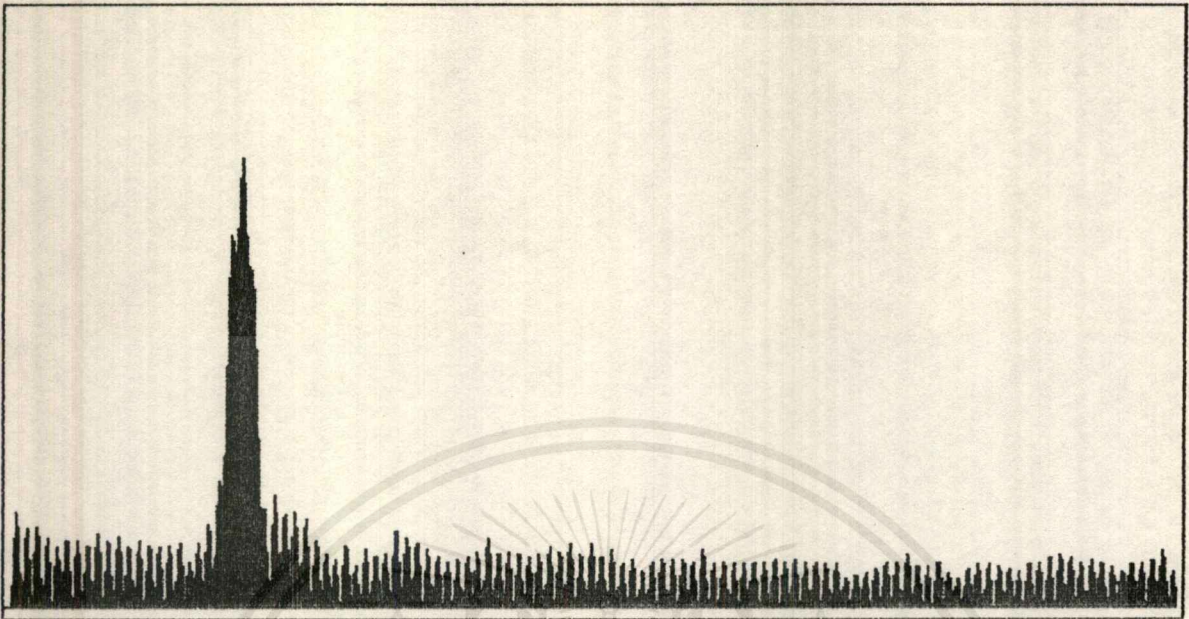


(b)

รูปที่ 4.6 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วใบเดียวกันครั้งแรกกับครั้งที่สอง ในเฟรมที่สี่ของโดเมนความถี่

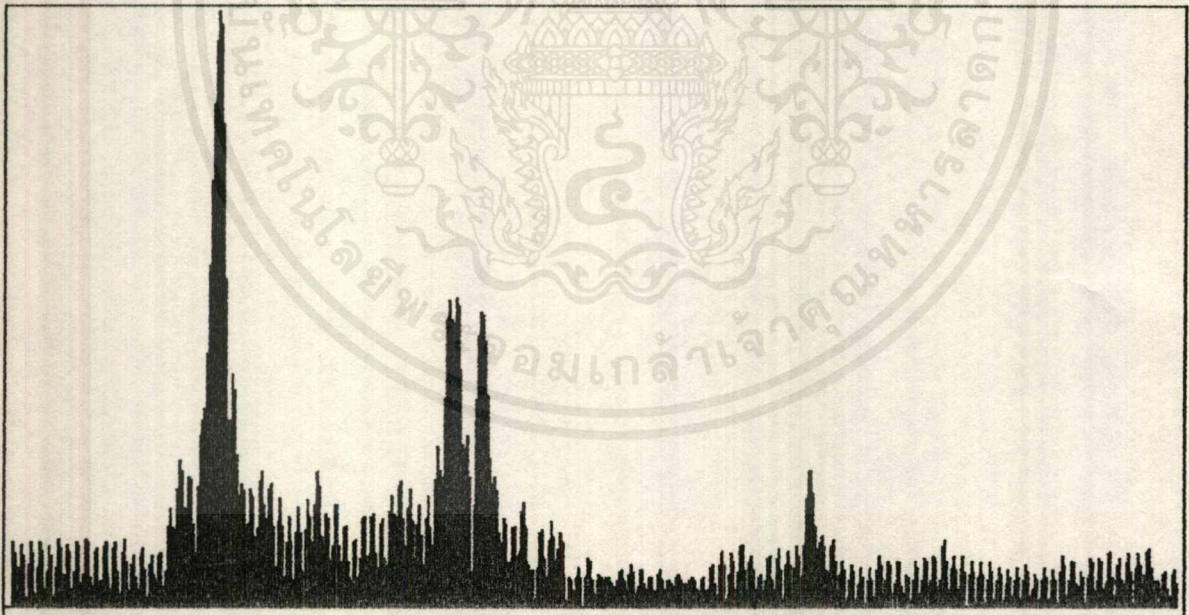
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

MAGNITUDE SPECTRUM

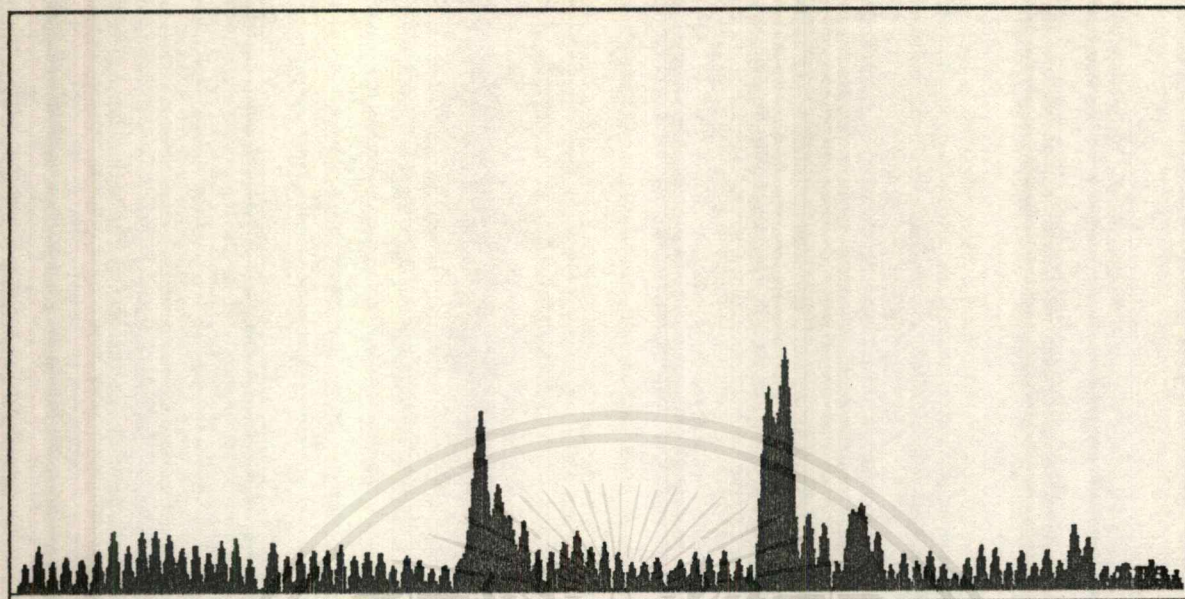


(b)

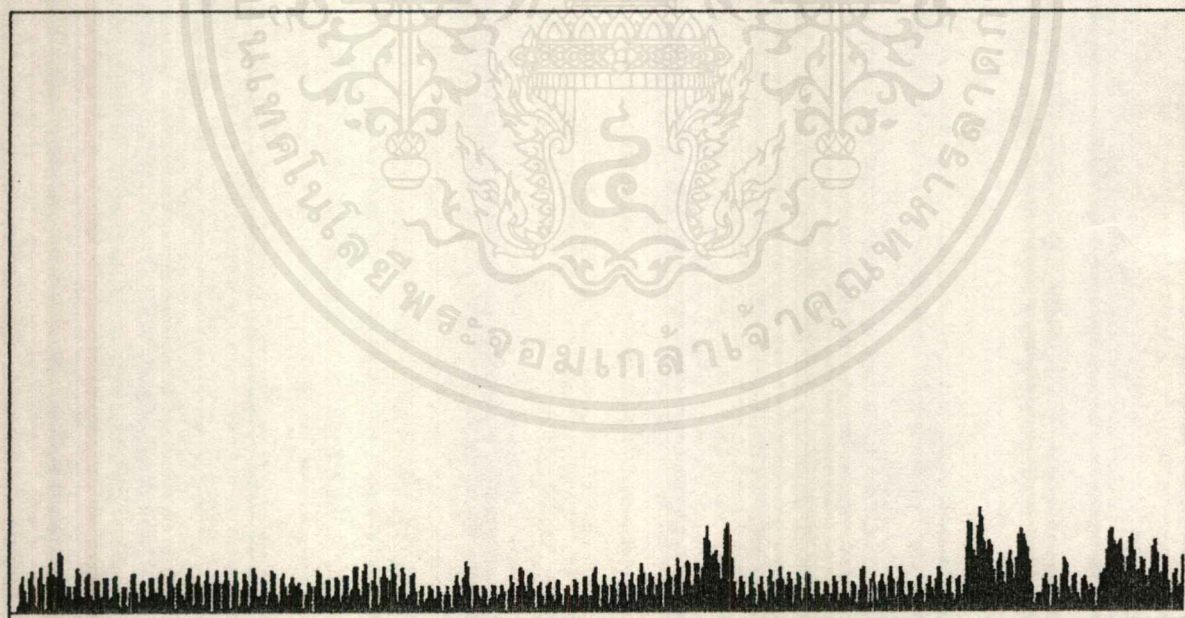
รูปที่ 4.7 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วกับเสียงเคาะกระเบื้องแผ่น
โนเฟรมที่สองของโตเมความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

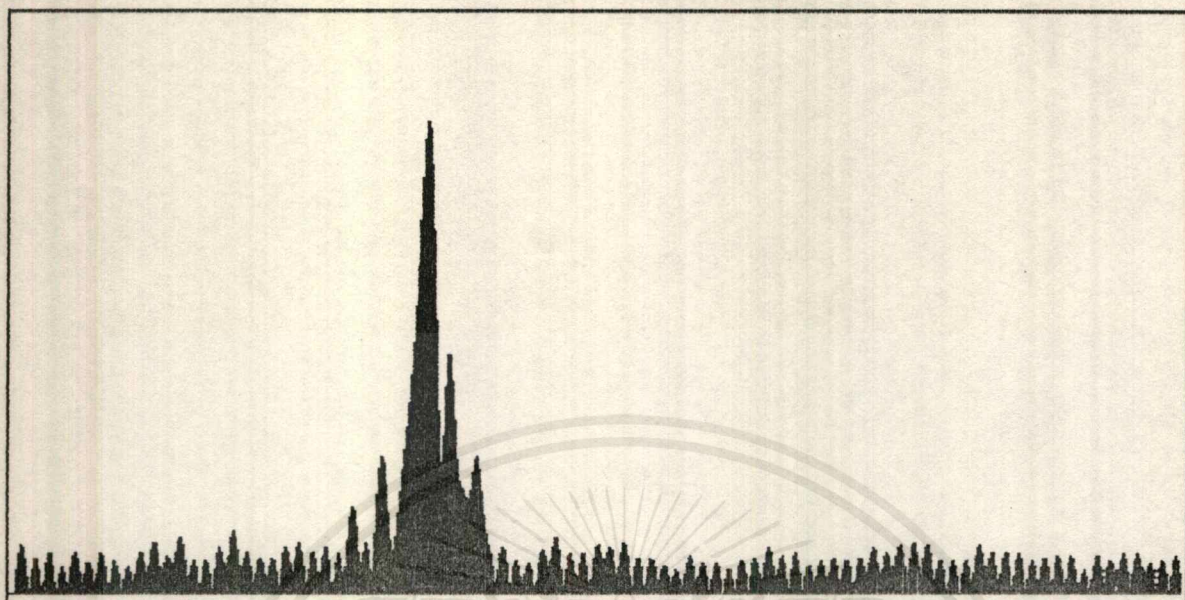


(b)

รูปที่ 4.8 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วกับเสียงเคาะกระเบื้องแผ่น
ในเฟรมที่สามของโดเมนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



(a)

MAGNITUDE SPECTRUM

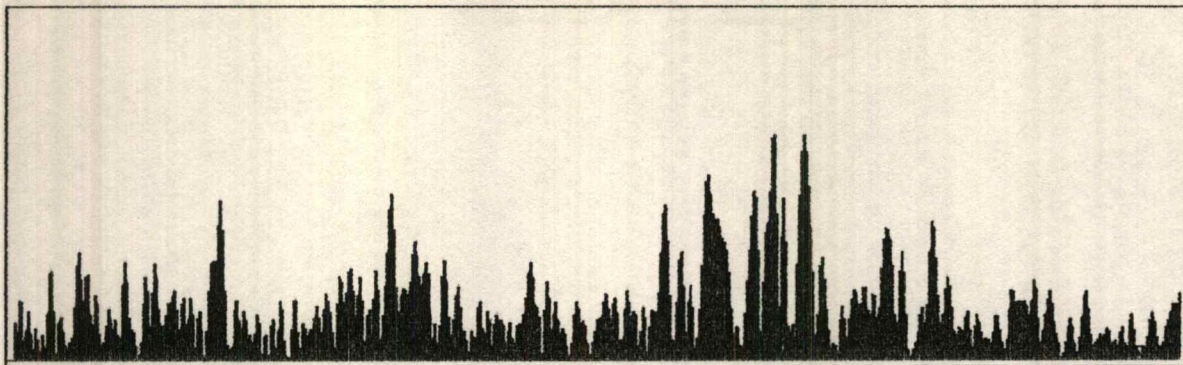


(b)

รูปที่ 4.9 แสดงการเปรียบเทียบความถี่เสียงเคาะแก้วกับเสียงเคาะกระเบื้องแผ่น
ในเฟรมที่สี่ของโดเมนความถี่

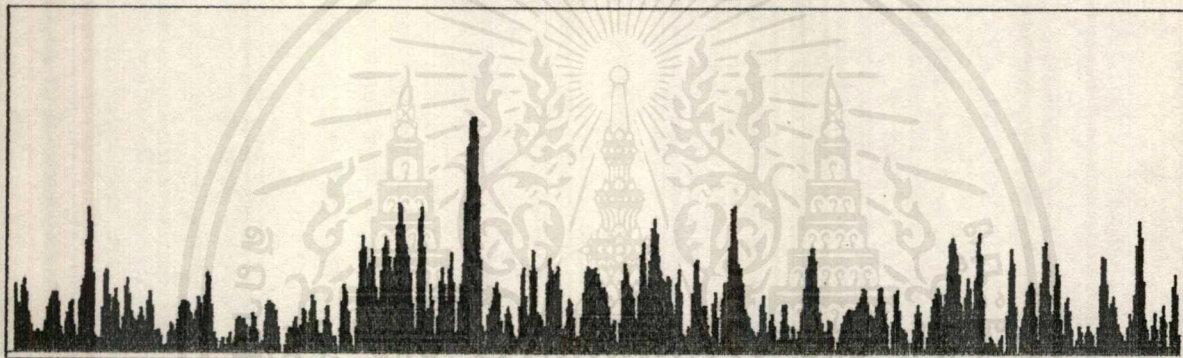
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



รูปที่ 4.10 (a) แสดงสเปกตรัมความถี่เสียงกระจกแตกเฟรมที่สอง

MAGNITUDE SPECTRUM



รูปที่ 4.10 (b) แสดงสเปกตรัมความถี่เสียงกระจกแตกเฟรมที่สาม

MAGNITUDE SPECTRUM

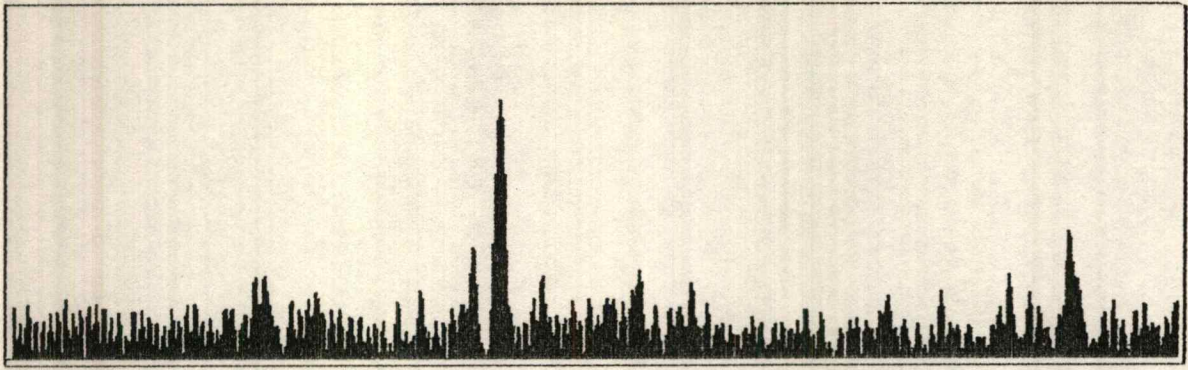


รูปที่ 4.10 (c) แสดงสเปกตรัมความถี่เสียงกระจกแตกเฟรมที่สี่

รูปที่ 4.10 แสดงสเปกตรัมความถี่เสียงกระจกแตกของรูปคลื่นจากรูปที่ 4.1 (f)

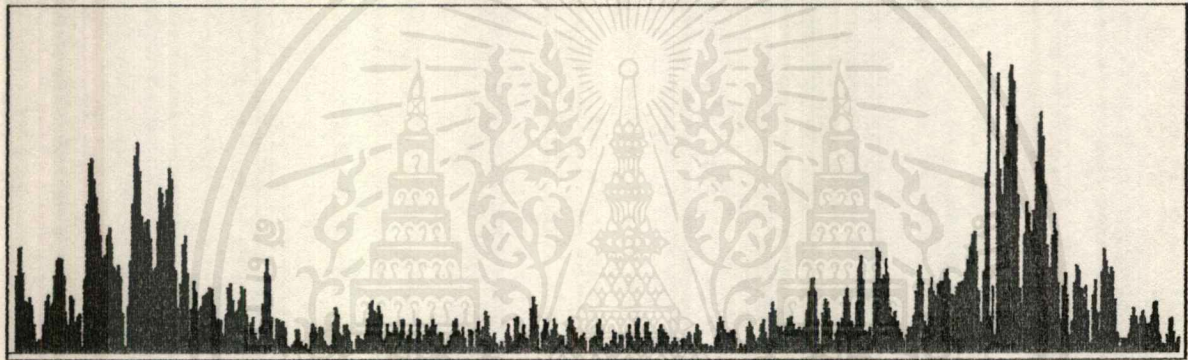
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MAGNITUDE SPECTRUM



รูปที่ 4.11 (a) แสดงสเปกตรัมความถี่เสียงกระเบื้องแตกเฟรมที่สอง

MAGNITUDE SPECTRUM



รูปที่ 4.11 (b) แสดงสเปกตรัมความถี่เสียงกระเบื้องแตกเฟรมที่สาม

MAGNITUDE SPECTRUM



รูปที่ 4.11 (c) แสดงสเปกตรัมความถี่เสียงกระเบื้องแตกเฟรมที่สี่

รูปที่ 4.11 แสดงสเปกตรัมความถี่เสียงกระเบื้องแตกของรูปคลื่นจากรูปที่ 4.1(d)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลอง อุปสรรค และข้อเสนอแนะ

สรุปผลการทดลอง

การทดลอง ทำโครงการนี้ในช่วงแรก ได้ทดลองนำวิธีการ LPC (Linear Prediction coding) มาใช้แต่มีปัญหาที่การโปรแกรมภาษาแอสเซมบลีของไอที TMS32010 ต้องใช้เวลาศึกษาและเขียนโปรแกรมลำบาก จึงได้นำวิธีการวิเคราะห์หา spectrum ของ FFT มาใช้ โดยใช้เครื่อง PC ทดลอง สามารถแสดงสัญญาณรูปคลื่นเสียงในโดเมนเวลา และเก็บไว้นำไปคำนวณ FFT และ plot spectrum ให้ดูที่หน้าจอ นำข้อมูลที่ได้ ไปเก็บเพื่อไว้ทำการเปรียบเทียบซึ่งก็ได้ผลพอสมควร สามารถเปรียบเทียบเสียงที่มี condition เหมือนกันได้ คือถ้าเป็นเสียงชนิดเดียวกัน เช่น แก้วแตกก็สามารถที่จะเปรียบเทียบได้ความเหมือนกันของ pattern ประมาณ 30-80% ถ้าเป็นเสียงที่ต่างกัน เช่น เสียงแก้วแตก กับ เสียงเหรียญตกกระทบพื้น ความเหมือนกันของ pattern อยู่ในช่วง 0-10% ในส่วนของ condition ที่ต้องเหมือนกันในเสียงชนิดเดียวกัน เช่น ระยะเวลาที่เกิดเสียง ความดังของเสียง เสียงก้องหรือไม่ก้อง วัตถุที่ทำให้เกิดเสียง จำนวนข้อมูลเหมาะสมที่นำมาใช้ในการเปรียบเทียบอยู่ในช่วงประมาณ 150 ค่าในส่วนของปัญหาต่างๆ ถ้านำไปใช้งานจริงจะต้องพัฒนาในส่วนนี้ต่อไป

อุปสรรค

1. การกำหนดขอบเขตจุดต้น และจุดปลายของสัญญาณเสียง ก่อนที่จะทำการแปลงโดยการ ใช้ FFT เรายังต้องกำหนดเองอยู่ เพราะสัญญาณเสียงบางเสียงยาวมากและมีแอมพลิจูด ในระดับต่ำ ๆ หลายช่วง

2. ปัญหา ช่วงความยาวของคลื่นเสียง ถึงแม้จะเป็นเสียงเดียวกัน วัตถุที่กระทบกันเป็นชนิดเดียวกัน แต่ความยาวคลื่น ที่เกิดขึ้นแต่ละครั้งไม่เท่ากันเสมอไป เช่น เสียงเหรียญตกกระทบพื้นสองครั้ง ครั้งแรกเกิดเสียง เป็นระยะเวลาสั้น 1 วินาที ครั้งที่สองเกิดเสียงเป็นระยะเวลา 1.5 วินาที เมื่อนำแปลง FFT และเรียงข้อมูลความถี่นำมาเปรียบเทียบกัน ผลที่ได้คือความถูกต้องจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อเสนอแนะ

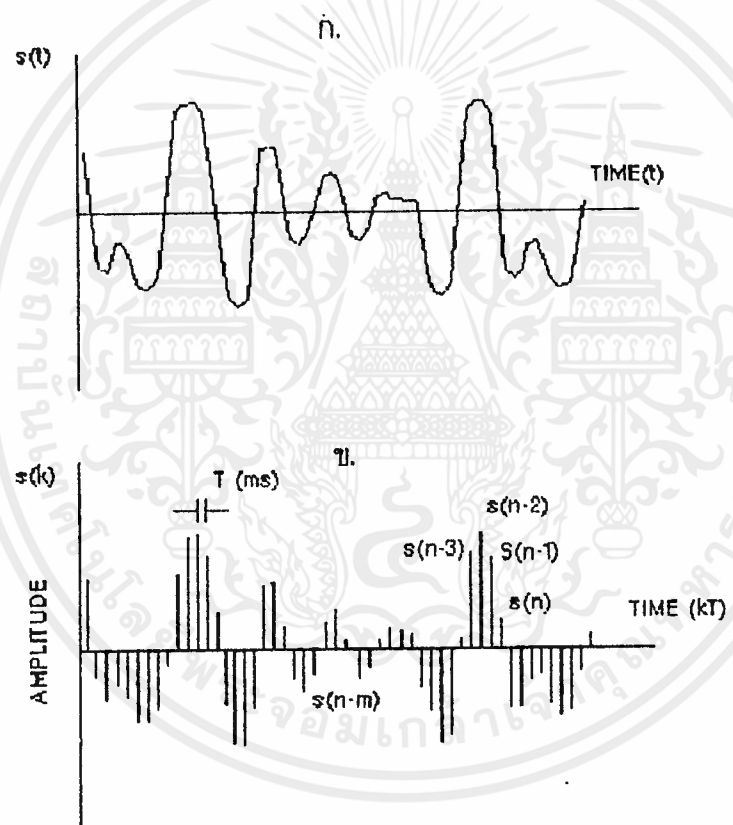
1. เครื่องตรวจจับเสียงนี้ ต้องการตัวประมวลผลที่มีความเร็วสูง ถ้าหากมี chip ประมวลผล ทางคณิตศาสตร์มาช่วย จะทำให้การคำนวณเร็วขึ้นมาก เพราะการแปลง ฟาสต์ฟูเรียร์ เป็นตัวแปรชนิด โพลิตติงพอยท์
2. ในส่วนของภาคขยายสัญญาณเสียงก่อนเข้า A/D ไม่มีการใช้วงจรควบคุมระดับความดังของสัญญาณเสียงอัตโนมัติ (AGC) ถ้าเสียงเกิดขึ้นดังมาก เมื่อตรวจจับเสียงแล้ว จะเกิดการขลิบสัญญาณ เมื่อนำสัญญาณนี้ไปแปลงเป็น Digital แล้วไปประมวลผลจะทำให้เกิดการผิดพลาดได้ เมื่อทำการเปรียบเทียบไฟล์
3. การตรวจจับการเกิดเสียงในโครงงานนี้ยังใช้ โปรแกรมเป็นตัวหน่วงเวลาก่อน การรับเสียง ซึ่งสามารถแก้ไขได้โดยการเพิ่ม Hardware ทำหน้าที่เป็น สวิตซ์ทำงานด้วย เสียงเข้ามาแทน
4. การวิเคราะห์ความถี่ นอกจากการใช้ Fast Fourier Transform แล้วยังสามารถใช้ Hardware ที่เรียกว่าการ Filter แยกความถี่เสียงแต่ละช่วงมาวิเคราะห์ และประมวลผลได้ จะทำให้การทำงานเร็วขึ้น แต่จะทำให้ค่าใช้จ่ายเพิ่มขึ้นด้วย.
5. การพัฒนาโปรแกรมและ board ประมวลผล ในการสั่ง run program ควรจะเป็นการสั่ง load, key และ run program ได้จากทาง pc เพื่อความสะดวก โดยอาจจะให้ ram สองทาง หรืออย่างอื่นอย่างไร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองของเสียงพูดตามวิธีพยากรณ์เชิงเส้น (Linear Prediction)

หลักการของการพยากรณ์เชิงเส้นคือ เราพยายามที่จะคาดการณ์เสียงพูดที่ส่งมาได้ โดยอาศัยความสัมพันธ์เชิงเส้นของข้อมูลเสียงพูดที่ผ่านมา ซึ่งถ้าเราพยายามที่จะทำให้ผลรวมของผลต่างระหว่างเสียงพูดที่ส่งขึ้นมา กับเสียงพูดที่เราคาดการณ์ยกกำลังสอง ในช่วงเวลาหนึ่ง ๆ มีค่าน้อยที่สุด เราจะได้ค่าสัมประสิทธิ์ของการพยากรณ์อยู่ 1 ชุด (ซึ่งสัมประสิทธิ์ของการพยากรณ์นี้ก็คือ ค่าที่จะเป็นตัวถ่วงน้ำหนักในความสัมพันธ์เชิงเส้นของเสียงพูดที่ผ่านมา)



รูปที่ 1 แสดงสัญญาณเสียงพูด ก. สัญญาณต่อเนื่อง ข. สัญญาณที่ส่งได้ (Sampling)

ถ้าเรากำหนดว่า การพยากรณ์สัญญาณ ณ ตำแหน่งเวลาที่ n มีค่าผิดพลาดเป็น $e(n)$ เราจะได้ว่า

$$e(n) = \sum_{i=0}^m a_i s(n-i)$$

$$= s(n) + \sum_{i=1}^m a_i s(n-i)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูผู้สอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งถ้าเรากำหนดให้ $s(n)$ เป็นค่าพยากรณ์และมีค่าเท่ากับ $-\sum_{i=0}^m a_i s(n-i)$
 เราจะได้ว่า $e(n) = s(n) - s(n)$

ใน Z-Transform เราจะได้ว่า

$$E(z) = S(z)A(z)$$

โดย $A(z) = 1 + \sum_{i=0}^m a_i z^{-i}$

หรือ $S(z) = \frac{E(z)}{A(z)}$

ซึ่งจะเห็นได้ว่า แบบจำลองของเสียงพูดตามวิธีการพยากรณ์เชิงเส้นมีความสอดคล้องกับแบบจำลองเสียงพูดของ Fant

ถ้าเรากำหนดให้ เป็นผลรวมของกำลังสองของค่าผิดพลาดในการพยากรณ์ในช่วงเวลาหนึ่ง ๆ จะได้ว่า

$$\begin{aligned} \alpha &= \sum_{n=n_0}^{n_1} e^2(n) \\ &= \sum_{n=n_0}^{n_1} \left[\sum_{i=0}^m a_i s(n-i) \right]^2 \\ &= \sum_{n=n_0}^{n_1} \sum_{i=0}^m \sum_{j=0}^m a_i s(n-i) s(n-j) a_j \end{aligned}$$

โดย n_0 และ n_1 เป็นช่วงของการพยากรณ์

ให้ $c_{i,j} = \sum_{n=n_0}^{n_1} s(n-i) s(n-j)$

จะได้ว่า $\alpha = \sum_{i=0}^m \sum_{j=0}^m a_i c_{i,j} a_j$

เพื่อให้ค่าผิดพลาดในการพยากรณ์น้อยที่สุดเราจะต้องทำการ differentiate

ค่า เทียบกับ a_i แล้วกำหนดให้เท่ากับ 0 ซึ่งจะได้ว่า เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ประโยชน์ในทางวิชาการเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วิธี AutoCorrolation จะกำหนดให้ $n_0 = -\alpha$ และ $n_1 = \alpha$
 โดยให้ $s(n) = 0$ สำหรับ $n < 0$ และ $n > N$ ซึ่งวิธีการนี้ทำให้การหาค่า $c_{i,j}$
 ง่ายขึ้นโดย

$$\begin{aligned}
 c_{i,j} &= \sum_{n=-\infty}^{\infty} s(n-1)s(n-j) \\
 &= \sum_{n=-\infty}^{\infty} s(n)s(n+1i-j1) \\
 &= \sum_{n=0}^{n-1-i-j} s(n)s(n+1i-j1) \\
 &= r(1i-j1)
 \end{aligned}$$

ทำให้เราได้ AutoCorrolation Matrix ดังนี้

$$\begin{bmatrix}
 r(0) & r(1) & r(2) & \dots & r(m-1) \\
 r(1) & r(0) & r(1) & \dots & r(M-2) \\
 \cdot & \cdot & \cdot & \dots & \cdot \\
 \cdot & \cdot & \cdot & \dots & \cdot \\
 \cdot & \cdot & \cdot & \dots & \cdot \\
 r(M-1) & r(M-2) & r(M-3) & \dots & r(0)
 \end{bmatrix}
 \begin{bmatrix}
 a_1 \\
 a_2 \\
 \cdot \\
 \cdot \\
 \cdot \\
 a_M
 \end{bmatrix}
 =
 \begin{bmatrix}
 r(1) \\
 r(2) \\
 \cdot \\
 \cdot \\
 \cdot \\
 r(m)
 \end{bmatrix}$$

ซึ่งแมทริกซ์นี้เป็น toepnitz แมทริกซ์ เพราะสมมาตรกัน และทุกตัวในแนว
 เส้นทะแยงมีค่าเท่ากัน

$$\frac{\alpha}{n_k} = 0 = 2 \sum_{t=0}^m a_t c_{jk}$$

แต่ $a_0 = 1$ ดังนั้นเราจะได้สมการอยู่ชุดหนึ่งคือ

$$\sum_{t=1}^m a_t c_{jk} = -c_{0k} \quad ; k=1,2,3,\dots,M$$

ถ้าเราถอดสมการชุดนี้ออกมา จะได้ค่า สัมประสิทธิ์ของการพยากรณ์

($a_i \quad ; i=1,2,3\dots M$) ซึ่งมีวิธีการในการหาอยู่ 2 วิธี คือ วิธี Covariance และ วิธี AutoCorrelation

สมมติว่าสัญญาณเสียงพูด ถูกสุ่มมา N จุด โดยมีค่าเป็น $s(0), s(1), \dots, s(n-1)$

สำหรับค่า n_0 และ n_1 ที่จะกำหนดนั้นเมื่ออยู่ 2 แบบใหญ่ ๆ คือ

1. วิธี Covariance จะกำหนดให้ $n_0 = M$ และ $n_1 = N-1$ ดังนั้น

ช่วงของการพยากรณ์โดยวิธีนี้ จะอยู่ในช่วง $[M, N-1]$ ซึ่งจะได้ชุดของสมการเป็น

$$\sum_{t=1}^m a_t c_{tk} = -c_{0k} \quad ; k=1,2,3,\dots,M$$

และ $c_{tk} = \sum_{n=m}^{n-1} s(n-i)s(n-k)$ ซึ่งเขียนเป็นแมทริกซ์ได้ดังนี้

$$\begin{bmatrix} C_{11} & C_{12} & C_{13} & \dots & C_{1M} \\ C_{21} & C_{22} & C_{23} & \dots & C_{2M} \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \cdot & \dots & \cdot \\ C_{M1} & C_{M2} & C_{M3} & \dots & \cdot \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ \cdot \\ a_M \end{bmatrix} = - \begin{bmatrix} C_{10} \\ C_{20} \\ \cdot \\ \cdot \\ \cdot \\ C_{M0} \end{bmatrix}$$

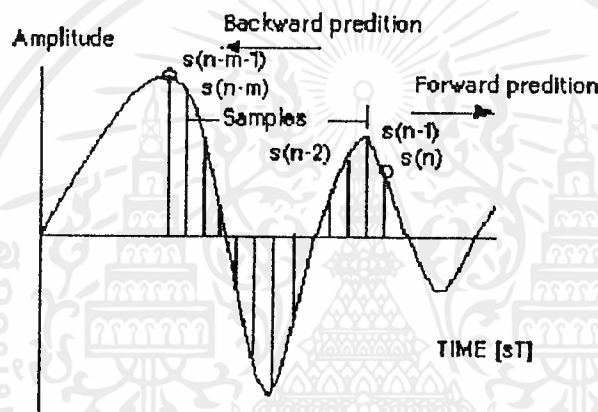
แต่ $c_{ij} = c_{ji}$ ดังนั้น Covariance Matrix จึงมีคุณสมบัติสมมาตร

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ การ نسخ หรือการเผยแพร่โดยไม่ได้รับอนุญาตจะถือว่าผิดกฎหมาย
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทคนิคของ PARCOR

PARCOR เป็นเทคนิคที่ใช้หลักการของการพยากรณ์เชิงเส้น ทั้งด้านหน้า และด้านหลังของสัญญาณ ณ ตำแหน่งที่จะพยากรณ์ ซึ่งทำให้การคำนวณหาสัมประสิทธิ์ของการพยากรณ์ (a_i) รวดเร็วขึ้น และสามารถนำหลักการของการกระทำซ้ำ (Recursive) ซึ่งจะทำให้สะดวกในการเขียนโปรแกรม

หลักการของ PARCOR จะพิจารณาได้จากสัญญาณเสียงที่ส่งมาได้ดังรูปข้างล่าง



รูปที่ 2 แสดงหลักการของ PARCOR

ถ้าสัญญาณ x_m เป็นสัญญาณที่เราตั้งการพยากรณ์ เราจะได้ค่าผิดพลาดในการพยากรณ์ไปข้างหน้า ($x_{m+}(n)$) เป็น

$$\begin{aligned} x_{m+}(n) &= x(n) - \left[- \sum_{i=1}^m a_{m,i} x(n-i) \right] \\ &= \sum_{i=0}^m a_{m,i} x(n-i) \quad \text{โดย } a_{m,0} = 1 \end{aligned}$$

และจะได้ค่าผิดพลาดของการพยากรณ์ไปข้างหลัง ($x_{m-}(n)$) เป็น

$$\begin{aligned} x_{m-}(n) &= x(n-m-1) - \left[- \sum_{i=0}^m b_{m,i} x(n-i) \right] \\ &= \sum_{i=1}^{m+1} b_{m,i} x(n-i) \quad \text{โดย } b_{m,m+1} = 1 \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้ากำหนดให้ ผลรวมกำลังสองของค่าผิดพลาด ในการพยากรณ์ไปข้างหน้า เป็น α_m และผลรวมกำลังสองของค่าผิดพลาด ในการพยากรณ์ไปข้างหลังเป็น β_m เงื่อนไขที่จะทำให้ การพยากรณ์ผิดพลาดน้อยที่สุดคือ

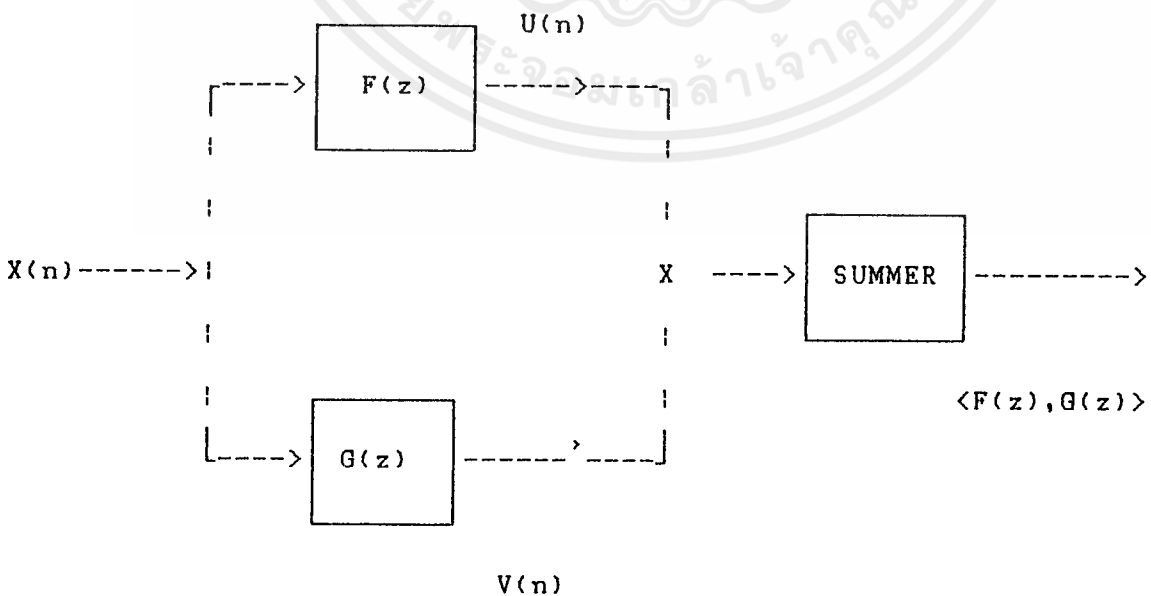
$$\frac{\partial \alpha_m}{\partial a_{m,t}} = 0 \quad \text{และ} \quad \frac{\partial \beta_m}{\partial b_{m,t}}$$

โดย
$$\alpha_m = \sum_{n=-\infty}^{\infty} [X_m + (n)]^2$$

และ
$$\beta_m = \sum_{n=-\infty}^{\infty} [X_m - (n)]^2 \quad m=1,2,\dots,M; i=1,2,\dots,m$$

หลักการของ inner products และ orthogonality

หลักการของ inner products ก็คือ การที่ filter 2 ตัว มีข้อมูล (input) เดียวกันในเวลาเดียวกัน เข้ามา และจะได้ผลลัพธ์ออกมา 2 ค่าหลังจากผลลัพธ์ทั้งสอง (ในเวลาเดียวกัน) จะคูณกัน และผลคูณนั้นจะถูกรวมกันในช่องเวลาหนึ่ง ๆ ขบวนการทั้งหมดนี้เรียกว่า inner products



รูปที่ 3 แสดงถึงหลักการของ inner products

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์หรือการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3 แสดงให้เห็นว่า ผลลัพธ์ของ inner products ของ filter $F(z)$ และ $G(z)$ มีค่าเท่ากับ

$$\langle F(z), G(z) \rangle = \sum_{n=n_0}^{n_1} u(n)v(n)$$

และในทำนองเดียวกันจะได้ว่า

$$\langle Z^{-i}, Z^{-j} \rangle = \sum_{n=n_0}^{n_1} X(n-i)X(n-j)$$

ซึ่งตามวิธีการแบบ Covariance จะได้ว่า

$$\langle Z^{-i}, Z^{-j} \rangle = c_{i,j} = c_{j,i}$$

และตามวิธีการแบบ AutoCorrelation จะได้ว่า

$$\langle Z^{-i}, Z^{-j} \rangle = r(i-j) = r(j-i)$$

สมมุติว่า $X_m(n)$ และ $X_{m+1}(n)$ เป็นผลลัพธ์ของ filter 2 ตัวคือ

$A_m(z)$ และ $B_m(z)$ โดย

$$A_m(z) = \sum_{i=0}^{m+1} a_{m,i} Z^{-i} \quad ; a_{m,0} = 1 \dots (ก)$$

และ

$$B_m(z) = \sum_{i=0}^{m+1} b_{m,i} Z^{-i} \quad ; b_{m,m+1} = \dots (ข)$$

โดยอาศัยหลักการของ inner products เราจะได้ว่า

$$\alpha_m = \langle A_m(z), A_m(z) \rangle = |A(z)|^2$$

$$\beta_m = \langle B_m(z), B_m(z) \rangle = |B(z)|^2$$

และจากเงื่อนไขของการพยากรณ์ที่ต้องการให้ ผลรวมของการ

ผิดพลาดยกกำลังสองมีค่าน้อยที่สุด สามารถพิสูจน์ได้ว่า

$$\langle A_m(z), Z^{-j} \rangle = 0 \quad ; j=1, 2, \dots, m$$

และ

$$\langle B_m(z), Z^{-j} \rangle = 0 \quad ; j=1, 2, \dots, m$$

ผลที่ได้ก็คือหลักการของ orthogonality

โดยใช้หลักการของ inner products และ Orthogonality ทำให้ได้

ผลสรุปว่า

$$\langle A_m(z), Z^{-j} \rangle = \sum_{n=n_0}^{n_1} X_m(n)X(n-j) = 0$$

และ $\langle B_m(z), Z^{-j} \rangle = \sum_{n=n_0}^{n_1} X_{m+1}(n)X(n-j) = 0$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก (ก) จะได้ว่า $A_{m-1}(z)$ เป็นโพลิโนเมียลของ z^{-1}

โดย $i=1,2,\dots,(m-1)$ และจาก (ข) จะเห็นว่า $B_{m-1}(z)$ เป็นโพลิโนเมียลของ z^{-1}

โดย $i=1,2,\dots,m$ ดังนั้น $a_m(z)$ จึงสามารถหาได้จาก $A_{m-1}(z)$ กับ $B_{m-1}(z)$

โดยความสัมพันธ์ดังนี้

$$A_m(z) = A_{m-1}(z) + k_{m,m-1} B_{m-1}(z)$$

จากหลักการของ orthogonality เราจะได้ว่า

$$k_m = \frac{-1}{\beta_{m-1}} \langle A_{m-1}(z), B_{m-1}(z) \rangle$$

$$= \frac{-1}{\beta_{m-1}} \sum_{n=0}^{\beta_{m-1}} X_{m-1}^{(n)} + (n) X_{m-1}^{-(n)}$$

และ

$$\alpha_{m+1} = \alpha_m - k_{m+1} 2 \beta_m$$

โดยตัวแปร k เรียกว่า partial correlation coefficient หรือสัมประสิทธิ์การสะท้อนกลับ (Reflection Coefficient) ซึ่งจะสอดคล้องกับ สัมประสิทธิ์ของการสะท้อนกลับ ที่หามาจากแบบจำลองของทางเดินของเสียง จากกล่องเสียงถึงริมฝีปาก

วิธีในการหาค่าพารามิเตอร์

การหาค่าพารามิเตอร์ของเสียง โดยวิธีการพยากรณ์เชิงเส้นทั้งสองวิธีคือ แบบ AutoCorrelation และแบบ Covariance มีกรรมวิธีการหาเหมือนกัน จะแตกต่างกันที่ตรงการกำหนดค่า n_0 และ n_1 ซึ่งได้กล่าวไว้ในหัวข้อ ค่าพารามิเตอร์ที่ได้ก็คือสัมประสิทธิ์การพยากรณ์ (a_1), สัมประสิทธิ์ของการสะท้อนกลับ (k_1) รวมทั้งค่าผิดพลาดในการพยากรณ์ยกกำลังสอง (α)

ขั้นที่ 1 คำนวณ Covariance หรือ AutoCorrelation แมทริกซ์

จากวิธี Covariance สัมประสิทธิ์ $c_{i,j}$ หาได้จาก

$$c_{i,j} = \sum_{n=i}^{n-1} X(n-i)X(n-j)$$

$j = 0, 1, 2, \dots, i$; $i = 0, 1, 2, \dots, M$ เมื่อกำหนดเงื่อนไขในช่วง $n = M$ ถึง $n=N-1$ ค่า $c_{i,j}$ ที่ต้องคำนวณทั้งหมด จะมีจำนวนเท่ากับ $M(M+1)/2$ ด้วยวิธี AutoCorrelation สัมประสิทธิ์ $c_{i,j}$ กลายเป็น $r(|i-j|)$

$$c_{i,j} = r(|i-j|) = r(k) \quad ; k = 0, 1, 2, \dots, M$$

$$r(k) = \sum_{n=0}^{n-1-k} X(n)X(n+k)$$

ขั้นที่ 2 หาค่าเริ่มต้น

สำหรับค่าเริ่มต้นกรณีที่ $m=0$ จะได้ว่า

$$a_{00} = 1 \text{ และ } b_{01} = 1$$

ซึ่งจากหลักการ inner products จะได้ว่า

$$\alpha_0 = \beta_0 = r(0)$$

และที่ $m = 1$ จะได้ว่า

$$\begin{aligned} A_1(z) &= A_0(z) + k_1 \beta_0(z) \\ &= 1 + k_1 z^{-1} \end{aligned}$$

เมื่อเทียบสัมประสิทธิ์แล้วจะได้ว่า

$$a_{10} = 1, a_{11} = k_1$$

และจะได้เอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 หาค่าพารามิเตอร์เมื่อ m เพิ่มขึ้น
ตามวิธี AutoCorrelation เราจะได้ว่า

$$\beta_m = \alpha_m$$

$$b_{m+1} = a_{m, m+1} - \alpha_m$$

และ
$$k_{m+1} = \frac{-1}{\alpha_m} \sum_{i=0}^m b_{m+1-i} a_{m+1-i}$$

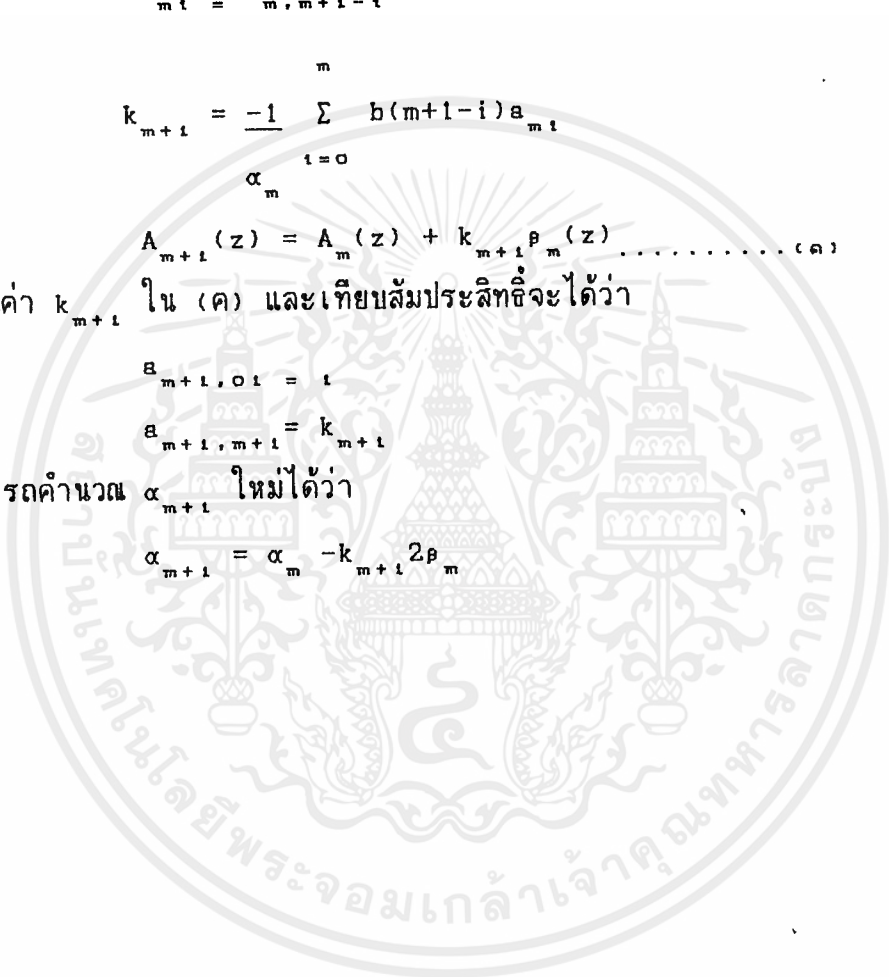
โดย $A_{m+1}(z) = A_m(z) + k_{m+1} \beta_m(z) \dots \dots \dots (ค)$
ดังนั้นแทนค่า k_{m+1} ใน (ค) และเทียบสัมประสิทธิ์จะได้ว่า

$$a_{m+1, 0} = 1$$

$$a_{m+1, m+1} = k_{m+1}$$

และสามารถคำนวณ α_{m+1} ใหม่ได้ว่า

$$\alpha_{m+1} = \alpha_m - k_{m+1} 2\beta_m$$



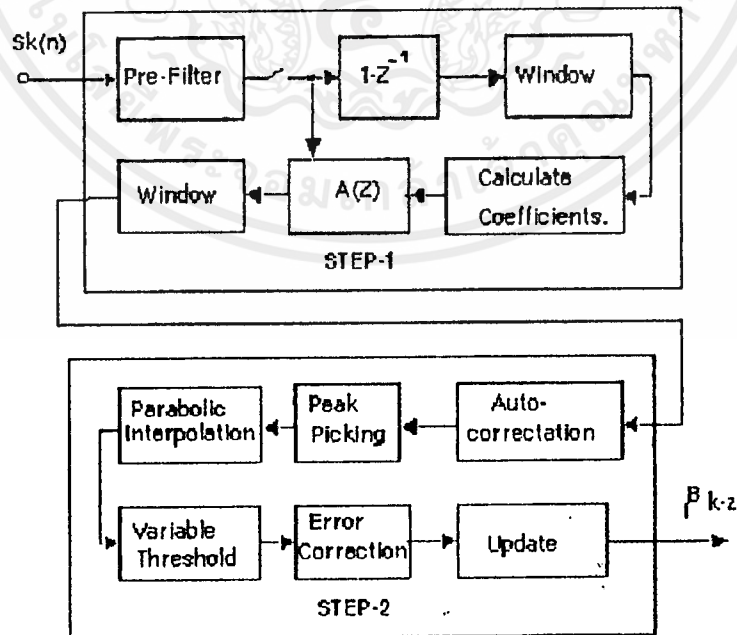
วิธีในการหาค่าความถี่มูลฐาน

โดยอาศัยคุณสมบัติของแบบจำลองเสียงพูด ตามแบบวิธี Linear Prediction เราสามารถที่จะหาค่า Pitch Period ของเส้นเสียงได้ โดยใช้ คุณสมบัติของรูปแบบการวิเคราะห์ดังในหัวข้อที่ 5.3 ได้ว่า

$$E(z) = S(z)A(z)$$

สำหรับการวิจัยในครั้งนี้ได้ใช้วิธีการที่เรียกว่า SIFT (Simplified inverse filter tracking)

โดยหลักการก็คือ เราจะต้องหาค่า a_n ของ inverse filter เพื่อแปลงสัญญาณเสียงพูดให้เป็นสัญญาณที่เส้นเสียง $e(t)$ ซึ่งก็คือ error signal นั่นเอง วิธีการของ SIFT ได้แบ่งเป็นสองขั้นตอนคือ ขั้นตอนการหาค่า error signal และขั้นตอนการตัดสินใจ และประมาณค่า Pitch Period จาก error signal ที่ได้ ดังรูป





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรและการทำงานของส่วนประมวลผล TMS32010

วงจรในส่วนนี้เป็นการทำงานโดยแยกระบบออกมา โดยใช้ IC TMS32010 เป็นตัวประมวลผล จะทำให้การทำงานเร็วขึ้น และประมวลผลแบบ REAL TIME ซึ่งเป็นส่วนสำคัญของการทำงานนี้

การทำงานของวงจรส่วนนี้ สัญญาณเสียงที่ได้จะผ่าน วงจร BANDPASS FILTER เพื่อกรองเอาส่วนความถี่ที่ไม่ต้องการออกไป ซึ่งประโยชน์ที่ช่วยลดข้อมูลที่ได้ และเวลาในการประมวลผลออกไป จากส่วน BANDPASS IC 3 ก็เข้าสู่ IC 2 เพื่อปรับระดับของแรงดันที่จะไปแปลงเป็นดิจิตอลให้ได้ 0-5 V หรือเพื่อเป็น BUFFER ของ ADC ซึ่งในที่นี้ใช้ IC CA3306 เป็นเป็น FLASH ADC 6 bit ข้อดีคือความเร็ว ข้อเสียคือข้อมูลที่ได้มีความความละเอียดต่ำ

ในส่วน TMS32010 จากสถาปัตยกรรม จะยอมให้มีการใช้เรียกหน่วยความจำภายนอกได้ 4K WORD คำสั่งในการเรียกข้อมูลจากหน่วยความจำภายนอกคือ TBLR ,TBLW ซึ่งในวงจรเราใช้เป็นหน่วยความจำโปรแกรม 2K WORD และหน่วยความจำข้อมูล 2K WORD โดยใช้การมัลติเพล็กซ์แอดเดรส ของหน่วยความจำข้อมูลไว้ ในส่วนของการเรียกใช้อุปกรณ์ I/O TMS32010 สามารถ DECODE I/O ได้อย่างละ 8 ตัว คือเป็นหน่วยอินพุตได้ 8 ตัว และหน่วยเอาพุตได้อีก 8 ตัว สัญญาณที่ใช้ได้จาก A0-A2, DEN, WE ซึ่งดูได้จากทฤษฎีในบทที่ 3 ในวงจรเราจะเรียก ใช้งาน ADC และ DAC ส่วนหน่วยความจำข้อมูลก็สามารถให้ TMS32010 มองหน่วยความจำข้อมูลเป็น I/O ได้ ผ่านทางนี้ โดยใช้คำสั่ง IN หรือ OUT

การต่อหน่วยความจำภายนอกสามารถต่อได้ 4K WORD ในการเรียกทั้ง ROM และ RAM ภายใต้คำสั่ง TBLR, TBLW สัญญาณที่ใช้คือ MEN, DEN และ WE การ DECODE ROM กับ RAM ให้สัญญาณจาก A11 สัญญาณ MEN และ A11 เป็น LOW จะเป็นการเลือก ROM ถ้า MEN เป็น LOW แต่ A11 เป็น HIGH ก็จะเป็นการเรียก RAM

ในอีกโมด การเรียกเข้าหน่วยความจำโปรแกรมจะเหมือนเดิม แต่ในส่วนของหน่วยความจำข้อมูลจะเรียกเป็น I/O โดยใช้สัญญาณ DEN และ WE ทั้งสองสัญญาณเป็น LOW ซึ่งจะ DECODE I/O ใน RAM ซึ่งเป็น PORT 0 ADC PORT 1 DAC PORT 2

ข้อแตกต่างในการทำงานหรือเข้าหาข้อมูลจาก RAM ทั้ง 2 โมดคือ ความเร็วในส่วนของโมดแรก เป็นการให้คำสั่ง TBLR/W ซึ่ง TMS 32010 ออกแบบให้มีการเรียกใช้หน่วยความจำในส่วนภายใน ซึ่งมีประโยชน์ในการทำงานด้านความเร็ว ในส่วน RAM ภายในมีเพียง 144 WORD ซึ่งน้อย จึงต้องให้หน่วยความจำภายนอก การเรียกแบบ TBLR/W จะใช้เวลา 3 CLOCK PULSE แต่ถ้าเรียกเป็นข้อมูลในโมดแบบ I/O จะใช้ 2 CLOCK PULSE

ในส่วนอัตราการแปลงสัญญาณเข้ามา กำหนดที่ IC1 74HC4060 ซึ่งเป็น IC IC ทารจะได้อัตราการแปลงค่าต่างๆ ซึ่งจะหาความถี่ของ CRYSTAL ด้วยค่าต่าง ๆ ตามสเปคของ IC ที่กำหนด ความถี่ที่ได้จะนำมาเข้าขา INT เพื่ออินเตอร์รัพท์การทำงานของ TMS32010 คือจะไปอ่านข้อมูลจาก ADC และเขียนข้อมูลออกที่ DAC



การหาค่าพารามิเตอร์ของเสียง โดยวิธีการพยากรณ์เชิงเส้นทั้งสองวิธีคือ แบบ AutoCorrelation และแบบ Covariance มีกรรมวิธีการหาเหมือนกัน จะแตกต่างกันก็ตรงการกำหนดค่า n_0 และ n_1 ซึ่งได้กล่าวไว้ในหัวข้อ ค่าพารามิเตอร์ที่ได้ก็คือสัมประสิทธิ์การพยากรณ์ (a_k), สัมประสิทธิ์ของการสะท้อนกลับ (k_k) รวมทั้งค่าผิดพลาดในการพยากรณ์ยกกำลังสอง (α)

ขั้นที่ 1 คำนวณ Covariance หรือ AutoCorrelation แมทริกซ์

กำหนดให้ในหนึ่งเฟรมของการวิเคราะห์มีจำนวนสัญญาณสุ่มเท่ากับ N แซมเปิล และฟิลเตอร์มีจำนวนออเดอร์เท่ากับ M ตามวิธี อัดสหัสสัมพันธ์ ค่าอัดสหัสสัมพันธ์ที่ใช้ในการคำนวณขั้นต่อไปจะมีจำนวนเท่ากับ $M+1$ ค่า

$$r(k) = \sum_{n=0}^{N-1-k} X(n)X(n+k) \quad ; k=0, 1, 2, \dots, M$$

โดย $X(n)$; $n = 0, 1, 2, \dots, N-1$ คือสัญญาณเสียงที่สุ่มมาในหนึ่งเฟรม

จากวิธี Covariance สัมประสิทธิ์ $c_{i,j}$ หาได้จาก

$$c_{i,j} = \sum_{n=i}^{n-1} X(n-i)X(n-j)$$

$j = 0, 1, 2, \dots, i$; $i = 0, 1, 2, \dots, M$ เมื่อกำหนดเงื่อนไขในช่วง $n = M$ ถึง $n=N-1$ ค่า $c_{i,j}$ ที่ต้องคำนวณทั้งหมด จะมีจำนวนเท่ากับ $M(M+1)/2$ ด้วยวิธี AutoCorrelation สัมประสิทธิ์ $c_{i,j}$ กลายเป็น $r(|i-j|)$

$$c_{i,j} = r(|i-j|) = r(k) \quad ; k = 0, 1, 2, \dots, M$$

ขั้นที่ 2 หาค่าเริ่มต้น

สำหรับค่าเริ่มต้นกรณีที่ $m=0$ จะได้ว่า

$$a_{00} = 1 \text{ และ } b_{01} = 1$$

ซึ่งจากหลักการ inner products จะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\alpha_0 = \beta_0 = r(0)$$

และที่ $m = 1$ จะได้ว่า

$$\begin{aligned} A_1(z) &= A_0(z) + k_1 \beta_0(z) \\ &= 1 + k_1 z^{-1} \end{aligned}$$

เมื่อเทียบสัมประสิทธิ์แล้วจะได้ว่า

$$a_{10} = 1, a_{11} = k_1$$

และจะได้

$$\alpha_1 = \alpha_0 - k_1^2 \beta_0$$

ขั้นที่ 3 หาค่าพารามิเตอร์เมื่อ m เพิ่มขึ้น
ตามวิธี AutoCorrelation เราจะได้ว่า

$$\beta_m = \alpha_m$$

$$b_{m1} = a_{m, m+1-i}$$

และ

$$k_{m+1} = \frac{-1}{\alpha_m} \sum b_{(m+1-i)a_{m1}}$$

โดย

$$A_{m+1}(z) = A_m(z) + k_{m+1} \beta_m(z) \dots \dots \dots (ค)$$

ดังนั้นแทนค่า k_{m+1} ใน (ค) และเทียบสัมประสิทธิ์จะได้ว่า

$$a_{m+1, 01} = 1$$

$$a_{m+1, m+1} = k_{m+1}$$

และสามารถคำนวณ α_{m+1} ใหม่ได้ว่า

$$\alpha_{m+1} = \alpha_m - k_{m+1}^2 \beta_m$$

ขั้นตอนทั้งสามนี้ถูกนำมาเขียนเป็นโปรแกรมย่อยชื่อ AUTO เพื่อใช้ในการวิเคราะห์เสียงพูดพารามิเตอร์ของฟิลเตอร์ในแต่ละเฟรม จำนวนแซมเปิ้ลสัญญาณที่ใช้วิเคราะห์ในหนึ่งเฟรมกำหนดที่ตัวแปร N และจำนวนออร์เตอร์ของฟิลเตอร์กำหนดที่ตัวแปร M จากตารางข้างล่างแสดงชื่อตัวแปรในโปรแกรมเทียบกับสัญลักษณ์ในสมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแอสเซมบลีของ TMS32010

* APPENDIX I

* AMDF PITCH ESTIMATION VER. I

*

* compute a pitch estimate from one out of 60 values witch
 * are stored in a table in internal RAM. Assumes voice data
 * to be stored in external RAM accessed using TBLR

```

INIT      ZALS      BIG-NUM-L,0
          ADDH      BIG-NUM-H,0
          SACL      MIN-AMDF-L,0      ;MIN-AMDF maintain min AMDF
          SACH      MIN-AMDF-H,0      ; value,initittosomething
          LACK      #60                ; large
          SACL      NUM-PITCHS,0      ;Num-Pitches is loop counter
          LANK      ARO,#P-TBL-ADDR ;ARO points to pitch table

LOOP-1    LACK      #S-ADDR            ;Initialize pointer to voice
          SACL      S-PTR,0            ; data s[n_ and s[n-T_

          SUB      *,ARO,0             ;Compute pointer to s[n-T_
          SACL      ST-PTR,0           ; by subtracting away crrent
          ZAC                                     ; pitch
          SACL      AMDF-L,0           ;Initialize accumulated AMDF
          SACH      AMDF-H,0           ; value
          LARK      ARO,#NUM-SAMPLES;ARO is loop counter
  
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOOP-2  ZALS    S--PTR,0      ;Load ACC w/ pointer to s[n]
        TBLR    S              ;Read s[n]
        ADD     THREE,0       ;Update pointer skipping
        SACL    S--PTR,0      ;three voice samples
        ZALS    ST--PTR,0     ;Do same w/ s[n-T]
        TBLR    ST
        ADD     THREE,0
        SACL    ST--PTR,0

        ZALS    S,0           ;Compute /s[n]-s[n-T]/
        SUB     ST,0
        ARS
        ADD     AMDF-L,0       ;Update AMdf value
        ADD     AMDF-H,15
        SACL    AMDF-L,0
        SACH    AMDF-H,0
        MAR     *-,ARI        ;if not finished w/ current
        BNAZ    LOOP-2        ; pitch value,loop
        ZALS    MIN-AMDF-L,0   ;Compare current AMDF value
        ADDH    MIN-AMDF-H,0   ; with previous minimum
        SUB     AMDF-L,0
        SUB     AMDF-H,15      ;if current is larger,loop
        BLZ     SAME          ; to next pitch value
        ZALS    AMDF-L,0       ;If current AMDF is smaller
        ADDH    AMDF-H,0       ; make it the new minimum
        SACL    MIN-AMDF-L,0
        SACH    MIN-AMDF-H,0
        ZALS    *,ARO,0
        SACL    T,0           ;Save the pitch value

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SAME    MAR    *+,ARO           ;Loop to next pitch value
        ZALS   NUM-PITCHS,0
        SUB    ONE,0
        SACL   NUM-PITCHS,0
        BGEZ   LOOP-1

```

* APPENDIX II

* PITCH PREDICTION COEFFICIENT VER.1

*

* Compute the pitch prediction coefficient. Assumes voice is stored
 * in external RAM access using TBLR

```

INIT    ZAC           ;Initialize numerator and
        SACL   NUM-L,0       ; denominator
        SACH   NUM-H,0
        SACL   DEN-L,0
        SACH   DEN-H,0
        LACK   #S-ADDR      ;Initialize pointer to s[n_
        SACL   S-PTR,0      ; and s[n-T]
        SUB    T,0
        SACL   ST-PTR,0
        LACK   ARO,#N       ;Initialize loop counter ARO

LOOP    ZALS   S-PTR
        TBLR   S           ;Read s[n_]
        ADD    ONE,0
        SACL   S-PTR,0

```

เอกสารนี้เป็นเอกสารที่มอบไว้สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ZALS      ST-PTR,0
TBLR      ST          ;Read s[n-T]
ADD       ONE
SACL      ST-PTR
ZALS      NUM-L,0
ADDH      NUM-H,0
LT        ST
MPY       S
APAC
SACL      NUM-L,0      ;Update numerator
SACH      NUM-H,0
ZALS      DEN-L,0
ADDH      DEN-H,0
MPY       ST
APAC
SACL      DEN-L,0     ;Update denominator
SACH      DEN-H,0
MAR       *-,ARO
BNAZ      LOOP
CALL      DIVIDE      ;Perform divide in
ZACH      QUOTIENT    ; a subroutine
SACH      ALPHA      ;Store result

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

* APPENDIX III
* COMBINED 1ST RESIDUAL AND AUTOCORRELATION COMPUTATION
*
* Computes 5 autocorrelation values along with the 1st residual
* direct from the voice signal. Assumes that voice is stored in
* external RAM and that it is accessed using TBLR

```

```

INIT      LACK      #S-ADDR      ;Initialize pointers to voice
          SACL      S-PTR,0      ; data
          SUB       T,0
          SACL      ST-PTR,0
          LACK      #N           ;Initialize main loop counter
          SACL      COUNT,0
          LARK      ARO,#R-ADDR
          LACK      ARI,#ORDER

LOOP-4    SACL      *+,ARO       ;Initialize autocorrelation
          MAR       *-,ARI       ; value to zero
          BNAC      LOOP-4

LOOP-1    ZALS      S-PTR,0
          TBLR      S             ;Read s[n]
          SUB       ONE,0
          SACL      S-PTR,0
          LT        ALPHA
          ZALS      ST-PTR,0
          TBLR      ST            ;Read s[n-T]
          SUB       ONE,0
          SACL      ST-PTR,0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ZALS      S                ;Compute eI[n] = s[n]- s[n-T]
MPY       ST
SPAC                                ;Place eI[n] value on a First-in
LARK      ARO,#e-ADDR ; First-out, stack which retains
SACL      #,0,ARO      ; five most previous residual
LT        #,ARO        ; values
LARK      #ORDER      ;Set up loop to compute
SACL      COR-COUNT,0 ; correlation values
LARK      ARI,# -ADDR ;Update, R0 through R4 values

LOOP-2   ZALS      #+,ARI,0
        ADDR      #-,ARI,0
        MPY       #-,ARI,0
        APAC
        SACL      #+,ARI,0
        SACH      #+,ARI,0
        ZALS      COR-COUNT,0
        SUB       ONE
        SACL      COR-COUNT,0
        BNEZ      LOOP-2
        LARK      ARO,#e4-ADDR;Reorder values in the residual
        LARK      ARI,#ORDER ; FIFO

LOOP-3   DMDV      #-,ARO,0
        MAR       #-,ARI
        BNAC      LOOP-3
        ZALS      COUNT,0      ;Update loop counter
        SUB       ONE,0
        SACL      COUNT,0
        BNEZ      LOOP-1 ;Re-do loop

```

เอกสารนี้เป็นเอกสารที่เผยแพร่ไว้สำหรับกองคลังงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

* APPENDIX IV

* LEROUX-GUEGUEN FOR COMPUTING LPC PARCOR COEFFICIENTS

```

LPC-INIT LACK      #4                ;Set up pointer to transfer
SACL     INIT-COUNTER,0 ; autocorrelation values to
                                ; init recursion

LARK     ARO, #ADDR-RO ;Point to R[0]
LARK     AR1, #ADDR-EO ;Point to e"[0]

INIT-LP-1 ZALS     *+,ARO            ;Auto correlation are double
SACH     *+,AR1      ; word (use upper word only)
MAR      *+,ARO,     ; Skip a word
ZALS     INIT-COUNTER
SUB      ONE,0
SACL     INIT-COUNTER
BNEZ     INIT-LP-1

LACK     #3          ;Load the other three autoco.
SACL     INIT-COUNTER ; values
LARK     ARO,#ADDR-R1 ;Aux registers pointed to data
LARK     AR1,#ADDR-E-1

INIT-LP-2 ZALS     *+,ARO
SACL     *+,AR1,0
MAR      *+,ARO      ;Skip word for doble word
ZALS     INIT-COUNTER ; autocorralation
SACL     INIT-COUNTER,0
BNEZ     INIT-LP-2

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INIT-PTRS LARK      ARO,#ADDR E-ARRAY ;init pointer to
SAR      ARO, E-ARRAY-START ; e'n'[-p+n+2]
LARK     ARO, #ADDR-k0
SAR     ARO, K-PTR
LARK     #7 ;init number of iterations
SACL    NUM-ITERATIONS,0
LARK     ARO, #ADDR-ENO ;init pointer to e'n'[0]
SAR     ARO, ENO-PTR
LARK     ARO, #ADDR-EN1 ;init pointer to e'n'[n+1]
SAR     ARO, EN1-PTR

LG-REC  LAR      ARO, EN1-PTR
ZALS    *+, ARO
SACL    NUMERATOR
SAR     ARO, ENT-PTR
LAR     ARO, FNO-PTR
ZALS    *-, ARO
SACL    DENOMINATOR
CALA    DIVIDE
ZALS    QUOTIENT
LAR     ARO, K-PTR
SACL    *, ARO, 0
LT      *+, ARO

SAR     ARO, K-PTR
LARK     ARO, #ADDR-E-END ;init pointers for from
LAR     AR1, E-ARRAY-START ; data
SAR     AR1, CROSS-PTR
LARK     AR1, #ADDR-SCR-END ;init ptr to put away data

```

```

SAR      ARI, TO-PTR
ZALS     NUM-ITERATIONS
BEZ      DONE
SACL     LG-COUNTER

LG LOOP  ZALS     *+, ARO, 0           ;e'n[i] accumulator
        LAR      ARI, CROSS-PTR
        MPY     *+, ARI           ;k[n]e'n[n+1-i]
        SPAC
        SAR     ARI, CROSS-PTR
        LAR     ARI, TO-PTR
        SACL    *+, ARI, 0
        SAR     ARI, TO-PTR
        ZALS    LG-COUNTER
        SUB     ONE
        SACL    LG-COUNTER, 0
        BNEZ   LG-LOOP

        ZALS    NUM-ITERATIONS
        SACL    LG-COUNTER, 0
        LARK   ARO, #ADDR-E-END

TERLP   LARK

        BALS   *+, ARI, 0
        SACL   *+, ARO, 0
        ZALS   LG-COUNTER
        SLB    ONE
        SACL   LG-COUNTER, 0
        BNEZ   TFR-LP

```

```

ZALS      NUM-ITERATIONS
SUB        ONE,0
SACL      NUM-ITERATIONS,0
LAR        ARO,E-ARRAY-START
MAR        *+,ARO
SAR        ARO,E-ARRAY-START
B          LG-REC

```

* APPENDIX V

* PREDITIVE QUANTIZER LOOP VER.1

*

* Assumes voice data as well as the pitch predictor state space
 * is stored in external RAM. The pitch predictor state space is
 * kept in a circular buffer whitch is accessed using TBLR and
 * TBLW

```

INIT      ZALS      N,0
          SACL      COUNTER,0      ; initialize loop counter
          ZALS      #S-ADDR,0      ; initialize pointer to
          SACL      S-PTTR          ; incoming voice

LOOP-1    LARK      ARO,#A-ADDR    ; ARO points to preditor
          ;          ; coefficients
          LARK      AR1,#E1-ADDR   ; AR1 points to spectral
          ZAC       ; filter state space

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    LT      *+,AR1      ;Compute spectral predictor
    MPY     *+,ARO
LOOP-2    LTD      *+,AR1
    MPY     *+,ARO
    BANZ    LOOP-2
    APAC
    SACH    Y,0        ;Y contains spectral predictor
    ZAC     ;Compute pitch prediction
    LT      ALPHA
    MPY     RT
    APAC
    SACH    X,0        ;X contains pitch prediction
    ZALS    S-PTR,0
    TBLR    S
    ADD     ONE,0
    SACL    S-PTR,0
    ZALS    S          ;Subtract two predictions from
    SUB     X,0        ; input voice
    SUB     Y,0
    BGEZ    DIFF-POS   ;Quantize residual, if neg.
    ZAC     ; transmit zero
    SACL    D,0        ;Scale variance of quantized
    LT      0          ; residual
    MPYK    MINUS-1
    APAC
    B       UPDATE
DIFF-POS  LACK    ONE      ;If residual is positive,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SACL      D,0          ;transmit one
ZALS      Q,0

UPDATE   ADD      Y,0          ;Compute spectral prediction
SACH      ; filter state variable
ADD      X,0
SACH      R,0          ;Compute pitch predictor state
; variable and store it
ZALS      R-OUT-PTR,0 ;Compute pointer into circular
ADD      ONE,0        ; buffer for storing data
SACL      R-OUT-PTR,0
SUB      R-BOTTOM,0
BLZ      OUTPUT-R
LACK      #ADDR-RBUF   ;If at end of buffer,point
SACL      ; back to beginning
OUTPUT-R ZALS      R-OUT-PTR,0
TBLW     R

ZALS      R-IN-PTR,0   ;Compute pointer into circular
ADD      ONE,0        ; buffer for retrieving data
SACL      R-IN-PTR,0
SUB      R-BOTTOM,0
BLZ      INPUT-R
LACK      #ADDR-RBUF
SACL      R-IN-PTR,0

INPUT-R  ZALS      R-IN-PTR,0
TBLR     RT

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ZALS    COUNTER,0
SUB     ONE
SACH    COUNTER,0
BNEZ   LOOP-1

```

* APPENDIX IX

* RECEIVER LOOP

*

* Assumes residual input is packed in 16-bit words in external
 * RAM. It is accessed using TBLR. Synthesized is stored in a
 * circular buffer in external RAM. This buffer is accessed via
 * the external I/O ports.

```

INIT    LACK    #N          ;Initialize a loop counter
        SACL    COUNTER,0
        LACK    #D-IN-ADDR  ;Initialize pointer to input
        SACL    D-IN-PTR,0  ; residual
        ZALS    FOUR
        SACL    ACCESS-CTR-WD,0;Initialize external memory
        CALL    EXT-MEM-INIT ; interface for read/write mode
        ZAC
        SACL    BYTE-COUNTER ; to serial conversion of input
        ; residual

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LOOP-1  ZALS    BYTE-COUNTER ;Obtain quantized input residual
        BNEZ    SHIFT      ; from 16-bit residual word
        ZALS    D-IN-PTR
        TBLR    RESIDUAL-BYTE,0
        ADD     ONE,0
        SACL    D-IN-PTR
        LACK    #16
        SACL    BYTE-COUNTER,0

SHIFT   ZALS    RESIDUAL-BYTE,1;Current residual value is
        SACL    RESIDUAL-BYTE,0; high order bit of
        SACH    P,0      ; residual-byte
        ZALS    BYTE-COUNTER,0
        SUB     ONE,0
        SACL    BYTE-COUNTER,0

        LT      0      ;scale variance of residual
        MPY     D
        ZAC
        APAC
        ADD     Y,0      ;Add in spectral prediction
        SACH    E,0      ; put result on spectral pred.
        ADD     X,0      ; filler state space
        SACH    S-HAT,0  ;Store reconstructed voice in
        OUT     S-HAT, RAM-PORT-1;circular buffer
        LARK    ARO,#A-ADDR ;Compute next spectral prediction
        LARK    ARI, #E-ADDR; value
        LT      *-,ARI
        MPY     *-,ARO

```

เอกสารนี้เป็นเอกสารที่รวบรวมไว้สำหรับอาจารย์ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ZAC
 LTD #-,AR1
 MPY *-,ARO
 BNAC LOOP-2
 SACH Y,0

IN ST-HAT ; Compute next pitch predictoin
 LT ALPHA
 MPY ST-HAT
 ZAC
 APAC
 SACH X,0
 ZALS COUNTER,0
 SUB ONE,0
 SACL COUNTER,0
 BNEZ LOOP-1

* APPENDIX X

* A/D D/A SERVICE ROUTINE INVOKED BY

* INTERRUPT FROM A/D CLOCK

*

* A/D PORTION

AD IN SN,ADC ;Input voice form ADC register

ZALS SN,0 ;Pre-emphasis

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LT      OLDSN
MPYK    PRE-FAC
SPAC
SACL    TEMP,0      ;Store preemphasis voicetemp.
ZALS    S-IN-PTR,0  ;Load pointer to input voice buffer
TRLW    TEMP        ;Write out preemphasized voive in
ADD     ONE,0       ;  buffer increment pointer
SACL    S-IN-PTR,0
ZALS    SN,0        ; Delay s[n]
SACL    OLDSN

```

*

* D/A Portion

*

```

ZALS    S-OUT-PTR-1,0 ;Retrieve pntr.to output voice
                                ;  buffer
TBLR    YN           ;Real in processed voice sample
ADD     ONE,0       ;Increment pointer
SACL    S-OUT-PTR-1,0 ;Re-store pointer
ZALS    YN           ;Do De-emphasis
LT      OLD SHATN
MPYK    PRC-FAC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 APAC
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SACL    OLD SHATN,0    ;Delay output voice sample
OUT     OLD-SHATN,DAC ;Output voice sample

```

*

* Check for end of buffer. If the end, switch speech buffers

* this is done by switching pointer

```

ZALS    S-PTR-1,0      ;Check for end of data
SUB     S-OUT-END,0
BGZ     DONE
ZALS    S-OUT-PTR-1    ;Toggle bit 8, switching form page
XOR     H,100          ;--->5 to 6
SACL    S-OUT-PTR-1,0
ZALS    S-OUT-PTR-1,0
XOR     H,100
SACL    S-AT-PTR-1,0
DONE RET ; Return form interrupt

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include <graphics.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <alloc.h>
#include <bios.h>
#include <conio.h>
#include <ctype.h>
#include <dos.h>
#include <fcntl.h>
#include <io.h>
#include <math.h>

#define p      512
#define Y      120
#define YY    455
#define shift  64
#define Uint   unsigned int
#define Uchar  unsigned char

Uint far *MemoryUint(long);
float far *MemoryFlt(long);
Uchar far *MemoryUch(long);
char      *Memory(long, int);
void      Clearbuff(char*, int);
void      LoadSound(void);
char      *GetSound(void);

Uint far *GetMag(void);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Uint far *GetDat(void);
void      OpenGraphic(void);
void      Display();
void      PlotGraph(void);
void      TranferData(void);
int       MovePage(void);
void      fft(int);
void      magnitude(void);
void      Magnitude(void);
void      sort(int,int);
void      Compare(void);
int       CheckKey(void);
int       Key(void);
void      scale(void);
void      flame1(void);
void      flame2(void);
int       music(Uint);

float     *xr,*xi;
long      N,start,end,count,MaxBuff;
Uint far *mag,*freq,*FirstMag,*SecondMag;
Uchar far *Data;

int LeftT = 60, Top= 200, RighthDown = 62+p+4, Down=460;
int leftT = 60, top= 20 , righthdown = 62+p+4, down=150;

union {
    int key;
    char ch[2];
}check;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Main program for control and display main manu. *****/

```

```

void main(void)

```

```

{

```

```

    char key;

```

```

    int i,x=10, y=5 ;

```

```

    textcolor(11);

```

```

    do

```

```

    {

```

```

        clrscr();

```

```

        gotoxy(x-5,y);   cprintf("MAIN MENU");

```

```

        gotoxy(x,y+1);   cprintf("    0 -> Exit.");

```

```

        gotoxy(x,y+2);   cprintf("    1 -> Sound record form ADC.");

```

```

        gotoxy(x,y+3);   cprintf("    2 -> PlotGraph.");

```

```

        gotoxy(x,y+4);   cprintf("    3 -> Compare magnitude.");

```

```

        gotoxy(x-5,y+5); cprintf("Select number...");

```

```

        switch(key=CheckKey())

```

```

        {

```

```

            case '1': LoadSound();

```

```

                    break;

```

```

            case '2': Data = GetSound();

```

```

                    PlotGraph();

```

```

                    free(Data);

```

```

                    break;

```

```

            case '3': Compare();

```

```

        }

```

```

    }while(key!='0');

```

```

}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Initialize for display graphic mode. *****/
void OpenGraphic(void)
{
    int gdriver=DETECT,gmode,errorcode;
    initgraph(&gdriver,&gmode,"c:\\borlandc\\bgi");
    errorcode=graphresult();
    if (errorcode != grOk)
    {
        cprintf("Graphics error : %s\n",grapherrormsg(errorcode));
        cprintf("Press any key to halt : ");
        getch();
        exit(1);
    }
}

/***** Set data before calculate FFT. *****/
void TranferData(void)
{
    long i,n;
    n = end - start;
    for(i=0;i<=n;i++)
    {
        *(xr+i) = *(Data+start+i);
        *(xi+i) = 0.0;
    }
    for(i=n+1;i<N;i++)
    {
        *(xr+i) = 0.0;
        *(xi+i) = 0.0;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

/***** Plotting graphics. *****/
void PlotGraph(void)
{
    char *box,*fname,numberR[10],numberL[10],FileName[128];
    unsigned int color,i,pass;
    int j,mul,v,volume;
    long y,numstart;
    unsigned long m;
    FILE *file;

    xr = MemoryFlt(N);
    xi = MemoryFlt(N);
    mag = MemoryUint(N);
    OpenGraphic();
    flame1();
    flame2();
    count = 0;
    MovePage();
    for(;check.ch[0] != 27 && check.ch[0]!='q' && check.ch[0]!='Q'
    {
        numstart=count-p;
        if(numstart<0) numstart = 0;
        sprintf(numberR,"%u",count);
        sprintf(numberL,"%u",numstart);
        setcolor(LIGHTGREEN);
        outtextxy(rightright-40,Y-10,numberR);
        outtextxy(leftT+10,Y-10,numberL);

```

```

CheckKey();
if(Key())
{
    flame1();
    flame2();
    MovePage();
    scale();
    setcolor(GREEN);
    moveto(shift,Y);
}
}
if(check.ch[0]==27)
{
    closegraph();
    free(xr);
    free(xi);
    farfree(mag);
    return;
}
// Tranfer data for calculate fft.
TranferData();
outtextxy(shift,down+10,"FFT solution.");
fft(1);
outtextxy(shift,down+25,"Magnitude solution.");
// Solution magnitude of fft.
count = 0;
magnitude();
Magnitude();
for(;check.ch[0] != 27;)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    sprintf(numberR,"%u",count);
    setcolor(LIGHTGREEN);
    outtextxy(RigthDown-40,YY-10,numberR);
    CheckKey();
    if(Key())
    {
        flame2();
        Magnitude();
        scale();
    }
}
closegraph();
farfree(xr);
farfree(xi);
freq = MemoryUint(N);
// Sortting magnitude and saving 20 maximum level.
volume = 200;
sort(volume,1);
printf(" TOP %d MAXIMUM MAGNITUDE.\n",volume);
printf("\tMag\t\tFreq\n\n");
for(i=0,pass=start;i<volume;i++,pass+=2)
{
    printf("%2d\t %u\t\t %u\n",i+1,mag[pass],freq[i]);
    if(!(i%20)&&i!=j) getch();
}
printf("\nEnter file name to save : ");scanf("%s",fname);
strcpy(FileName,"c:\\data\\");
strcat(FileName,fname);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if((file= fopen(FileName,"wb"))==NULL)
{
    printf("Open file error.\n");
    exit(0);
}

for(i=0;i<volume;i++) fprintf(file,"%u ",freq[i]);
fclose(file);
free(freq);
free(mag);
}

/***** Display new data of time domain. *****/
MovePage(void)
{
    unsigned int i;
    long y;

    moveto(shift,Y);
    for(i=0;i<p && MaxBuff;i++,count++)
    {
        MaxBuff--;
        y = (long)Data[count]-100;
        (y<0)? (y=Y):(y=Y-y);
        if(y <= top ) y = top+3;
        setcolor(GREEN);
        lineto(i+shift,y);
    }

    return i;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 3
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        getch();
    }

    /******* Load data form card ADC. *****/
    void LoadSound(void)
    {
        char    *fname,DataPath[128],Time;
        char    *buf,*b;
        long    size;
        int     i;
        FILE    *file;

        MaxBuff = 4096;
        buf     = Memory(MaxBuff,1);
        b       = buf;
        printf("\nEnter file name to save : ");scanf("%s",fname);
        strcpy(DataPath,"c:\\data\\");
        strcat(DataPath,fname);
        if((file = fopen(DataPath,"wb")) == NULL)
        {
            printf("\nOpen file error");
            exit(0);
        }
        textcolor(10);
        for(i=3;i>=0;i--)
        {
            delay(1000);
            gotoxy(10,12);
            cprintf("\nDELAY TIME 3 SECOND BEFORE RECCORD : [%d]\n",i);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

wr = cos(M_PI/lel);
wi = sign*sin(M_PI/lel);
for (j=0; j<lel; j++)
{
    for (i=j; i<N; i=i+le)
    {
        id=i+lel;

        tr = *(xr+id)*ur - *(xi+id)*ui;
        ti = *(xr+id)*ui + *(xi+id)*ur;
        *(xr+id) = *(xr+i) - tr;
        *(xi+id) = *(xi+i) - ti;
        *(xr+i) = *(xr+i) + tr;
        *(xi+i) = *(xi+i) + ti;
    }
    tr = ur*wr - ui*wi;
    ti = ur*wi + ui*wr;
    ur = tr; ui = ti;
}
}
if (FFT!=1)
for (i=0; i<N; i++)
{
    *(xr+i) /= N;
    *(xi+i) /= N;
}
}

```

/** Calculate magnitude from fft function. */

```
void magnitude(void)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i;
double fraction, integer;

for(i=0;i<N;i++)
{
    fraction = sqrt((xr[i]*xr[i])+(xi[i]*xi[i]));
    fraction = modf(fraction,&integer);
    mag[i] = (unsigned long)(integer*20);
    if(fraction>0.5) mag[i]++;
}
}

/***** Plot magnitude in window 2. *****/
void Magnitude(void)
{
    int v,i,color;

    color = getcolor();
    setcolor(LIGHTRED);
    for(i=0;i<p && MaxBuff;i++,count++)
    {
        moveto(i+shift,YY);
        v = YY - (mag[count]/250);
        if(v <= Top || v <= 0) v = Top+3;
        lneto(i+shift,v);
    }
    setcolor(color);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*****
/*      Sort data of magnitude from maximum to minimum      */
/*      and Transfer frequency.*/
/*****
void sort(int volume,int format)
{
    long          i,j;
    unsigned int  Temp;
    // Auto mark end & start freq. for sort
    start = 200; end = 3896;
    for(j=0;j<(end-start);j++) *(freq+j) = j+start;
    for(j=0;j<(volume*2);j++)
    {
        for(i=(j+1);i<(end-start);i++)
        { // Max to min Ranking.
            if(format)
            {
                if(mag[j+start] < mag[i+start])
                {
                    Temp = mag[j+start];
                    mag[j+start] = mag[i+start];
                    mag[i+start] = Temp;

                    Temp = freq[j];
                    freq[j] = freq[i];
                    freq[i] = Temp;
                }
            }
        }
    }
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Compare data frequency of 2 files. *****/
void Compare(void)
{
    unsigned int i,j,k,frequency;

    clrscr();
    printf("FIRST FILE.");
    FirstMag = GetMag();
    printf("\nSECONDE FILE.");
    SecondMag = GetMag();
    printf("Volume of compare : ");
    scanf("%d",&j);
    clrscr();
    /* printf("NUMBER   Freq 1st       Freq 2nd\n");
    for(i=0;i<j;i++)
    {
        printf("%2d\t %u\t\t %u\n",i+1,FirstMag[i],SecondMag[i]);
        if(!(i%20)&&i!=j) getch();
    }
    */

    frequency = 0;
    for(i=0;i<j;i++)
        for(k=0;k<j;k++)
            if(FirstMag[i] == SecondMag[k]) frequency++;

    frequency = (frequency*100)/j;
    printf("\n Sound Compate : %d%%",frequency);
    farfree(FirstMag);
    farfree(SecondMag);
    music(frequency);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/***** Calculate FFT. *****/
void fft(int FFT)
{
    int log2n,n2,n1,i,j,k,l,le,lel,id;
    int sign = -1;
    double ur,ui,wr,wr,ti;

    if (FFT==1) sign = 1;
    log2n = log(N)/log(2);
    n2 = N/2;
    ((end-start)<=N/2)? (n1 = N/2-1):(n1 = N-1);
    for (i=j=0;i<n1;i++)
    {
        if(i<j)
        {
            tr = *(xr+j); ti = *(xi+j);
            *(xr+j) = *(xr+i);
            *(xi+j) = *(xi+i);
            *(xr+i) = tr, *(xi+i) = ti;
        }
        k = n2;
        while (k<=j) { j -= k; k /= 2 ; }
        j += k;
    }
    for (k=1 ; k<=log2n; k++)
    {
        le = pow(2.,k);
        lel = le/2;
        ur = 1.; ui = 0.;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for(size=0;size<30000;size++);
textcolor(11);
outportb(0x300,0);
for(size=0;size<5000000;size++);
for(size=0;size<MaxBuff;size++)
{ // Out pluse start to ADC (frequency of sample signal)
    *buf++ = inportb(0x300);
    for(Time=150;Time;Time--);
}
*buf = '\0';
fwrite(b,sizeof(char),size,file);
fclose(file);
free(b);
}

/***** Save data sound to disk. *****/
char *GetSound(void)
{
    char    fname[15],DataPath[25];
    char    *ch;
    int     handle;
    long    size;
    FILE    *file;

    printf("\nEnter file name to load : ");
    scanf("%s",fname);
    strcpy(DataPath,"c:\\data\\");
    strcat(DataPath,fname);

```

๑ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 if((handle = open(DataPath,O_RDONLY)) == -1)
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    printf("\nOpen file error");
    exit(0);
}
MaxBuff = N = filelength(handle);
ch      = Memory(MaxBuff,1);
read(handle,ch,MaxBuff);
close(handle);
return ch;
}

/***** Read data frequency for compare *****/
Uint far *GetMag(void)
{
    char      fname[15],DataPath[25];
    Uint far  *buff;
    long      size,i=0;
    FILE      *file;

    printf("\nEnter file name to load : ");
    scanf("%s",fname);
    strcpy(DataPath,"c:\\data\\");
    strcat(DataPath,fname);
    if((file = fopen(DataPath,"rb")) == NULL)
    {
        printf("\nOpen file error");
        exit(0);
    }

    N = filelength(file);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 MaxBuff = N/sizeof(Uint);
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

buff    = MemoryUint(MaxBuff);
while(!feof(file))
    fscanf(file,"%u",&buff[i++]);
fclose(file);
return buff;
}

/***** Save data frequency to disk. *****/
Uint far *GetDat(void)
{
    char    fname[15],DataPath[25];
    Uint far *buf;
    int     handle,i;
    long    size;
    FILE    *file;

    printf("\nEnter file name to load : ");
    scanf("%s",fname);
    strcpy(DataPath,"c:\\data\\");
    strcat(DataPath,fname);
    if((handle = open(DataPath,O_RDONLY)) == -1)
    {
        printf("\nOpen file error");
        exit(0);
    }
    MaxBuff = N = filelength(handle);
    if((buf = (unsigned int *)malloc(sizeof(unsigned int)
        *(MaxBuff/sizeof(unsigned int))))==NULL)
    {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        printf("Error Mem.\n");
        exit(0);
    }
    //read(handle,buf,MaxBuff);
    i=0;
    while(!eof(handle))
        read(handle,buf+i++,sizeof(unsigned int));
    close(handle);
    return buf;
}

/***** Memory allocate for data frequency. *****/
Uint far *MemoryUint(long Size)
{
    Uint far *buff;

    if(( buff = (Uint far*)farmalloc(Size*sizeof(Uint)) )== NULL)
    {
        printf("Memory alloc error.");
        exit(0);
    }

    return buff;
}

float far *MemoryFlt(long Size)
{
    float far *buff;

```

```

{
    printf("Memory alloc error.");
    exit(0);
}
return buff;
}

```

```

char far *MemoryChar(long Size)
{
    char far *buff;

    if(( buff = (char far*)farmalloc(Size*sizeof(char)) )== NULL)
    {
        printf("Memory alloc error.");
        exit(0);
    }
    return buff;
}

```

```

char *Memory(long size,int exite)
{
    char *buff;

    if(( buff = (char *)malloc(size) )!= NULL)
    {
        gotoxy(0,23);
        printf("Memory alloc error.");
        if(exite)
        {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        exit(0);
    }
    else delay(2000);
}
return buff;
}

/***** Check Special key. *****/
int CheckKey(void)
{
    /* function 1 returns 0 until a key is pressed */
    while(!bioskey(1));
    check.key = bioskey(0);
    return check.key;
}

/***** Key for control display and calculate. *****/
int Key(void)
{
    int success=0,j,mul;

    switch(check.ch[0])
    {
        case 'S' : //Mark start point for calculate.
        case 's' : start = count-p+1; break;
        case 'E' : //Mark end point for calculate.
        case 'e' : end = count; break;
        case '\t':
        case '\b': check.ch[1] = 75;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(check.ch[1])
{
    case 77 : // Display next point.
    case 73 : // Display previous page.
    case 75 : // Display previous point.
switch(check.ch[1])
{
    case 77 : j = 1; mul = 1; break;
    case 73 : if(count>p){j = 0; mul = 2;}
              else{j = 0; mul = 1;}
              break;
    case 75 : if(count>=p)
    {
        j = -1; mul = 1;
        if(check.ch[0]=='\t') j = 50;
    }
    else{j = 0; mul = 1;}
    if(check.ch[0]=='\b') j = -50;
}
if(count>=p)
{
    count -= (j+p*mul);
    MaxBuff += (j+p*mul);
    if(count<0)
    {
        count = 0;
        MaxBuff = N;
    }
    success = 1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

    case 81 :  if(count<N) success = 1; //Display next page.
    }

    return success;
}

/***** Draw scale windows 1 & 2. *****/

void scale()
{
    int m,numscal;
    char number[4];
    setcolor(LIGHTBLUE);
    for (m = (shift+2); m<=(shift+p-2);m+=20)
    {
        line(m,115,m,125);
        line(m,YY,m,YY-5);
        numscal = (m-2-shift)/20;
        sprintf(number,"%d",numscal);
        outtextxy(m-4,127,number);
    }
}

/***** Draw window 1 for display data sound. *****/

void flame1(void)
{
    int a;

    a=getcolor();

    setcolor(LIGHTRED);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outtextxy(270,8,"TIME DOMAIN");

setcolor(LIGHTCYAN);

setfillstyle(EMPTY_FILL,0);

bar3d(leftT,top,rightright,down,0,0);

setcolor(LIGHTMAGENTA);

line(leftT+2,Y,rightright-5,Y);           // X axis 1

scale();

moveto(shift,130);

setcolor(a);
}

/***** Draw window 2 for display data frequency. *****/
void flame2(void)
{
    int a;

    a=getcolor();
    setcolor(LIGHTRED);
    outtextxy(245,185,"MAGNITUDE SPECTRUM");
    setcolor(LIGHTCYAN);
    setfillstyle(EMPTY_FILL,0);
    bar3d(LeftT,Top,Rigtright,Down,0,0);
    setcolor(LIGHTMAGENTA);
    line(LeftT+1,YY,Rigtright-5,YY);       // X axis 2
    moveto(shift,365);
    setcolor(a);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int music(Uint frequency)
{
    int j,k=150;

    if(frequency>=20)
    {
        for(j=100;j<3000;j+=250)
        {
            k = k-10;
            sound(j);
            delay(k);
            nosound();
        }
    }
    else if(frequency<10)
    {
        sound(400);
        delay(400);
        nosound();
        sound(300);
        delay(650);
        nosound();
    }

    return 0;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิติกรรมประกาศ

คณะผู้จัดทำโครงการนี้ขอขอบคุณอาจารย์ ทรงชัย วิระทวีมาศผู้เป็นอาจารย์
ที่ปรึกษา โครงการนี้ และอาจารย์ท่านอื่น ที่ได้ให้ความกรุณาสละเวลาให้คำปรึกษา
ชี้แนะ และขอขอบคุณเพื่อน ๆ ทุกท่านที่ช่วยสนับสนุน ชี้แจงแก้ไข ทำให้การทำโครง
งานนี้ลุล่วงไปได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ทวิ ประทุมทาน. "การตรวจรู้เสียงพูดภาษาไทย โดยใช้หน่วยพยางค์" .
วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมคอมพิวเตอร์ บัณฑิตวิทยาลัย
จุฬาลงกรณ์มหาวิทยาลัย, 2530

จุฬา รุ่งเรืองชยพร, ประวัติ สุทธิจารุวัฒน์. "การอินเตอร์เฟสดีจิตอลซิกแนลโปรเซส
เซอร์กับไมโครคอมพิวเตอร์", วิทยานิพนธ์ปริญญาวิศวกรรมศาสตร์บัณฑิต ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร
ลาดกระบัง, 2529

กฤษดา เรยเสส. "เครื่องสังเคราะห์เสียงพูดด้วยวิธีเข้ารหัสแบบบริเนียนรีดิกทีฟ"
วิทยานิพนธ์ปริญญามหาบัณฑิต ภาควิชาวิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย จุฬาลง
กรณ์มหาวิทยาลัย, 2530

ณรงค์ ย่างสกุล, ตาลัด โชคสุขขลาลักษณ์. คอมพิวเตอร์จำเสียงพูดได้อย่างไร
วารสารเคมีคอนดัคเตอร์. ฉบับที่ 73., 2529

ด.ร. บรรจบ ชคสมบัติ. การสร้างระบบการจำเสียงพูดให้คอมพิวเตอร์
วารสารอิเล็กทรอนิกส์เวิลด์. ฉบับที่ 97., 2528

Makhoul, J., "Linear Prediction: A Tutorial Review", Pro. IEEE,
vol. 63, pp. 561-580, Apr. 1975.

E. Oran Brigham. "The Fast fourier Transform and its
Applications", Prentice-Hall, International Editions, 1988

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้