

ปีการศึกษา 2535

การจดจำเสียงพูด
(Speech Recognition)

โดย

นาย ธงชัย	วิโนทัย	รหัสประจำตัว	32.1108
นาย อีร์ศักดิ์	แซ่หยาง	รหัสประจำตัว	32.1127
นาย อมรินทร์	รัตนะรัต	รหัสประจำตัว	32.1418

อาจารย์ที่ปรึกษา

รศ.ดร.พฤกษ์ศักดิ์ ชิวสุวิทย์



ปริญญานิพนธ์ปีการศึกษา 2535

ประจำภาคการศึกษาที่ 2

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะ วิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าฯ เจ้าคุณทหาร ลาดกระบัง

เรื่อง การจดจำเสียงพูด (Speech Recognition)

ผู้จัดทำ

- | | | | |
|-------------------|----------|--------------|---------|
| 1) นาย ธงชัย | วิโนทัย | รหัสประจำตัว | 32.1108 |
| 2) นาย อีร์ศักดิ์ | แซ่หย่าง | รหัสประจำตัว | 32.1127 |
| 3) นาย อมรินทร์ | รัตนะรัต | รหัสประจำตัว | 32.1418 |

อาจารย์ที่ปรึกษา

(รศ.ดร. พุศัคดี ชิวสุวิทย์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีโอกาสไปใช้

032517



การจดจำเสียงพูด
(Speech Recognition)

ผู้จัดทำ

- 1) นาย ธงชัย วิโนทัย
- 2) นาย อธิศักดิ์ แซ่หย่าง
- 3) นาย อมรินทร์ รัตนะรัต

อาจารย์ที่ปรึกษา

รศ. ดร. พุศักรดี ชิวสุวิทย์

ปีการศึกษา 2535

บทคัดย่อ

ปฏิญานินพนธ์ฉบับนี้ จะกล่าวถึง ทฤษฎีของการจดจำเสียงพูดโดยย่อ และส่วนของฮาร์ดแวร์ที่ใช้กับระบบจดจำเสียงพูด ทฤษฎีการรู้ม การเก็บข้อมูลเสียงพูดลงบนเครื่องไมโครคอมพิวเตอร์ เพื่อเก็บลงบนหน่วยความจำ และ นำข้อมูลที่เก็บไว้ไปทำการวิเคราะห์โดยใช้เทคนิคการแปลงฟาสต์ฟูเรียร์ (FFT) เพื่อเปลี่ยนอินพุตเสียงในโดเมนเวลา ให้เป็นสเปกตรัมในโดเมนความถี่ และ หาคุณลักษณะของเสียงพูดโดยการตรวจจับความถี่จุดลุ่มที่มีขนาดสูงสุดในโดเมนความถี่ และ แสดงผลเสียที่เกิดจากการรบกวนของลอนข้างเมื่อใช้ฟังก์ชันถ่วงน้ำหนักแบบสี่เหลี่ยม และ แบบอื่น ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SPEECH RECOGNITION

By

1. Mr. Winothai Dhongchai
2. Mr. Sae-Hyang Dhirasak
3. Mr. Ratanarat Amarin

Adviser

Dr. Cheevasuvit Fusak

SECOND SYMMETRY of 1992

ABSTRACT

This thesis describes about the theory of the Speech Recognition, the hardware and software part of the Speech Recognition System, the sampling theory, the storage of data which is store in memory and be analysed by using "Fast Fourior Transform" technique to change the input speech signal in time domain to spectrum in frequency domain and find the feature of speech signal by detecting the maximum peak of all frequency sampled point. And show the leakage affective from sidelobe when using rectangular data weighting function and other.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ

abstract

บทนำ

จุดประสงค์และขอบเขต

บทที่ ๑ ทฤษฎีการสุม และการเข้ารหัสแบบฟิลส์โค้ดมอดูเลขชั้น

บทที่ ๒ การแปลงฟูเรียร์

บทที่ ๓ การแปลงฟาสต์ฟูเรียร์และการประยุกต์ใช้งาน

บทที่ ๔ ลักษณะของเสียงพูด

บทที่ ๕ การวิเคราะห์เสียงพูด

บทที่ ๖ การจดจำเสียงพูด

บทที่ ๗ ฮาร์ดแวร์

บทที่ ๘ สรุปผลการทดลอง

ภาคผนวก ก.

วงจรถ่ายงาน

ภาคผนวก ข.

โปรแกรมควบคุมการทำงาน

กิตติกรรมประกาศ

หนังสืออ้างอิง

๑

๒

๓

๑๐

๑๕

๔๕

๔๙

๕๗

๖๒

๗๗

๕

บทนำ

ในปัจจุบัน เทคโนโลยีต่างได้มีการพัฒนาไปอย่างรวดเร็ว โดยเฉพาะการควบคุมการทำงานต่าง ๆ โดยใช้ computer แต่ เนื่องจากการป้อนข้อมูล หรือ คำสั่งโดยผ่านทาง keyboard นั้นช้า จึงต้องมีวิธีการที่สามารถสั่งงาน computer ได้เร็วและสะดวกยิ่งขึ้น การใช้เสียงพูดสั่งงานโดยตรงก็เป็นวิธีการหนึ่ง

ในการที่จะสามารถใช้เสียงสั่งงาน computer ได้นั้น อย่างแรก computer จะต้องจดจำเสียงพูดนั้น ๆ ให้ได้เสียก่อนว่า คำ ๆ นั้นเป็นคำสั่งใด Project นี้จะเป็นการศึกษาเกี่ยวกับการจดจำเสียงพูดของ computer การวิเคราะห์และประมวลเสียงพูด โดยใช้อุปกรณ์ hardware ต่าง ๆ ประกอบกับการใช้โปรแกรมคอมพิวเตอร์

เนื้อหาภายในวิทยานิพนธ์นี้ กล่าวถึงเรื่องต่าง ๆ ที่เกี่ยวข้อง เช่น ทฤษฎีการสุ่มตัวอย่าง และ การเข้ารหัสของเสียง , วิธีการวิเคราะห์โดยใช้การแปลงฟาสต์ฟูเรียร์ และ อัลกอริทึมฟาสต์ฟูเรียร์ , รูปแบบการจดจำ และ วิธีการจดจำ เป็นต้น

จุดประสงค์ของปัญญานิพนธ์ :

เพื่อศึกษาระบบการจดจำเสียงพูดแบบแยกคำและไม่ขึ้นกับผู้พูด ให้สามารถแยกแยะเสียงพูดได้อย่างชัดเจน เมื่อนำไปประยุกต์ใช้ในทางอุตสาหกรรมต่อไปในภายภาคหน้า

ขอบเขตของปัญญานิพนธ์ :

ภาคเรียนที่ 1

1. เพื่อวิเคราะห์และตรวจสอบระบบ hardware ที่ใช้กับระบบจดจำเสียงพูด
2. เพื่อศึกษาแนวทางและความเป็นไปได้ ตลอดจนวิธีการในการพัฒนาระบบจดจำเสียงพูดนี้ให้มีประสิทธิภาพและสามารถนำไปใช้งานได้เป็นอย่างดี

ภาคเรียนที่ 2

1. ศึกษาถึงวิธีการในการหาพารามิเตอร์ที่จะใช้ในการจดจำเสียงพูด และศึกษาโปรแกรมทางคอมพิวเตอร์ที่เกี่ยวข้อง เพื่อนำมาประยุกต์ใช้ต่อไป
2. ค้นคว้าหาแนวทางในการจดจำเสียงพูด

บทที่ 1

ทฤษฎีการสุ่ม

และ

การเข้ารหัสข้อมูลแบบพัลส์โค้ดมอดูเลชัน

1.1 ทฤษฎีอัตราการสุ่ม

ในระบบที่ต่อเนื่อง input output และตัวแปรอื่นๆในระบบทั้งหมด เราสามารถที่จะวัดได้เมื่อเวลาใดๆ และสามารถที่จะเปลี่ยนแปลงไปได้อย่างต่อเนื่องได้ตามเวลาในระบบที่ไม่ต่อเนื่อง หรือทำงานเป็นช่วงที่แน่นอน เมื่อเราหาค่าตัวแปรใดๆในช่วงเวลาช่วงหนึ่ง ๆ ก็จะได้ค่าตัวแปรตัวหนึ่งออกมา ซึ่งค่าที่ได้มานี้ อาจจะเป็นค่าที่ใกล้เคียงกับค่าจริงๆ ในระบบที่ต่อเนื่องนั้น เพราะมันจะมีค่าเท่ากับในระบบต่อเนื่อง เฉพาะในขณะที่ทำการแปลงข้อมูลเป็น digital เท่านั้นและจะเป็นจริงก็ต่อเมื่อ มีข้อมูลจำนวนบิตเท่ากับอนันต์เท่านั้น ดังนั้นจึงเป็นการยากลำบากพอสมควรที่จะนำเอา computer ไปทำงานกับระบบที่มีการเปลี่ยนแปลงอย่างต่อเนื่องตามเวลา เพราะว่าการทำงานของ computer เป็นการทำงานที่ไม่ต่อเนื่อง จึงมีความจำเป็นที่สัญญาณของระบบต่อเนื่องเหล่านั้น จะต้องถูกทำการสุ่มตามช่วงเวลาที่ไม่ต่อเนื่อง ราวกับว่า มันเองก็ทำงานแบบไม่ต่อเนื่องเช่นกัน เราจะเรียกการทำงานของระบบที่ทำการสุ่มข้อมูล (Sample data system) ซึ่งสัญญาณที่ต่อเนื่องหรือสัญญาณ analog ใด ๆ ซึ่งถูกสุ่ม และ เปลี่ยนให้เป็นข้อมูลในเชิงตัวเลขว่า การดิจิไทซ์ จากที่ได้กล่าวมาทั้งหมดนี้ เรารู้จักกันในรูปแบบของการแปลงสัญญาณ analog เป็น digital ส่วนในกระบวนการที่ทำงานตรงข้ามกันกับกระบวนการดังกล่าวก็คือ การแปลงสัญญาณ digital เป็น analog นั่นเอง

สัญญาณที่ถูกนำเข้ามาผ่านระบบทำการสุ่ม บางทีก็มีความหมายเป็นลำดับ (sequence) แล้วแต่ว่าจะเป็นลำดับเข้าหรือลำดับออก โดยธรรมชาติแล้ว สัญญาณใด ๆ นั้นจะมีความยาวเป็นอนันต์ ดังนั้น เราจึงใช้เวกเตอร์ เป็นตัวที่ใช้แยกข้อมูลให้มีความยาวแน่นอน ซึ่งจะทำให้การวิเคราะห์นั้น สามารถทำได้ง่ายขึ้น โดยการคำนวณด้วยระบบ digital

จากทฤษฎีดังกล่าว ทำให้เราสามารถอ้างได้ว่า ข้อมูลเชิงตัวเลขที่เราเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

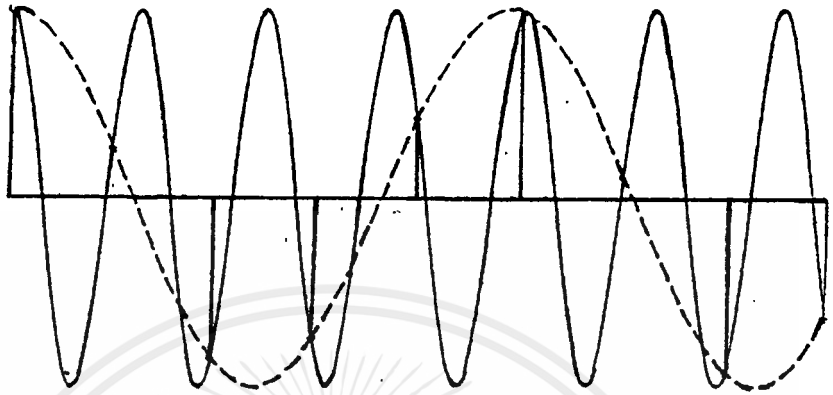
ลุ่มมา เพื่อเป็นตัวแทนสัญญาณทาง analog ต่าง ๆ นั้นย่อมมีจุดที่สูงสุดที่แน่นอน หรือ เราสามารถใช้ข้อมูลเชิงตัวเลขต่าง ๆ นั้นเป็นตัวแทนสัญญาณต่างๆทาง analog ได้เสมอ และมีจำนวนที่แน่นอน สามารถควบคุมได้ เราเรียกความแตกต่างระหว่างข้อมูลทางตัวเลขที่เราสามารถกำหนดได้นี้ว่า ข้อมูลควันไทซ์ (quntize)

การลุ่มข้อมูลแบบนี้ อาจจะมีการเกิดความคลาดเคลื่อนบ้าง เราเรียกว่า ความผิดพลาดทางควันไทซ์ ซึ่งความผิดพลาดนี้ เป็นปัญหาในเรื่องสัญญาณรบกวน จะเห็นว่า เป็นเรื่องปกติสำหรับการทำงานแบบดิจิตอล และมักก่อความไม่เที่ยงตรงในระบับที่ทำงานเกี่ยวกับสัญญาณ ดิจิตอลนี้

ในระบบที่เรา กำลังศึกษา นี้ จะใช้ช่วงเวลาในการลุ่มข้อมูลแต่ละครั้ง เท่ากันเสมอ ดังนั้น อัตราการลุ่มข้อมูลของระบบจะเท่ากับ ส่วนกลับของเวลาที่แตกต่างกัน ระหว่างการลุ่มข้อมูลแต่ละครั้ง

ในการลุ่มข้อมูลจากสัญญาณทาง analog ใด ๆ ควรจะประมาณ อัตราการลุ่มไว้ให้สูงพอสมควร เพราะถ้าใช้อัตราการลุ่มที่ค่อนข้างต่ำอาจมีปัญหาในระหว่างที่ทำการลุ่มข้อมูลจากสัญญาณต่อเนื่อง เพราะในสัญญาณนั้นอาจจะมีข้อมูลที่สำคัญมาก เช่นในสัญญาณหัวใจของคนไข้ที่มีปัญหา อาจมีบางช่วงที่มีลักษณะของความถี่สูง หากใช้อัตราการลุ่มที่มีค่าสูงกว่าสัญญาณหัวใจเพียงแค่ว่า 2-3 เท่า สัญญาณดังกล่าวก็จะไม่ได้ถูกลุ่มเข้ามา หรือระบบการลุ่มไม่สามารถทำการเก็บรายละเอียดของสัญญาณได้อย่างเพียงพอ ผลลัพธ์ที่ตามมาก็คือ เมื่อทำการแปลงข้อมูลกลับไปเป็นสัญญาณ analog แล้ว จะได้สัญญาณที่ผิดเพี้ยนจากต้นฉบับเดิมมาก

ในทฤษฎีการลุ่มนั้น การลุ่มข้อมูลจากสัญญาณที่ต่อเนื่อง กำหนดให้ใช้อัตราการลุ่มอย่างน้อย 2 เท่าของความถี่สูงสุดของสัญญาณที่นำมาทำการลุ่ม เราเรียกอัตราการลุ่มที่น้อยที่สุดที่ยอมรับได้นี้ว่า NYQUIST RATE แต่หากต้องการรายละเอียดของสัญญาณสูง ต้องเลือกอัตราการลุ่มมากกว่า 2 เท่าขึ้นไป สำหรับสัญญาณต่าง ๆ ที่จะนำมาลุ่มข้อมูลที่มีความถี่ 3-4 kHz อัตราการลุ่มที่เหมาะสมที่สุดเท่ากับ 8-9 kHz



รูปที่ 1.1 แสดงตัวอย่างของการกำหนดอัตราการสุ่มต่ำกว่ามาตรฐาน
 เส้นหนา แสดงถึง สัญญาณอินพุตที่มีความถี่ประมาณ 795 Hz
 จุด แสดงถึง การสุ่มที่มีความถี่ประมาณ 1 kHz
 เส้นประ แสดงถึง สัญญาณเอาต์พุตที่ได้ออกมา มีความถี่เท่ากับ
 ผลต่างของ อัตราการสุ่ม กับ สัญญาณอินพุต มีค่าประ
 มาณ $1000 - 795 = 205$ Hz

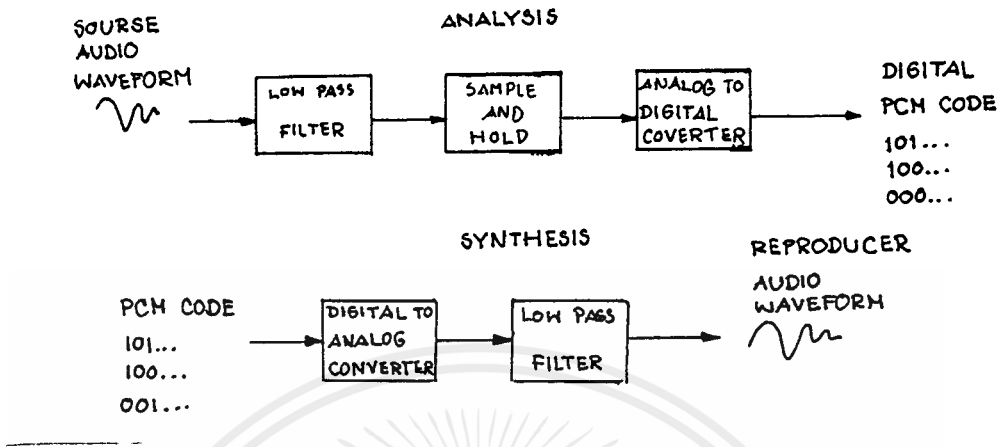
ในระบบใด ๆ ที่ใช้อัตราการสุ่มน้อยกว่า NYQUIST RATE ที่เป็นมาตรฐานนั้น เราเรียกว่า Undersampling สำหรับสัญญาณที่ถูกสุ่มแบบ Undersampling นั้น จะได้ผลลัพธ์ที่มีความถี่เท่ากับผลต่างของ ความถี่ของ อัตราการสุ่ม กับความถี่ของสัญญาณที่ป้อนเข้ามา ซึ่งจะมีควมรผิดเพี้ยนมาก

แต่ถ้ากำหนดให้อัตราการสุ่มมีค่าสูงเกินไป ก็ไม่ใช่ผลดีนัก เพราะปริมาณข้อมูลที่ทำการสุ่มออกมาได้นั้น อาจจะมีมากเกินไปจนเกิดความจำเป็น ซึ่งทำให้สิ้นเปลืองเนื้อที่ของหน่วยความจำ ดังนั้น เราควรเลือกความถี่ของอัตราการสุ่มให้ถูกต้องตามมาตรฐาน และเมื่อใช้อัตราการสุ่มค่าหนึ่งแล้ว เราอาจจะใช้อุปกรณ์ประมวลผลสัญญาณเป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกทกรองความถี่ผ่านได้ด้วย เพื่อป้องกันข้อมูลที่สำคัญ ๆ พลาดไป โดยการออกแบบวงจรกรองความถี่ ที่มากกว่าครึ่งหนึ่งของอัตราสุ่มที่เรากำหนด แต่ในการออกแบบวงจรกรองความถี่นั้น ไม่ควรรีให้จุดที่ทำการ cut off นั้น ทำการ cut off เร็วเกินไป ควรเลือกให้การตอบสนองของวงจรกรองความถี่ผ่านนั้น ลดลงอย่างช้า ๆ จะดีกว่ามาก

1.2 การเข้ารหัสแบบพัลส์โค้ดมอดูเลชัน

พัลส์โค้ดมอดูเลชัน คือ วิธีการที่ใช้กันทั่วไปในการแทนสัญญาณ analog ให้อยู่ในรูปสัญญาณแบบ digital สำหรับจุดใด ๆ บนสัญญาณ analog ที่ต่อเนื่องกันจะถูกแทนได้เป็นข้อมูลทาง digital ค่าหนึ่งในระบบเลขฐาน 2 เมื่อไรที่ค่าหนึ่งบนสัญญาณ analog ถูกทำการวัด ค่านั้น ๆ จะถูกนำมาเปรียบเทียบว่า จะต้องแทนด้วยโค้ดอะไรในเลขฐาน 2 เรียกว่า พัลส์โค้ด ถ้าหากว่า ค่านั้นอยู่ในช่วงระหว่าง 2 ค่าในเลขฐาน 2 จะถือว่า ค่าเลขฐาน 2 ที่ใกล้เคียงมากที่สุดนั้น จะเป็นโค้ดสำหรับแทนสัญญาณ analog ที่เข้าไป วิธีการแบบนี้เรียกว่า ควันไทซ์เซชัน (Quantization) การแบ่งคลื่นออกเป็นช่วงย่อย ๆ โดยแต่ละช่วงจะถูกกำหนดให้เท่ากับโค้ดในระบบเลขฐาน 2 ค่าหนึ่ง ดังนั้น ในการวิเคราะห์สัญญาณเสียงใด ๆ เมื่อสัญญาณเสียงผ่านกระบวนการเหล่านี้ ชุดข้อมูลที่เป็น พัลส์โค้ด จะถูกส่งออกมาอย่างต่อเนื่อง ผลที่ได้คือ สัญญาณพัลส์โค้ดมอดูเลชัน แต่เพราะว่าสัญญาณเสียงที่จะนำมาวิเคราะห์และทำการบันทึกไว้นั้น จะถูกนำมาเก็บในรูปของสัญญาณ digital ดังนั้น ในการที่จะนำเอาสัญญาณเสียงนั้นๆ มาผ่านกระบวนการต่างๆ เราจึงต้องทำการปรับปรุงแต่งสัญญาณนั้นให้เหมาะสมเสียก่อน ซึ่งได้แสดงวิธีการต่างๆ ดังรูป ซึ่งแสดงถึง แผนผังการทำงานของเครื่องสังเคราะห์สัญญาณเสียง หรือ Speech Synthesizer ซึ่งเครื่องนี้ จะทำการแปลงสัญญาณเสียง ที่เราต้องการที่จะเก็บบันทึกไว้ ให้อยู่ในรูปของข้อมูลทาง digital ที่เราเรียกว่า พัลส์โค้ดมอดูเลชัน



รูปที่ 1.2 แสดง diagram ของระบบการวิเคราะห์และสังเคราะห์เสียง
โดยวิธี พัลส์โค้ดมอดูเลชัน

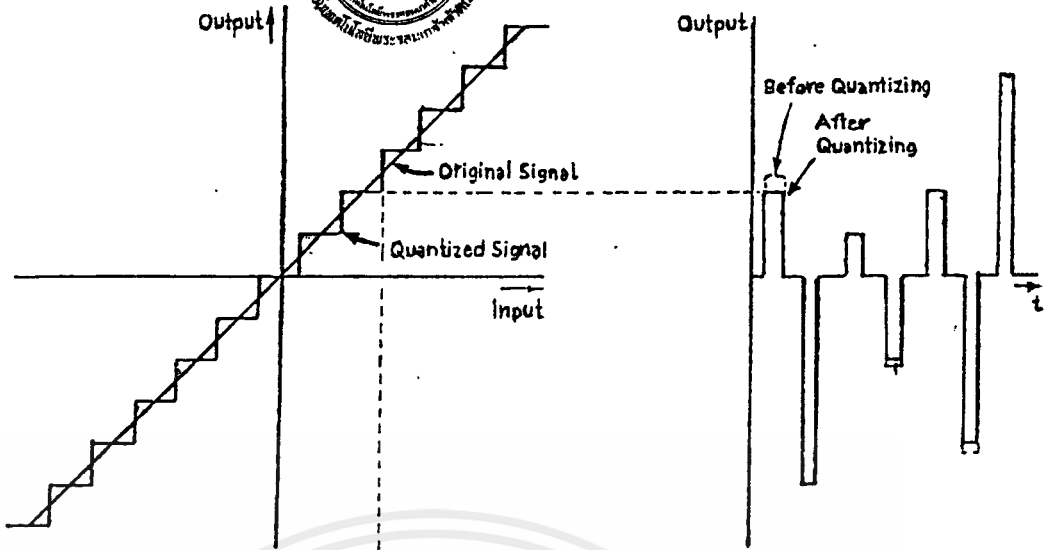
จากรูปสามารถอธิบายการทำงานได้ดังนี้ คือ สัญญาณเสียงโดยทั่วไป ซึ่งบางส่วนของสัญญาณ อาจมีความถี่สูงเกินไป ทำให้ผลต่างระหว่างระดับสัญญาณ 2 จุดใด ๆ ในขณะที่ทำการสุ่ม มีค่ามากเกินกว่า ค่าสูงสุดของ พัลส์โค้ด ที่เราได้กำหนดไว้ มักทำให้เกิดความผิดพลาดของสัญญาณ ที่เราเรียกว่า compliance error ดังนั้น เราจึงต้องนำสัญญาณที่เราต้องการจะวิเคราะห์ และ บันทึก มาผ่านวงจรความถี่ต่ำผ่าน ซึ่งจะสามารถลดปัญหานี้ได้ จากนั้นสัญญาณที่ได้จะถูกนำเข้าวงจรส่วนที่เรียกว่า วงจรสุ่มและหน่วงสัญญาณ (sample & hold circuit) เหตุที่ต้องใช้วงจรนี้ เพราะ ในวงจรนี้เราได้เลือกใช้อุปกรณ์ที่ทำการแปลงสัญญาณ analog ให้เป็นสัญญาณ digital ที่ทำงานแบบ Successive approximation ซึ่งอุปกรณ์แปลงสัญญาณนี้ จะทำการแปลงสัญญาณออกมาทีละบิต แล้วกลับไปเช็คสัญญาณอินพุทใหม่อีก แล้วทำการแปลงไล่ไปจนครบทุกบิต จึงถือว่าเสร็จการทำงาน 1 ครั้ง ถ้าหาก ไม่มีวงจรสุ่มและหน่วงสัญญาณไว้ จะทำให้สัญญาณทางอินพุทของ ADC ไม่ใช้สัญญาณเดิม ในระหว่างกระบวนการ ทำการแปลงข้อมูลเหล่านั้นกลับไปเป็นสัญญาณเสียง มักจะมีความผิดพลาดอยู่เสมอ จากนั้นข้อมูลทาง digital ที่ได้รับออกมาแต่ละไบท์จะนำมาผ่านกระบวนการทาง software โดย microprocessor เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อทำการแปลงสัญญาณออกมาในรูปของ พัลส์โค้ด แล้วนำข้อมูลไปเก็บไว้ที่หน่วยความจำ มาถึงจุดนี้ก็คือ เสร็จสิ้นขั้นตอนในการวิเคราะห์สัญญาณเสียง เพื่อเก็บบันทึกในรูปแบบของสัญญาณ digital แบบ พัลส์โค้ดมอดูเลชัน

สำหรับขั้นตอนในการสังเคราะห์สัญญาณเสียงก็คือ การนำเอาข้อมูล พัลส์โค้ด ที่มีอยู่ในหน่วยความจำ มาทำการแปลงกลับให้เป็นสัญญาณเสียง โดยนำมาผ่านกระบวนการแปลงสัญญาณเหล่านั้น กลับออกมาในรูปของข้อมูล 8 บิต โดยวิธีการทางซอฟต์แวร์ จากนั้นจึงนำเอาข้อมูลนั้น มาผ่านวงจรแปลงสัญญาณ digital ให้เป็นสัญญาณ analog แต่สัญญาณที่ออกมาจากวงจรแปลงสัญญาณดังกล่าว มักจะไม่เรียบ หรือ ไม่มีความเป็นเชิงเส้นในลักษณะสัญญาณทาง analog เท่าที่ควร จึงมีความจำเป็นที่จะต้องนำเอา วงจรกรองความถี่ต่ำมาใช้ เพื่อให้สัญญาณมีความเป็นเชิงเส้นอย่างสมบูรณ์ จึงเสร็จทุกขั้นตอน ในการทำงานของระบบการวิเคราะห์และสังเคราะห์เสียงโดยวิธี พัลส์โค้ดมอดูเลชัน

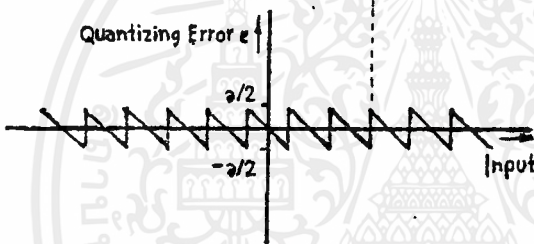
หลังจากการสุ่มข้อมูล ขนาดของ amplitude ที่ได้ จะเรียกว่า ควันไทซ์ (Quantize) ซึ่งมีขนาดแบบไม่ต่อเนื่อง สัญญาณใหม่ที่ได้นี้ จะเป็นสัญญาณแบบคร่าว ๆ ของสัญญาณเดิม ซึ่งสัญญาณทั้ง 2 ย่อมไม่เหมือนกันทุกส่วน ทำให้เกิดความผิดพลาดขึ้น ความแตกต่างระหว่าง 2 สัญญาณนี้เราเรียกว่า Quantizing error ในระบบการส่งข้อมูลแบบ พัลส์โค้ดมอดูเลชัน สัญญาณรบกวนที่เกิดจากการควันไทซ์ (Quantizing noise) เป็นสัญญาณรบกวนที่สำคัญที่สุด ต้องระวังอย่างมาก

สัญญาณรบกวนที่เกิดจากการควันไทซ์นี้ สามารถลดได้โดย การลดขนาดของช่วงการควันไทซ์ลง หรือเพิ่มระดับให้มากขึ้นไปอีก สำหรับการสื่อสารนั้น มักจะใช้การเข้ารหัสในรูปของข้อมูล 8 บิต ซึ่งมีทั้งหมด 256 ระดับ



(a) Transfer Function of a Linear Quantizer

(b) Typical PAM Signal



(c) Quantizing Noise

รูปที่ 1.3 แสดงการคว้นไตซ์สัญญาณ analog และความผิดพลาดจากการคว้นไตซ์

ถ้าใช้รหัส N บิต จะได้ช่วงการ คว้นไตซ์ทั้งหมด 2^N ระดับ ซึ่งได้ค่า

$$\text{rms S/N ratio} = 1.8 + 6N \text{ เดซิเบล}$$

ยกตัวอย่างเช่น การเข้ารหัส 8 บิต อัตราส่วนของสัญญาณกับ noise คือ 49.8 เดซิเบล สำหรับสมการนี้ ได้กำหนด amplitude สูงสุดที่ระบบจะรับได้ โดยไม่ถูกขลิบ (clip)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การแปลงฟูเรียร์

FOURIER TRANSFORM

การแปลงฟูเรียร์เป็นเครื่องมือวิเคราะห์ในงานวิทยาศาสตร์สาขาต่าง ๆ ในปัจจุบัน การประยุกต์ใช้งานที่นิยม คือ ใช้วิเคราะห์ระบบเชิงเส้นที่ไม่แปรผันตามเวลา (LINEAR TIME-INVARIANT SYSTEM) แต่การแปลงฟูเรียร์ที่จะพูดถึงต่อไปนี้เป็นหลักหรือคุณสมบัติสำคัญซึ่งสามารถใช้วิเคราะห์ในงานทั่วไป

2.1) ฟูเรียร์อินทิกรัล

ฟูเรียร์อินทิกรัล มีคำจำกัดความ เป็นสมการว่า

$$H(f) = \int_{-\infty}^{\infty} h(t)e^{j2\pi ft} dt \quad \dots\dots\dots(1)$$

$h(t)$ เป็นเทอมของฟังก์ชันช่องตัวแปรเวลา และ $H(f)$ เป็นเทอมของฟังก์ชันของตัวแปรความถี่ การแปลงฟูเรียร์ของฟังก์ชันเวลา จะแทนด้วยตัวอักษรตัวใหญ่ ในรูปทั่วไป การแปลงฟูเรียร์ มีรูปเป็นจำนวนเชิงซ้อน (COMPLEX NUMBETR)

$$H(f) = R(f) + jI(f) = |H(f)| e^{j\theta(f)} \quad \dots\dots\dots(2)$$

เมื่อ $R(f) =$ เป็นส่วนจริงของการแปลงฟูเรียร์

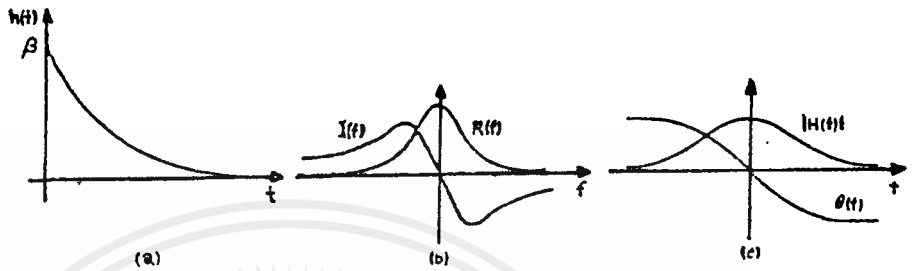
$I(f) =$ เป็นส่วนจินตภาพของการแปลงฟูเรียร์

$|H(f)| =$ เป็นแอมพลิจูด หรือ ฟูเรียร์สเปกตรัม ของ $h(t)$
ซึ่งได้จาก

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(i) = เป็นมุมเฟสของการแปลงฟูเรียร์ ซึ่งได้จาก $\tan^{-1}[I(f)/R(f)]$



รูปที่ 2.1 (a) ตัวอย่างของฟังก์ชันในโดเมนเวลา
 (b) ส่วนจริงและส่วนจินตภาพของการแปลงฟูเรียร์
 (c) แอมพลิจูดและเฟสของการแปลงฟูเรียร์

2.2) การแปลงอินเวอร์สฟูเรียร์ (INVERSE FOURIER TRANSFORM)

การแปลงอินเวอร์สฟูเรียร์ มีค่าจำกัดความดังนี้

$$h(t) = \int_{-\infty}^{\infty} H(f) e^{j2\pi ft} dt \quad \dots\dots\dots(3)$$

2.3) การแปลงฟูเรียร์แบบไม่ต่อเนื่อง

DISCRETE FOURIER TRANSFORM หรือ DFT

DFT เป็นลำดับของความถี่ไม่ต่อเนื่อง. ที่ได้จากการสุ่มในหนึ่งคาบเวลาของการแปลงฟูเรียร์

เมื่อกำหนดให้จำนวนการสุ่มเท่ากับ N สุ่มบนคาบเวลา $0 < \omega < 2\pi$
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น

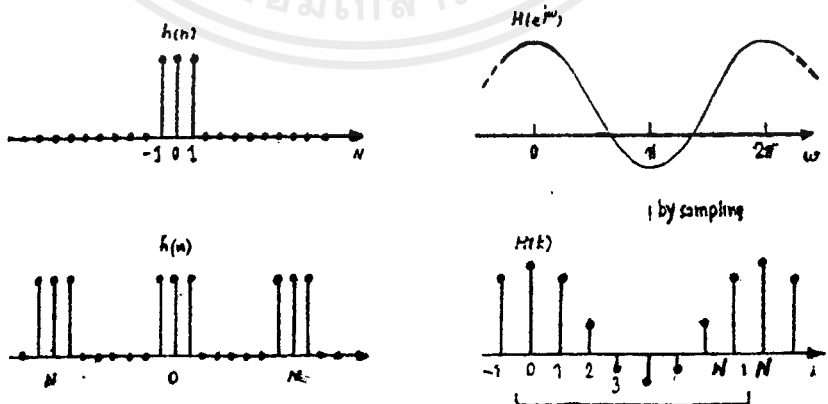
$$w_k = 2\pi k/N \quad \text{เมื่อ } 0 < k < N-1 \quad \dots\dots\dots(4)$$

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

$$H(k) = H(e^{j\omega}) \quad \text{เมื่อ } \omega = \omega_k = 2\pi k/N \quad \text{และ } 0 < k < N-1 \dots(5)$$

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

เราจะเห็นว่า $H(e^{j\omega})$ มีคาบเวลาเท่ากับ ω หรือ 2π ซึ่งได้จากการกระจายอนุกรมฟูเรียร์ และ ค่าวนค่าสัมประสิทธิ์จากลำดับ $\{ h(n) \}$ แสดงว่าเมื่อสัญญาณเวลาต่อเนื่อง ถูกสุ่มด้วยคาบเวลาการสุ่ม T_s สเปกตรัมของผลลัพธ์จากการแปลงฟูเรียร์ จะมีฟังก์ชันคาบเวลาของความถี่เท่ากับ $2\pi/T_s$ จากรูปที่ 2.1 เมื่อ $H(e^{j\omega})$ ถูกสุ่มด้วยคาบเวลาการสุ่ม $\omega_s = 2\pi/N$ เมื่อแปลงกลับเป็นลำดับของฟังก์ชันเวลาที่ไม่ต่อเนื่อง $\{ h(n) \}$ จะที่คาบเวลา เป็น $2\pi/\omega_s = N$



รูปที่ 2:2 แสดงผลลัพธ์ในลำดับของเวลา จาก Sampled spectrum

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

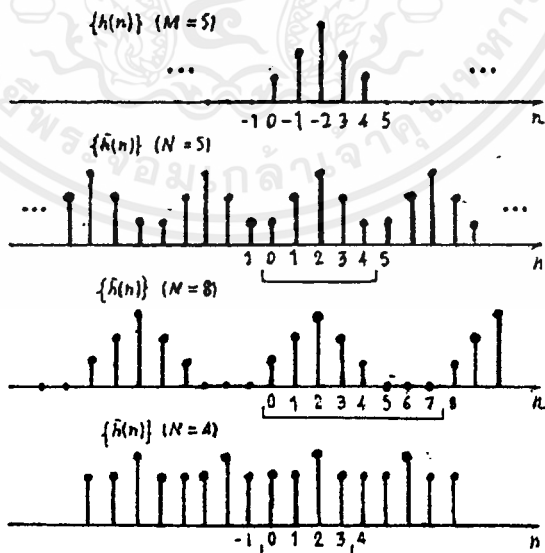
ดังนั้นลำดับ Discrete time ที่อยู่ในฟังก์ชันคาบ สามารถหาใน
เทอมของ $\{h(n)\}$ ได้

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

ลำดับของ $\{h(n)\}$ เรียกว่า periodic extension ของ $\{h(n)\}$ จำนวนสุ่มในหนึ่งคาบของ spectrum N จะมีค่าเท่ากับคาบของ $\{h(n)\}$ เราสามารถเลือกค่า N เท่าใดก็ได้ ซึ่งเป็นจำนวนการสุ่มของ $H(e^{j\omega})$ เมื่อ $0 < \omega < 2\pi$ และ N เป็นคาบของ $\{h(n)\}$ เราต้องเลือกค่าที่ไม่น้อยเกินไป ในลักษณะที่ $\{h(n)\}$ เป็นฟังก์ชันคาบเวลาแบบไม่จำกัด ทำให้เกิดลักษณะสัญญาณที่ถูกสุ่มทับกัน สัญญาณที่ได้จะผิดพลาดไป สามารถแสดงได้จากตัวอย่างข้างล่าง

$$\text{กำหนด } h(n) = \begin{cases} h(n) & \text{เมื่อ } 0 \leq n \leq M-1 \\ 0 & \text{เมื่อ } M \leq n \leq N-1 \end{cases}$$

จะได้จุดที่เกิดจากการสุ่ม เมื่อ N มีค่าต่าง ๆ ดังรูป



รูปที่ 2.3 แสดงความสัมพันธ์ของช่วง เวลาของลำดับ M และ จำนวนจุด

ในการสุ่มสเปกตรัมของ N

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปจะเห็นว่า ถ้าค่า $M > N$ รูปสัญญาณจะทับกัน (overlap)
หรือเกิด time-aliasing จึงสามารถป้องกันได้โดยเลือกค่า $N \geq M$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การแปลงฟูรีเยร์ และการประยุกต์ใช้งาน

FFT (FAST FOURIER TRANSFORM) เป็นอัลกอริทึม(algorithm) ที่ช่วยในการคำนวณ DFT ให้มีประสิทธิภาพมากขึ้น และลดจำนวนการกระทำทางคณิตศาสตร์ และ ขนาดของหน่วยความจำที่จะต้องใช้ในการคำนวณ DFT ดังนั้น อัลกอริทึมของ FFT จึงเป็นเทคนิคที่ถูกเลือกใช้ในการประยุกต์ใช้งานด้านต่าง ๆ

3.1) ลักษณะของปัญหา

ให้ $\{ X(m) \}$ เป็นลำดับของข้อมูล $X(m), m = 0, 1, \dots, N-1$ ที่ได้จากการสุ่มในช่วงจำกัดของสัญญาณ $x(t)$ เรามีอัลกอริทึมที่ใช้ในการคำนวณค่าสัมประสิทธิ์ $C_x(k)$ ดังสมการนี้

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

เมื่อ $W = e^{-j2\pi/N}$, $j = \sqrt{-1}$ เราเรียกสมการ (1) นี้ว่า DFT ของ $\{ X(m) \}$ อัลกอริทึมที่เราจะกล่าวถึงต่อไปนี้เรียกว่า การแปลงฟูรีเยร์ (FFT) เป็นอัลกอริทึมที่พัฒนาโดย COOLEY และ TUKEY บางทีอาจเรียกว่าอัลกอริทึมของ COOLEY และ TUKEY ก็ได้

ตั้งแต่นี้ไป เราจะตั้งข้อสมมติฐานว่า

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

จากที่เราได้เข้าใจกันแล้ว ถ้าไม่ต้องการให้ข้อมูลสูญหาย ต้องมีขนาด N ที่เหมาะสม ตามทฤษฎีการสุ่ม (Sampling theorem) นั่นคือ $N \geq 2BL$ เมื่อ B เป็น bandwidth ของ $x(t)$ ในหน่วย H_z และ L คือช่วง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาของมัน สำหรับในกรณีนี้ N ไม่เท่ากับ 2^n ซึ่งเราจะได้พูดกันต่อไป

3.2) เข้าสู่การหาอัลกอริทึม

เราจะพิจารณาในกรณีที่ $\{X(m)\}$ เป็นค่าจำนวนจริง และ $N = 8$ จากคุณสมบัติ คอมเพล็กซ์ คอนจูเกต (Complex conjugate) เรามี

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

ดังนั้น เมื่อเทียบกับสมการ (1) ทำให้เหลือการคำนวณเพียง

$$C_x(k) = 1/8 \sum_{m=0}^7 X(m) W^{km}, \quad k = 0, 1, \dots, 4 \dots (2)$$

เมื่อ $W = e^{-j2\pi/4}$

จากสมการ (1) จะได้ว่า

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

เมื่อ

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

และ

$$B(k) = j \sum_{m=0}^7 X(m) \sin(mk\pi/4), \quad k = 0, 1, \dots, 4 \dots (3)$$

ใช้สัญลักษณ์เมตริกซ์ สมการ (3) จะเป็น

$$A = CX$$

$$B = SX$$

เมื่อ A และ B มีมิติ (5x1) , X มีมิติ (8x1) , C และ S มีมิติ (5x8) ดังนี้

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

และ

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

เป็นสมการ (4)

เมื่อพิจารณาสมการ (4) จะเห็นว่า สมาชิกของเมตริกซ์ C และ S มีค่าซ้ำกัน ซึ่งเป็นเหตุผล 2 ประการที่พัฒนาอัลกอริทึมฟาสต์ฟูเรียร์ ดังนี้

1) ความเป็นซายน์ (Sinusoidal) จัดกลุ่มลักษณะเด่น ซึ่งจะเกิดขึ้นได้เมื่อเลือกจุดสุ่มที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ถ้า N เป็นจำนวนจุดลุ่มของ $x(t)$ ที่มีรูปร่างไม่แน่นอนแล้ว การเกิดซ้ำกันของเมตริกซ์ C และ S จะเพิ่มขึ้นตามจำนวนตัวประกอบของ $N = 2^n$, $n = 2 \times 2 \times 2 \times \dots \times 2$ ดังนั้น เมื่อจำนวนของข้อมูลมาก การใช้อัลกอริทึมฟาสต์ฟูเรียร์ จะช่วยประหยัดเวลาและหน่วยความจำเป็นอย่างมาก

3.3) แนวทางการพัฒนาอัลกอริทึม

แนวทางที่จะพัฒนาอัลกอริทึมที่ต้องการคือ การหาความสัมพันธ์กับการซ้ำกันในรูปแบบทั่วไปของเมตริกซ์ C และ S กับกำลังของ W หรือ หลักรากที่ N ของหนึ่ง (Principle N -th root of unity) เพื่อหาสัญลักษณ์ของรูปทั่วไป สัญลักษณ์ แสดงค่าในฐาน 10 แต่ละตัวของ m , $0 \leq m \leq N-1$ ให้อยู่ในรูปของเลขฐาน 2 หรือไบนารี (Binary) ดังนี้

$$m = m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_12^1 + m_02^0$$

เมื่อ $m_v = 0$ หรือ 1 , $v = 0, 1, \dots, n-1$, $n = \log_2 n$

ในทำนองเดียวกันค่าในฐาน 10 แต่ละตัวของ k , $0 \leq m \leq N-1$ เปลี่ยนรูปเป็น

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

เมื่อ $k_v = 0$ หรือ 1 , $v = 0, 1, \dots, n-1$

ในรูปแบบไบนารีของ $X(m)$ แทนด้วย $X(m)$ จะได้ว่า

$$\begin{aligned} X(m) &= X(m_{n-1}2^{n-1} + m_{n-2}2^{n-2} + \dots + m_12^1 + m_02^0) \\ &= X(m_{n-1}, m_{n-2}, \dots, m_1, m_0) \dots \dots (5) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปลี่ยนรูปสมการ (5) ตามความสัมพันธ์ที่ได้ ดังนี้

$$\begin{aligned} & \uparrow X(m) W^{km} \\ = & \uparrow \uparrow \dots \uparrow X(m_{n-1}, m_{n-2}, \dots, m_1, m_0) W^{k(m_{n-1}2^2 + m_0^2)} \\ & \dots \dots \dots (6) \end{aligned}$$

3.4) การพัฒนาอัลกอริทึม

ขั้นตอนการพัฒนาอัลกอริทึม จะอธิบายในกรณี $N = 8 = 2^3$ ดังนี้

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

เมื่อ $W = e^{-j\pi/4}$ เราจะสามารถแสดงค่าของ m ในรูปไบนารี ได้ดังนี้

$$m = m_2 2^2 + m_1 2^1 + m_0 2^0 \quad \dots \dots (8)$$

จากสมการ (6) สมการ (7) และ (8) จะได้ว่า

$$\begin{aligned} 8C_x(k) &= \uparrow \uparrow \uparrow X(m_2, m_1, m_0) W^{k(4m_2 + m_1 + m_0)} \\ &= \uparrow \uparrow \uparrow X(m_2, m_1, m_0) W^{4km_2} W^{2km_1} W^{km_0} \\ & \dots \dots \dots (9) \end{aligned}$$

จากสมการ (9) ผลบวกในสุดของ m_2 แทนด้วยสัญลักษณ์ M_2

เราจะได้ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

เพราะว่า $(-1)^{m(4k+2k+k)} = 1$ M_2 สามารถแสดงได้ในรูปของ

$$M_2 = \tau X(m_2, m_1, m_0) (-1)^{k_0 m_2} \dots\dots(10)$$

จากสมการ (10) ผลบวกบน m_2 แสดงค่าที่ รวมได้ในรูปฟังก์ชัน
ของ k_0, m_1, m_0

$$\begin{aligned} M_2 &= \tau X(m_2, m_1, m_0) (-1)^{k_0 m_2} \\ &= X_1(k_0, m_1, m_0) \dots\dots(11) \end{aligned}$$

แทนสมการ (11) ลงในสมการ (9)

$$8C_x(k) = \tau \tau X_1(k_0, m_1, m_0) W^{2k_1 m} W^{k m_0} \dots\dots(12)$$

ทำซ้ำอีกครั้ง ในส่วนในสุดของสมการ (12) และแทนด้วยสัญลักษณ์

M_1

$$\begin{aligned} M_1 &= \tau X_1(k_0, m_1, m_0) W^{2km} \\ &= \tau X_1(k_0, m_1, m_0) (-i)^{(4k_2 + 2k_1 + k_0) m_2} \\ &\dots\dots(13) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในสมการ (13) จำนวน $(-i)^{4k} 2^{m_1} = 1$ จะได้รูปแบบที่ง่ายขึ้น ดังนี้

$$M_1 = \dagger X_1 (k_0, m_1, m_0) (-i)^{(2k_1+k_0)m_1} \dots (14)$$

ผลบวกบน m_1 จะได้ผลลัพธ์ M_1 ที่เป็นฟังก์ชัน ของ k_0, k_1, m_0

$$M_1 = \dagger X_2 (k_0, k_1, m_0) \dots (15)$$

แทนสมการ (15) ลงในสมการ (12) จะได้

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

เหมือนกับ M_2 และ M_1 เราแทนสัญลักษณ์ผลบวกบน m_0 ด้วย M_0

$$M_0 = \dagger X_2 (k_0, k_1, m_0) W^{m_0 k}$$

ซึ่งจะได้

$$M_0 = \dagger X_2 (k_0, k_1, m_0) [(1-i)/\sqrt{2}]^{(4k_2+2k_1+k_0)} \dots (16)$$

การทำผลบวกบน m_0 จะได้ M_0 ที่เป็นฟังก์ชันของ k_0, k_1, k_2 ดังนี้

$$M_0 = X_3 (k_0, k_1, k_2) \dots (17)$$

$$\begin{aligned} 8C_x(k) &= 8C_x (k_0, k_1, k_2) \\ &= X_3 (k_0, k_1, k_2) \dots (18) \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นแรกในการหาคำตอบ พิจารณา

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

2) coefficient $(-1), (-i), [(1-i)/\sqrt{2}]$ มีความสัมพันธ์กับรากที่ 2, 4, 8 ตามลำดับ ของ unity $e^{-i2\pi}$

ดังนั้น จึงกำหนดสัญลักษณ์ใหม่ ดังนี้

$$A_2 = e^{-i2\pi/2}, \quad r = 1, 2, \dots, \log_2 N$$

.....(19)

เมื่อ A_2 เป็นรากที่ 2^r ของ unity โดยมีคุณสมบัติดังนี้

$$1) \quad A_2 = W^{N/2}, \quad W = e^{-i2\pi/n}$$

.....(20-1)

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

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

.....(20-2)

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

.....(20-3)

จากสมการข้างต้นสามารถเขียนในเทอมของ A_2 จะได้

$$X_1(k_0, m_1, m_2) = \dagger X(m_2, m_1, m_0) A_2^{k_0 m_2} \quad \text{.....(21)}$$

ในสมการ (21) k_0 มีค่า 0 หรือ 1 ทำให้ได้สมการสี่สมการ ในแต่ละค่า k_0

$$\text{กรณี 1 : } k_0 = 0 \quad \dots\dots\dots(22-1)$$

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

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

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

$$X_1(0,1,1) = X(0,1,1) + X(1,1,1) \implies X_1(3) = X(4) + X(7)$$

$$\text{กรณี 2 : } k_0 = 1 \quad \dots\dots\dots(22-2)$$

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

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

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

$$X_1(1,1,1) = X(0,1,1) + X(1,1,2) \implies X_1(7) = X(4) - X(7)$$

จากสมการ (15) ในทอม A_4

$$X_2(k_0, k_1, m_1) = X_1(k_0, 0, m_0) + X_1(k_0, 1, m_0) A^{(2k+k)}$$

$$\text{กรณีที่ 1 : } (k_1, k_0) = (0, 0) \quad \dots\dots\dots(23-1)$$

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

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

$$\text{กรณีที่ 2 : } (k_1, k_0) = (0, 1) \quad \dots\dots\dots(23-2)$$

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

$$\text{กรณีที่ 3 : } (k_1, k_0) = (1, 0) \quad \dots\dots\dots(23-3)$$

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

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

$$\text{กรณีที่ 4 : } (k_1, k_0) = (1, 1) \quad \dots\dots\dots(23-4)$$

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

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

สมการ (17) เขียนในเทอม A_0 จะได้

$$X_2(k_0, k_1, k_2) = X_2(k_0, k_1, 0) + X(k_0, k_1, 1) A_0^{(4k+2k+k)} \quad \dots\dots\dots(24)$$

$$\text{กรณีที่ 1 : } (k_2, k_1, k_0) = (0, 0, 0) \quad \dots\dots\dots(25-1)$$

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

$$\text{กรณีที่ 2 : } (k_2, k_1, k_0) = (0, 0, 1) \quad \dots\dots\dots(25-2)$$

$$X_3(1,0,0) = X_2(1,0,0) + A_0 X_2(1,0,1) \implies X_3(4) = X_2(4) + A_0 X_2(5)$$

$$\text{กรณีที่ 3 : } (k_2, k_1, k_0) = (0, 1, 0) \quad \dots\dots\dots(25-3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$X_3(0,1,0) = X_2(0,1,0) + A_0 X_2(0,1,1) \implies X_3(2) = X_2(2) + A_0 X_2(3)$$

กรณีนี้ 4 : $(k_2, k_1, k_0) = (0, 1, 1)$ (25-4)

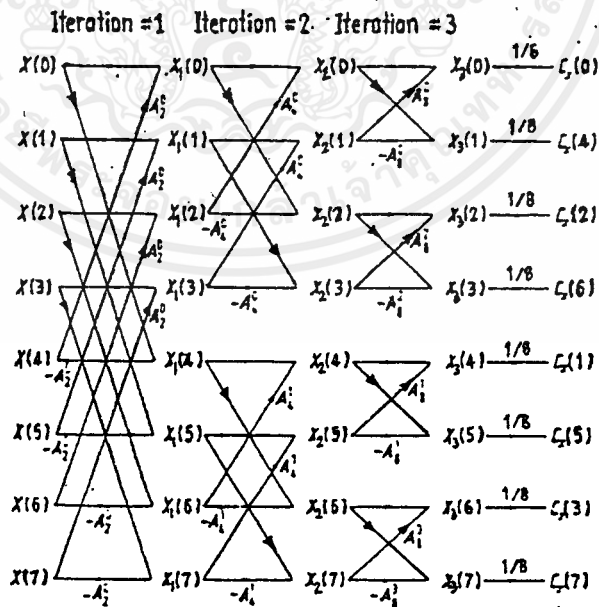
$$X_3(1,1,0) = X_2(1,1,0) + A_0 X_2(1,1,1) \implies X_3(6) = X_2(6) + A_0 X_2(7)$$

กรณีนี้ 5 : $(k_2, k_1, k_0) = (1, 0, 0)$ (25-5)

$$X_3(0,0,1) = X_2(0,0,0) + A_0 X_2(0,0,1) \implies X_3(1) = X_2(0) - X_2(1)$$

ลำดับการกระทำทางคณิตศาสตร์ ตั้งแต่สมการ (25-1) ถึง (25-5) ได้แสดงไว้ในรูปที่ 1 และการทำซ้ำ (iteration) ครั้งที่ 3 ก็คือ $r = 3$ ในสมการ (19) เพื่อให้ได้ค่า $C_x(k)$ ตามที่ต้องการ ใช้สมการ

$$8C_x(k_2, k_1, k_0) = X_3(k_0, k_1, k_2) \dots\dots\dots (26)$$



รูปที่ 3.1 แสดงแผนภาพการไหล (signal flow graph) ของสัญญาณ FFT เมื่อ $N=8$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่า มีการสลับตำแหน่งของสัมประสิทธิ์โบนารี k_0 , k_1 และ k_2 ใน $X_3()$ เมื่อเทียบกับตำแหน่งจริง $C_x(k)$ เรียกลักษณะเช่นนี้ว่าปรากฏการณ์กลับบิต (bit reversal) $X_3()$ มีความสัมพันธ์กับ $C_x(k)$ ตามสมการข้างล่างนี้

$$\begin{aligned} X_3(0) &= 8C_x(0) \\ X_3(4) &= 8C_x(1) \\ X_3(2) &= 8C_x(2) && \dots\dots\dots (27) \\ X_3(6) &= 8C_x(3) \\ X_3(1) &= 8C_x(4) \end{aligned}$$

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

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

จะได้

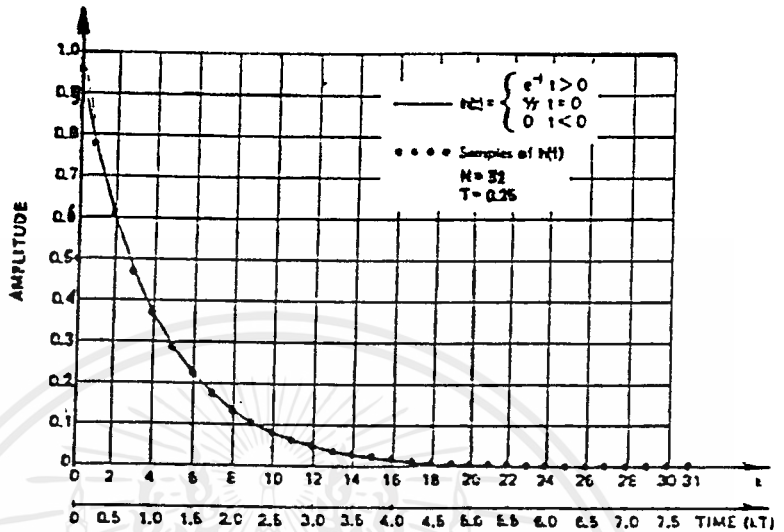
$$\begin{aligned} C_x(5) &= C_x(3) \\ C_x(6) &= C_x(2) \\ C_x(7) &= C_x(1) \end{aligned}$$

$C_x(5)$, $C_x(6)$, $C_x(7)$ สามารถหาได้จากสมการ (18) และ (24)

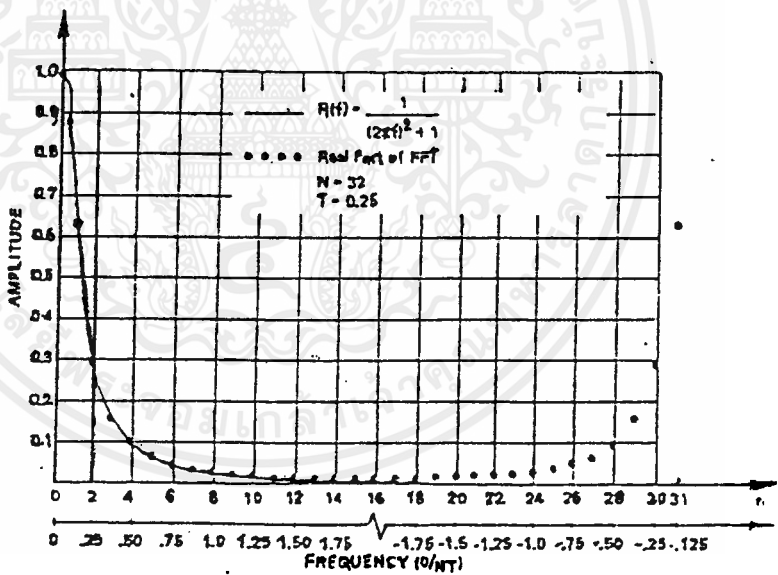
$$\begin{aligned} 8C_x(5) &= X_3(5) = X_2(4) - A_6X_2(5) \\ 8C_x(6) &= X_3(3) = X_2(2) - A_6X_2(3) \\ 8C_x(7) &= X_3(1) = X_2(6) - A_6X_2(7) \end{aligned} \dots\dots\dots (28)$$

จากขั้นตอนข้างต้น เป็นขั้นตอนและแนวทางการเขียน โปรแกรม FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 3.2 แสดงตัวอย่างการแปลงฟูเรียร์ โดยผ่าน FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5) การประยุกต์ใช้งาน FFT

3.5.1 ความละเอียด (resolution) ของ FFT

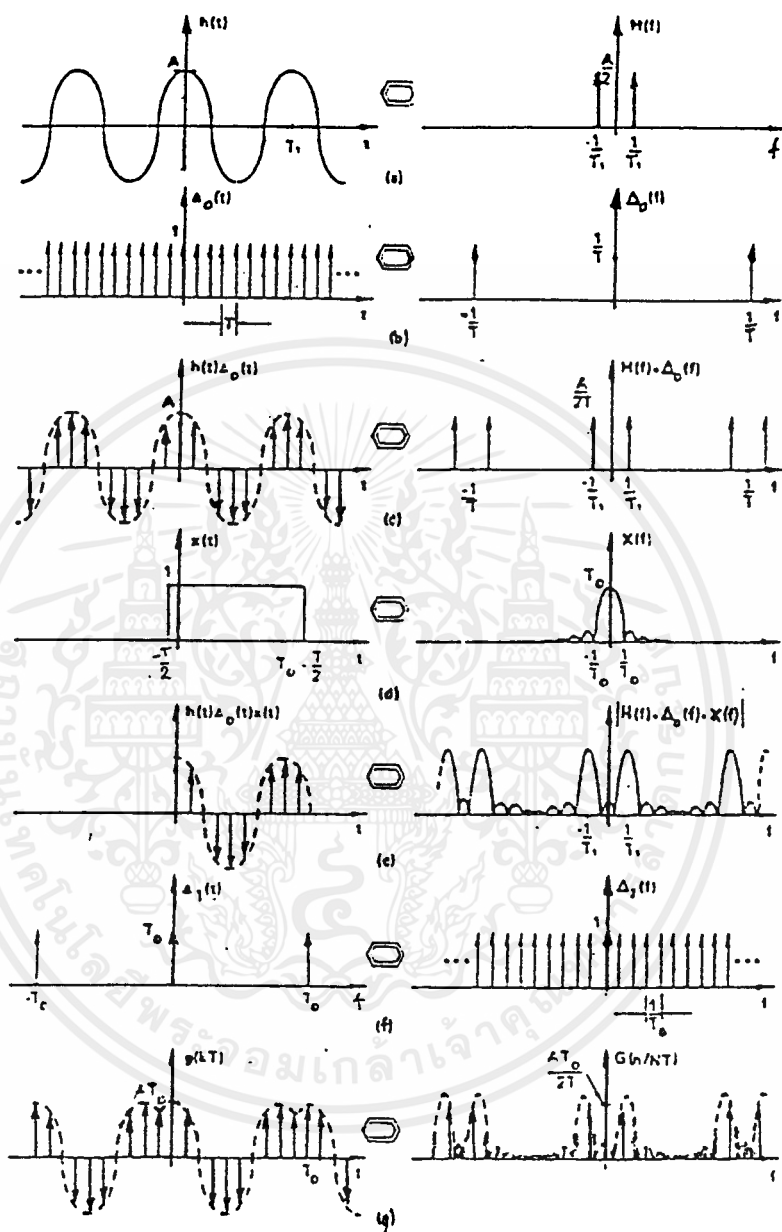
ผลลัพธ์ของ FFT ในรูปที่ 3.2b และ 3.2c แต่ละความถี่มีระยะต่างกัน $f_0 = 1/NT$ ดังนั้นจุดสุ่มในโดเมนความถี่จึงมีค่าตั้ง $0/NT$, $1/NT$, $2/NT$, เป็นเทอมที่แสดงถึงความละเอียดของ FFT แต่ละสมาชิกของความถี่ที่ได้เรียกว่า สมาชิกความละเอียด (resolution element) หรือ เซลความละเอียด (resolution cell) เราอาจจะคิดว่าความละเอียดเป็นเทอมที่ทำให้สามารถแยกความแตกต่างระหว่างแต่ละความถี่ โดยประมาณค่าความถี่ที่ได้จากการแปลงเป็นความถี่ตั้งแต่ $0/NT$, $1/NT$, $2/NT$ ไปจนถึง $(N/2)/NT$ ซึ่งเราอาจจะลดระยะห่างระหว่างแต่ละความถี่ได้ โดยการเพิ่มจำนวนข้อมูล N ซึ่งก็คือการเพิ่มระยะการตัดทอน (truncation) ของฟังก์ชันที่ต้องการแปลง ถ้าค่าของ N เพิ่มขึ้นเป็น 2 เท่า ระยะห่างของความถี่ก็ลด 2 เท่า เช่นเดียวกัน

จากรูปที่ 3.3 ระยะห่างของความถี่ (ความละเอียด) ของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ถูกกำหนดจากความกว้างของสี่เหลี่ยมที่ไปคูณ หรือ ตัดทอนฟังก์ชันก่อนทำการแปลง การตัดทอนในโดเมนเวลา นั่นก็คือ การทำคอนโวลูชันของฟังก์ชัน $[\sin(f)/f]$ กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้น การทำคอนโวลูชันกับฟังก์ชัน $[\sin(f)/f]$ ทำให้ผลจากการแปลงคลุ่มเคลือ และ เลื่อนลง ถ้าฟังก์ชันที่ตัดทอนในโดเมนเวลามีความกว้างมากขึ้น จะทำให้ผลของฟังก์ชัน $[\sin(f)/f]$ แคบลง และความคลุ่มเคลือของความถี่น้อยลงด้วย ความคลุ่มเคลือของความถี่ยิ่งน้อยเท่าใด ก็ยิ่งทำให้ประสิทธิภาพของการใช้ FFT ในการแก้ปัญหาแต่ละงาน เป็นไปได้ดีมากยิ่งขึ้น

ความเข้าใจผิดโดยทั่ว ๆ ไปของผู้ใช้ FFT คือ การเพิ่มจำนวนข้อมูล N โดยการต่อเติม 0 และ ตัดทอนฟังก์ชัน และตีความหมายฟังก์ชันความถี่ที่ได้ เป็นฟังก์ชันความถี่ที่มีความละเอียดสูง ตอนนี้เรามาเปรียบเทียบรูปที่ 5.2 (ในบทที่ 5) กับรูปที่ 3.4 ในรูป 3.4a แสดงผลจากการแปลงดิสครีตฟูเรียร์ เช่นเดียวกับรูปที่ 5.2

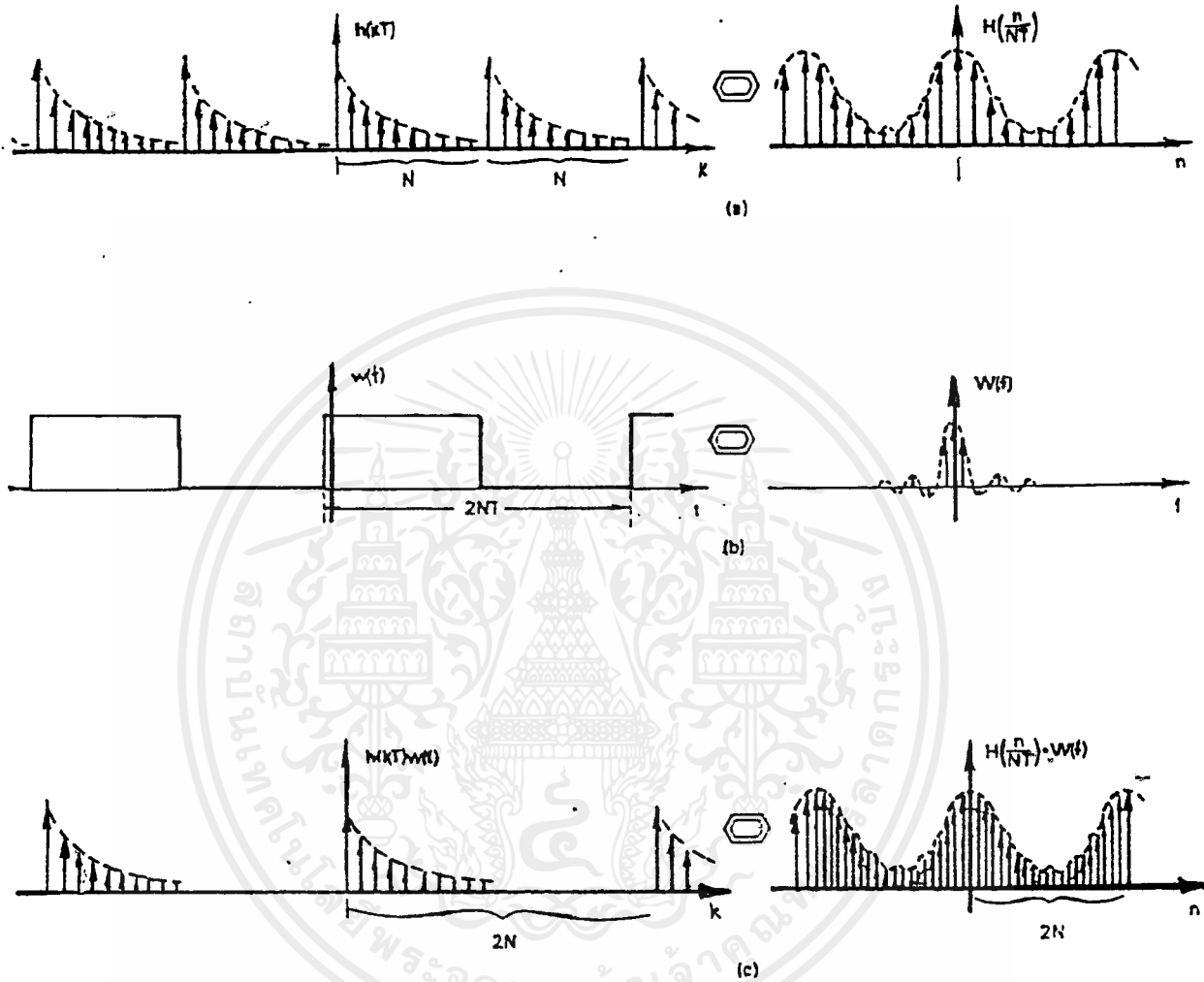
เราต้องการการตรวจสอบผลของการเพิ่ม \circ ให้กับฟังก์ชันเวลาของรูปที่ 3.4a สมมุติว่าจำนวน \circ ที่เพิ่มเข้าไปเท่ากับ N ซึ่งทำได้โดยการคูณด้วยฟังก์ชันคาบเวลาในรูปที่ 3.4b ผลของความถี่ที่ได้แสดงในรูปถัดมา การคูณทำให้ฟังก์ชันมีคาบเวลาเท่ากับ $2N$ โดยที่จุดที่ไม่ใช่ \circ เท่ากับจุดสุ่ม N จำนวนในรูป 3.4a ผลของฟังก์ชันความถี่ที่ได้คือ การทำคอนโวลูชันของฟังก์ชันความถี่ของรูปที่ 3.4a และรูปที่ 3.4b แต่ความละเอียดของความถี่ได้ถูกกำหนดไว้ในรูป 3.4a แล้ว การทำคอนโวลูชันเป็นการเพิ่มจำนวนจุดสุ่ม ในโดเมนความถี่ โดยการเฉลี่ยค่า (interpolation) ด้วยฟังก์ชัน $[\sin(f)/f]$ กับผลการแปลงฟูเรียร์ ของฟังก์ชันเริ่มต้น ดังนั้น ถึงแม้ว่าระยะห่างของผลลัพธ์จาก FFT จะมีระยะใกล้ขึ้น โดยการต่อเติม \circ ก็ตาม แต่ความละเอียดก็ไม่ได้เปลี่ยนแปลง ความละเอียดของ FFT ไม่สามารถทำให้ความละเอียดของข้อมูลเพิ่มขึ้น โดยการต่อเติม \circ นอกเสียจากว่า ฟังก์ชันที่จะทำการต่อเติม \circ มีค่า \circ ครอบคลุมช่วงนั้นอยู่แล้ว

จากหัวข้อนี้ เราอาจสรุปได้ว่า ความละเอียดนั้นต้องพิจารณาจากช่วงเวลาในโดเมนเวลาของสัญญาณ และในการประยุกต์ใช้ FFT ช่วงเวลาของสัญญาณถูกตั้งใหม่เป็นช่วงระยะห่างของการตัดทอนข้อมูล



รูปที่ 3.3 แสดง DFT ของสัญญาณคาบเวลา โดยมีช่วงการตัดทอน น้อยกว่า 1 คาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 แสดงตัวอย่างการเพิ่มความละเอียดของ FFT โดยการต่อเติม ๐

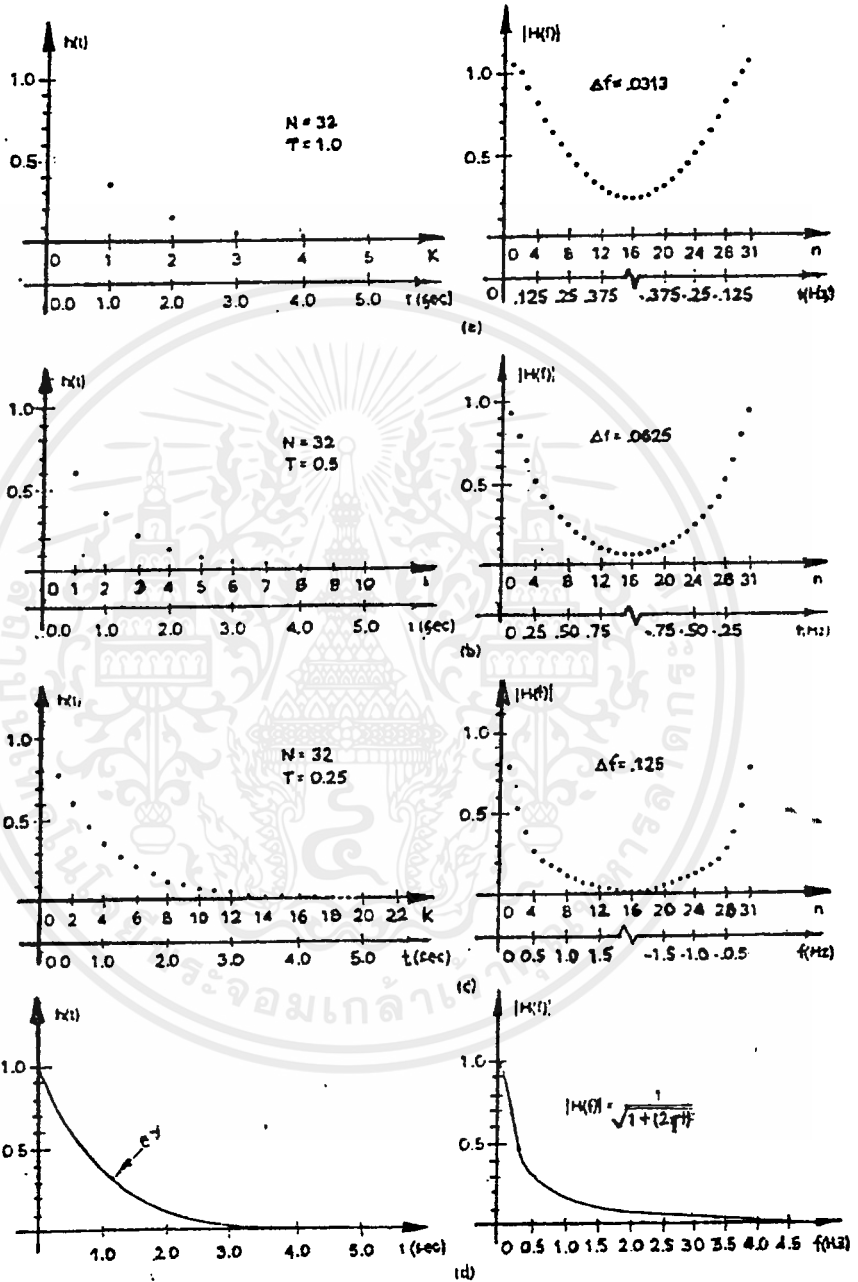
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 ผลความผิดเพี้ยนจาก FFT (FFT Aliasing)

ปัญหาหนึ่งที่เรพบกันมากในการคำนวณการแปลงฟูเรียร์ FFT คือ ความผิดเพี้ยน (aliasing) จากที่เคยกล่าวมาแล้ว ในการสุ่มข้อมูล ความผิดเพี้ยน จะเกิดขึ้นเมื่อ จุดสุ่มของฟังก์ชันเวลา มีระยะห่างของการเก็บมากเกินไป ผลของฟังก์ชันความถี่ที่ผิดเพี้ยนนี้ เราเรียกว่า การพับ (fold) หรือการซ้อน (overlap) ในตัวมันเอง ซึ่งได้แสดงตัวอย่างไว้ในรูปที่ 3.5

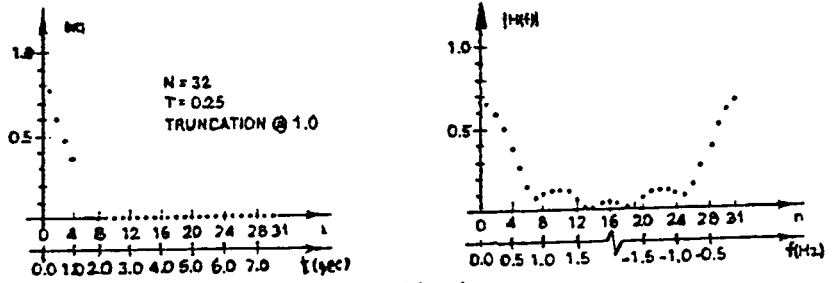
ในรูปที่ 3.5 เรามีจุดสุ่มของฟังก์ชัน $h(t) = e^{-t}$, $t > 0$ โดยช่วงเวลาการสุ่ม $T = 1.0, 0.5$ และ 0.25 ตามลำดับ จำนวนข้อมูลตั้งให้เท่ากับ 32 สำหรับแต่ละกรณี ขนาดของการแปลงฟูเรียร์ ที่ได้ คำนวณโดยใช้ FFT ซึ่งได้แสดงรวมไว้ในรูปที่ 3.5a ถึงรูปที่ 3.5c แล้วจากผลที่ได้จาก FFT จะเห็นว่าเมื่อ $T = 1.0$ ผลที่ได้จะมีความผิดเพี้ยนมากที่สุด (ขนาดของการแปลงฟูเรียร์แบบต่อเนื่องได้แสดงไว้ในรูป 3.5d) และจะเห็นว่าความผิดเพี้ยนมีค่าลง เมื่อมีคาบเวลาเท่ากับ 0.5 และ เมื่อคาบเวลาเท่ากับ 0.25 ผลที่ได้จะมีความคล้ายคลึงกับผลที่ได้จากการคำนวณแบบต่อเนื่องมากที่สุด

ในรูปที่ 3.5 ได้แสดงหลักการลดความผิดเพี้ยนโดยการลดช่วงการสุ่ม ในกรณีนี้จะเห็นว่า ไม่มีผลของการตัดทอน เนื่องจาก T มีค่าต่ำ จึงทำให้ NT มีค่ามากกว่าความกว้างของช่วงที่ไม่เป็น 0 (nonzero interval) ของฟังก์ชันมาก

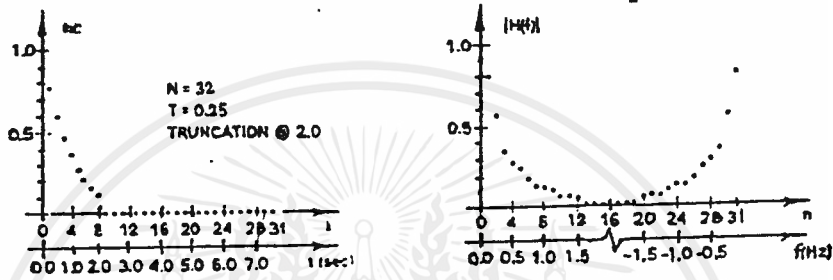


รูปที่ 3.5 แสดงตัวอย่างของความผิดพลาดในโดเมนความถี่จากอัตราการสุ่ม

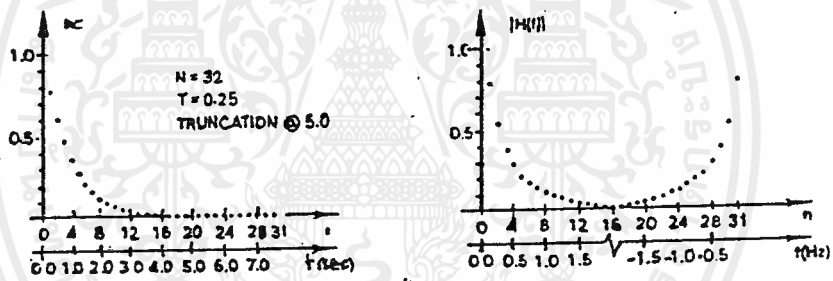
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



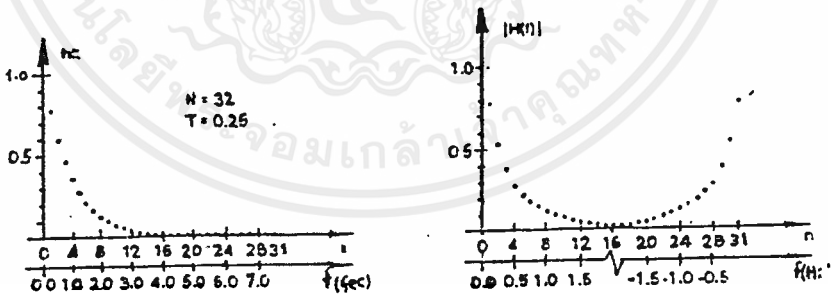
(a)



(b)



(c)



(d)

รูปที่ 3.6 แสดงการตัดทอนในโดเมนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3) การตัดทอนของ FFT ในโดเมนเวลา

FFT Time-Domain Truncation

ปัญหาความผิดพลาดอื่นที่มักจะพบในการประยุกต์ใช้ FFT ในการแปลงฟูเรียร์ คือความผิดพลาดเนื่องจากการตัดทอนข้อมูลในโดเมนเวลา (Time-Domain Truncation) ความผิดพลาดนี้เกิดจากการเลือกกลุ่มข้อมูลจุดสุ่มเพื่อหาลักษณะของฟังก์ชันเวลา ไปตัดทอนลักษณะบางส่วนของรูปคลื่นเบื้องต้น รูป 3.6a และ 3.6b แสดงตัวอย่างของจุดนี้ เมื่อเราตัดทอนฟังก์ชัน $h(t)$ ที่ $NT = 1, 2, 5$ S. ตามลำดับ ขนาดของการแปลงฟูเรียร์ที่ผ่าน FFT ได้แสดงในรูป

การตัดทอนที่ 1 วินาที ทำให้ผลที่ได้จาก FFT มีการกระเพื่อม (rippling) เมื่อใช้ช่วงเวลาของการตัดทอนมากขึ้น เท่ากับ 2 วินาที ผลของ FFT ที่ได้มีการกระเพื่อมน้อยลง และ เมื่อเพิ่มช่วงของการตัดทอนให้เป็น 5 วินาที ผลของ FFT ที่ได้ไม่ปรากฏการกระเพื่อม เหมือนกับรูปที่ถูกต้องในรูปที่ 3.6c รูปที่ 3.6 แสดงผลของการทดลอง โดยการพิจารณาหาช่วงเวลาการตัดทอนที่ดีที่สุด โดยเพิ่มช่วงเวลาการตัดทอน ซึ่งเราก็ได้เห็นผลของการลดลงของการกระเพื่อม

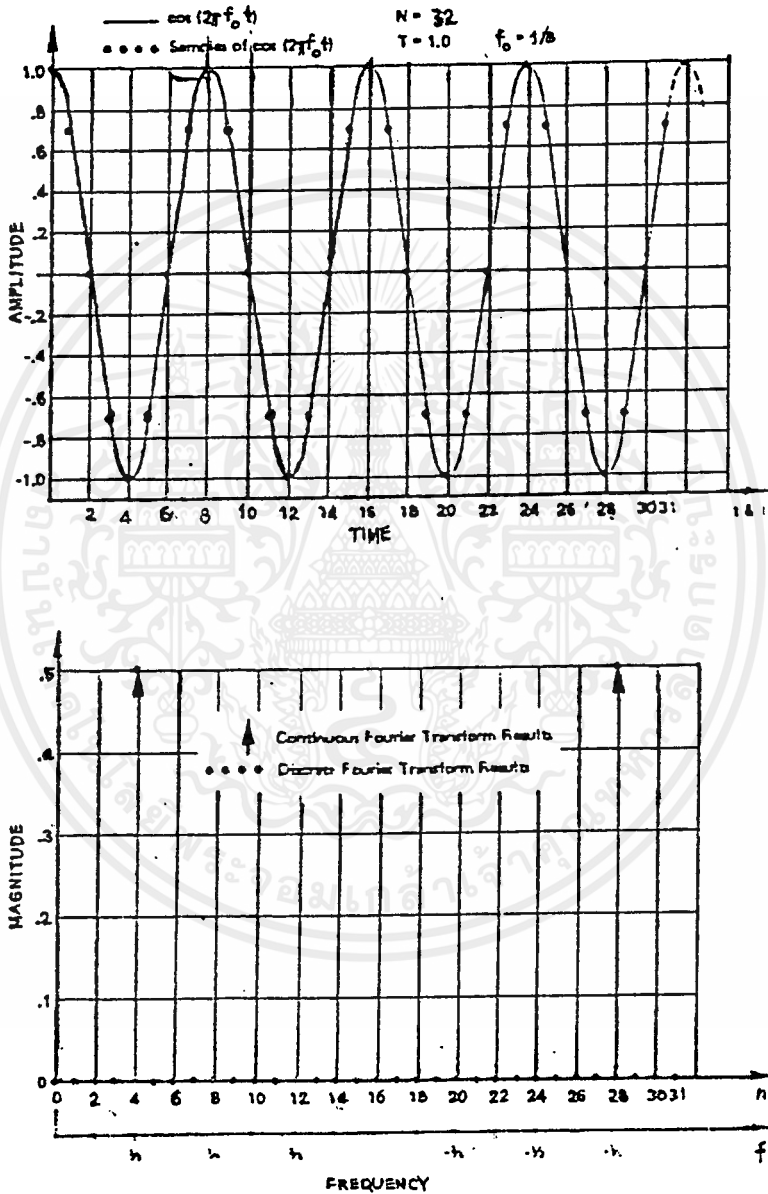
3.5.4) FFT ฟังก์ชันคาบเวลา

การคำนวณหา FFT ของฟังก์ชันคาบเวลา เราต้องเลือกช่วงการสุ่มและช่วงการตัดทอน จากที่กล่าวมาแล้ว การเลือก T ต้องเลือกให้เหมาะสมเพื่อไม่ให้เกิดความผิดพลาด การตัดทอนของฟังก์ชันเวลาคาบเวลาเป็นปัญหาใหม่ที่ไม่ได้อธิบายไว้ในหัวข้ออื่นมาก่อน

ถ้าเราเลือกจำนวนจุดสุ่ม N ของการแปลงฟูเรียร์ เท่ากับหนึ่งคาบเวลาพอดีหรือเท่ากับจำนวนเท่าของคาบเวลา ผลของความถี่ที่ได้จะตรงกับสัญญาณเริ่มต้น

เพื่ออธิบายจุดนี้ เราทำการคำนวณ FFT ของฟังก์ชันโคไซน์ ในรูปที่

3.7(a) โดยมีช่วงการสุ่ม $T = 1.0$ s และจำนวนจุดสุ่มเท่ากับ 32 และแสดงผลลัพธ์ในรูปที่ 3.7 (b) โดยแสดงเป็นขนาดที่ได้จาก FFT ของจุดเหล่านี้ จากรูปจะเห็นว่า จุดอื่นเป็นศูนย์หมด นอกจากจุดที่เป็นความถี่จริง



รูปที่ 3.7 แสดงผลจาก FFT ของฟังก์ชันคาบเวลา โดยมีช่วงการตัดตอนเท่ากับจำนวนเท่าของคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ฟังก์ชันถ่วงน้ำหนักข้อมูล (Data Weighting function)

จากหัวข้อที่แล้ว การตัดทอน (Truncation) อาจจะทำให้ผลจากการแปลงฟูเรียร์คลาดเคลื่อน การประมวลผลข้อมูลจำนวนจำกัด N ของฟังก์ชันคาบเวลาที่ไม่รู้คาบเวลาแน่นอน จำเป็นต้องใช้วินโดว์ของข้อมูลหรือที่เรียกว่าฟังก์ชันถ่วงน้ำหนักข้อมูลในหัวข้อนี้ เราจะได้อธิบายเทคนิคการใช้ฟังก์ชันถ่วงน้ำหนักเพื่อลดผลเสียที่เกิดจากการตัดทอนให้เหลือน้อยที่สุด

3.6.1 ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม

(Rectangular Weighting Function)

จากรูปที่ 3.2 เราสามารถสังเกตการณ์โดยการคูณด้วย อิมพัลส์ฟังก์ชัน (Impulse Function) ซึ่งได้แสดงไว้ในรูป 3.2(b) นำผลที่ได้ (รูป 3.2(c)) ไปคูณด้วยฟังก์ชันตัดทอนสี่เหลี่ยม ซึ่งแสดงไว้ในรูปที่ 3.2(b) เพื่อจำกัดจำนวนของจุดสุ่มให้เหลือเท่ากับ N ในที่นี้การตัดทอนของโดเมนเวลา (Time Domain) คือการถ่วงน้ำหนักข้อมูลโดยคูณด้วยฟังก์ชันสี่เหลี่ยม ผลของการตัดทอนโดเมนเวลาได้แสดงไว้ในรูปที่ 3.2(e) เรารู้ว่าฟังก์ชันโดเมนความถี่ (Frequency Domain) ของฟังก์ชันสี่เหลี่ยมคือฟังก์ชัน $[\sin(f)/f]$ ดังนั้นการแปลงเป็นโดเมนความถี่ของฟังก์ชันเวลาที่ตัดทอนแล้ว จึงเป็นการทำคอนโวลูชัน (Convolution) ของฟังก์ชันเวลากับฟังก์ชันสี่เหลี่ยม ทำให้ผลที่ได้มีองค์ประกอบของความถี่เพิ่มขึ้นมา ซึ่งก็คือลอนข้าง (side lobe) ของฟังก์ชันสี่เหลี่ยม องค์ประกอบที่เพิ่มขึ้นมานี้เรียกว่า ส่วนรั่วไหล (Leakage) ทำให้อิมพัลส์ของความถี่ของฟังก์ชันรั่วออกไปที่ลอนข้างของฟังก์ชัน $[\sin(f)]/f$

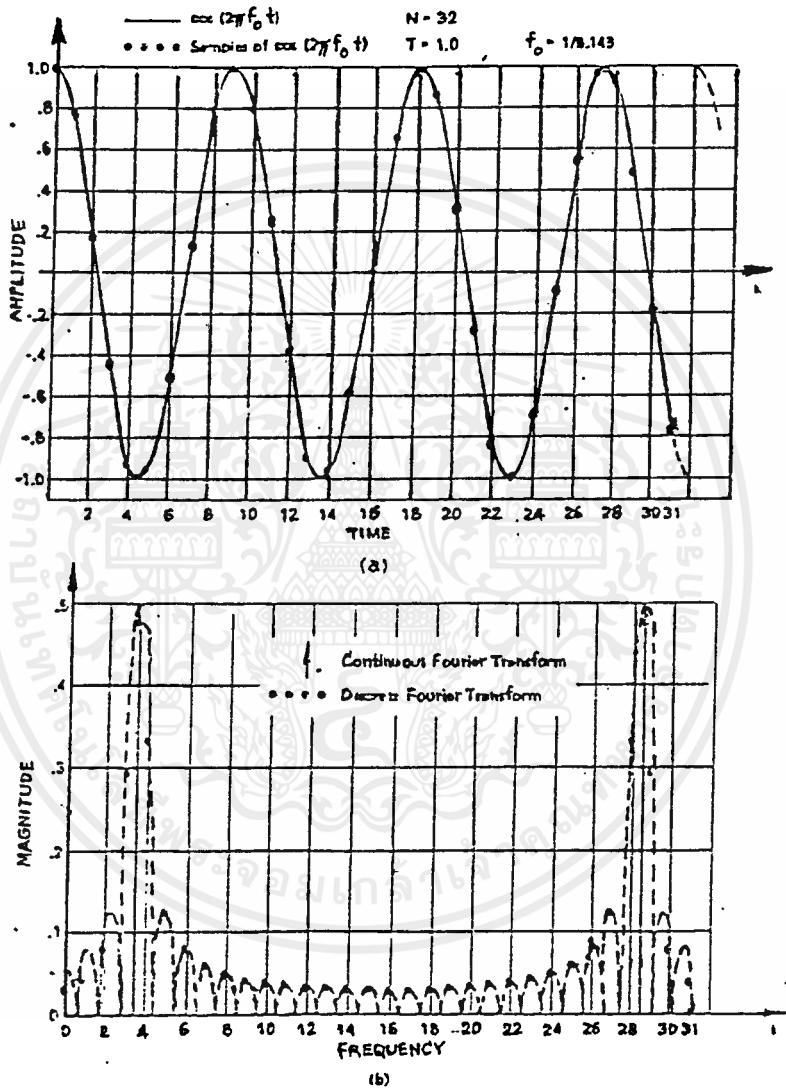
ถึงแม้ว่าฟังก์ชันในโดเมนเวลาเริ่มต้นเป็นสัญญาณชายนี แต่เมื่อผ่านการสุ่มแล้ว รูปคลื่นของสัญญาณที่ถูกสุ่มในโดเมนเวลาจะไม่ใช่สัญญาณชายนีอีก เพราะว่าช่วงของการตัดทอน (Truncation Interval) ไม่เท่ากับคาบเวลาหรือจำนวนเท่าของคาบเวลา (เป็นจำนวนเต็ม) ทำให้คอนโวลูชันในโดเมนเวลาในรูปที่ 3.2(e) และ 3.2(f) ไม่ตรงกับฟังก์ชันคาบเวลาเริ่มต้นเนื่องจากการทำคอน

โวลูชันของฟังก์ชันเวลาเป็นแบบไม่ต่อเนื่อง (discontinuous) จึงได้แสดงผลของการเกิดการกระเพื่อม (Rippling Effect) ไว้ในแบบไม่ต่อเนื่องในรูปที่ 3.2(g) ด้วย

ต่อไปเราจะสาธิตตัวอย่างของผลที่เกิดจากการใช้ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมดังนี้

สมมติว่าเราต้องคำนวณ FFT ของฟังก์ชันโคไซน์ (Cosine Function) ซึ่งแสดงไว้ในรูปที่ 3.7(a) โดยมีคาบเวลา $T = 1.0$ และ $N = 32$ ในรูปที่ 3.7(b) เราได้แสดงขนาดของการแปลงดิสครีตฟูเรียร์ (Discrete Fourier Transform) ของจุดสุ่มในรูปที่ 3.7(a) FFT จะทำให้ผลเป็นความถี่ไม่ต่อเนื่องทางด้านบวกทั้งหมด จากที่ได้อธิบายในหัวข้อที่แล้ว องค์ประกอบของความถี่เพิ่มขึ้นมาที่เรียกว่าส่วนรั่วไหลเป็นผลมาจากลักษณะการมีลอนข้างของฟังก์ชัน $[\sin(f)]/f$

เพื่อลดผลจากการรั่วไหล เราจำเป็นต้องใช้วิธีการตัดทอนโดเมนเวลาหรือฟังก์ชันถ่วงน้ำหนักที่มีลักษณะของลอนเข้าในโดเมนความถี่เล็กกว่าของฟังก์ชันสี่เหลี่ยม ลอนข้างที่เล็กกว่านี้ทำให้การรั่วไหลของผลลัพธ์จาก FFT ลดน้อยลงเพื่อให้เข้าใจจุดนี้ได้ชัดเจนยิ่งขึ้น ขอให้พิจารณารูปที่ 5.2 อีกครั้งหนึ่ง ถ้าเปลี่ยนฟังก์ชันตัดทอนในรูปที่ 3.2(d) เป็นฟังก์ชันที่มีลอนข้างต่ำ จะทำให้การประมาณค่าของการแปลงฟูเรียร์ดีขึ้น การใช้ฟังก์ชันถ่วงน้ำหนักข้อมูลเพื่อตัดทอนและถ่วงน้ำหนักข้อมูลต้องทำกับจุดสุ่ม N ก่อนคำนวณ FFT



รูปที่ 3.8 แสดงผลจาก FFT ของฟังก์ชันคาบเวลา โดยมีช่วงการตัดทอน
ไม่เท่ากับจำนวนเท่าของฟังก์ชันคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6.2 ลักษณะของฟังก์ชันถ่วงน้ำหนักชนิดต่าง ๆ

ในรูปที่ 3.9(a) ได้แสดงฟังก์ชันตัดทอนหรือถ่วงน้ำหนักที่นิยมใช้กับ FFT ผลของการตอบสนองความถี่ของฟังก์ชันได้แสดงไว้ในรูปที่ 3.9(b) และในตารางที่ 2 แสดงข้อมูลการเปรียบเทียบลักษณะของฟังก์ชันถ่วงน้ำหนักแต่ละตัว ทั้งในโดเมนเวลาและโดเมนความถี่ โดยมีจุดศูนย์กลางอยู่ที่จุดศูนย์กลางเพื่อง่ายต่อการสังเกต

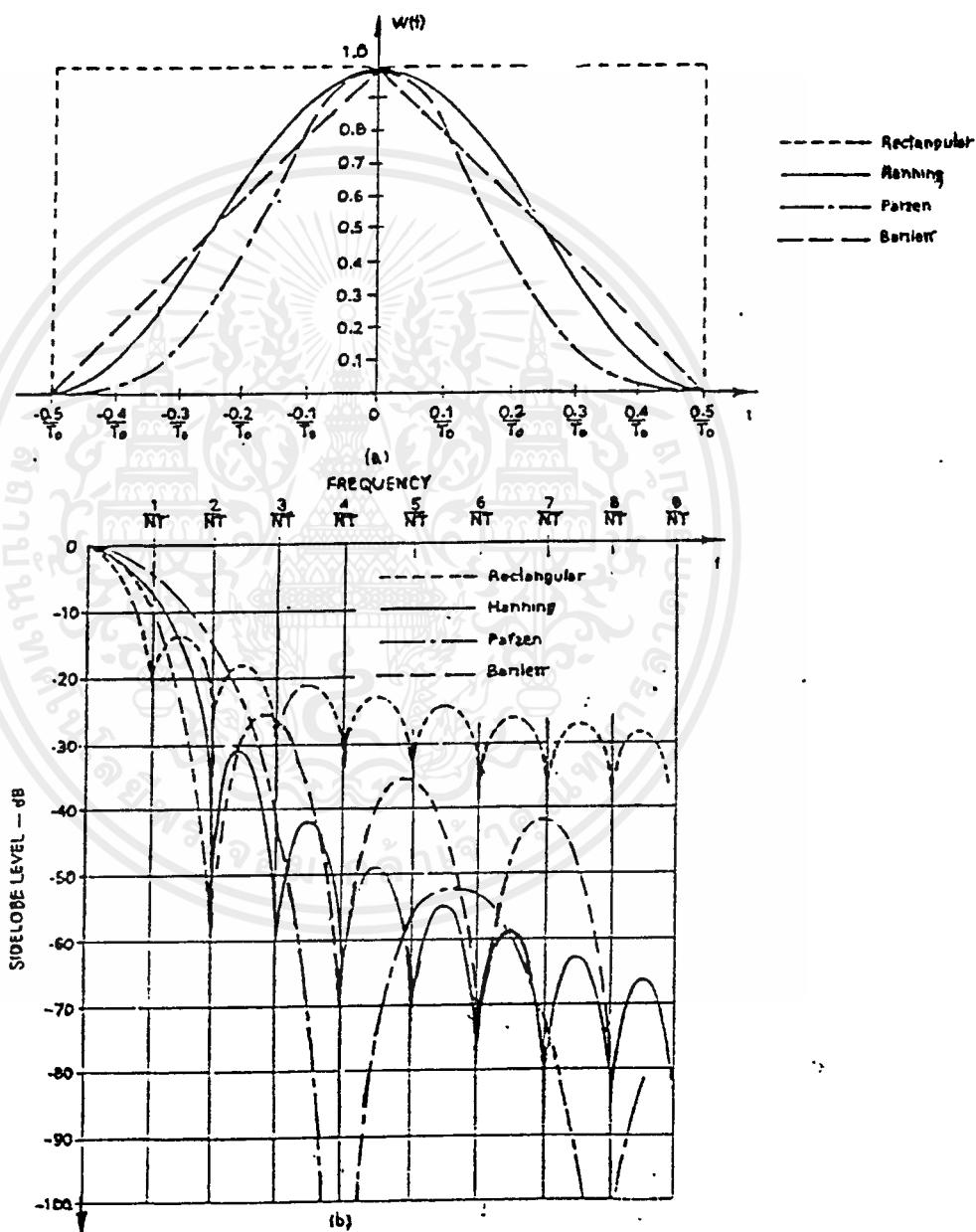
จากรูปที่ 3.9(b) จะเห็นว่าฟังก์ชันถ่วงน้ำหนักทั้งหมดมีลอนข้างในโดเมนความถี่ที่มีขนาดเล็กกว่าฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมทั้งนั้น ซึ่งจะช่วยให้ผลของการรั่วไหลน้อยลง อย่างไรก็ตามฟังก์ชันถ่วงน้ำหนักทั้งหมดนี้มีลักษณะของลอนหลัก (main lobe) ที่กว้างกว่า ดูรูปที่ 5.4(d) และ (e) ในบทที่ 5 อีกครั้ง ลองเปรียบเทียบผลที่ได้เมื่อใช้ฟังก์ชันถ่วงน้ำหนักตามรูปที่ 3.9 (b) จะเห็นว่ายิ่งลอนหลักมีความกว้างมากขึ้นเท่าใด ผลที่ได้จาก FFT ยิ่งมีความคลุมเคลือมากยิ่งขึ้น นั่นก็หมายความว่า ถ้าลอนหลักมีขนาดกว้าง ก็จะทำให้แยกความแตกต่างของแต่ละความถี่ได้ยาก

ข้อได้เปรียบเสียเปรียบระหว่างส่วนรั่วไหล (ระดับการรั่วไหล) และความละเอียด (แบนวิทของลอนหลัก) เป็นที่รู้จักกันดีในงานวิทยาศาสตร์สาขาต่างๆ ในตารางที่ 3.2 แสดงระดับสูงสุดของลอนข้างและแบนวิท 3 dB สำหรับแต่ละฟังก์ชันถ่วงน้ำหนัก สำหรับงานทดลองทั่วไปนิยมใช้ฟังก์ชันแฮนนิ่ง (Hanning function) เพราะโครงสร้างง่ายกว่าแบบอื่น การเลือกใช้ฟังก์ชันถ่วงน้ำหนักที่ทำให้ผลดีที่สุดต้องขึ้นกับลักษณะงานที่เรานำไปใช้ เป็นสำคัญ

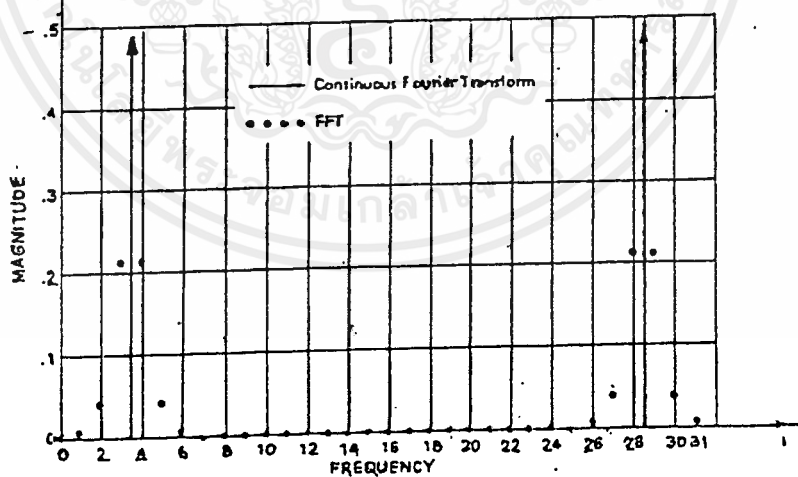
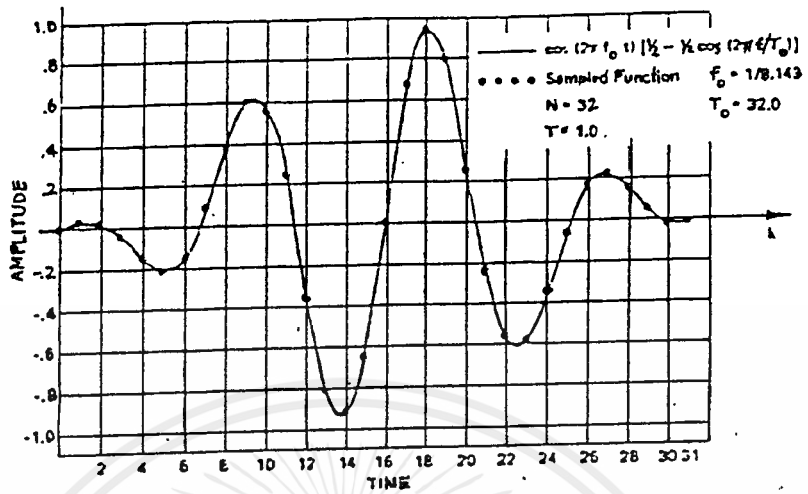
ผลที่ได้จาก FFT จะมีช่วงความถี่ $f_0 = 1/NT$ ซึ่งอาจทำให้เกิดความสับสนกับค่า $1/NT$ ของฟังก์ชันถ่วงน้ำหนักที่เลือกใช้ ผลของความละเอียดของความถี่ของผลลัพธ์ที่ได้จาก FFT เป็นเพียงฟังก์ชันของแบนวิทของฟังก์ชันถ่วงน้ำหนักแต่ละตัวเท่านั้น ดูรูปที่ 3.9(b) ดังนั้นการใช้ค่านิยามของความละเอียด $1/NT$ ต้องทำอย่างระมัดระวังและระลึกอยู่เสมอว่าเป็นเพียงช่องว่างระหว่างความถี่ที่เกิดจากผลที่ได้จาก FFT เท่านั้น และไม่เกี่ยวกับ $1/NT$ ของวินโดว์ที่ใช้

ต่อไปเราจะแสดงผลจากการใช้ฟังก์ชันถ่วงน้ำหนักที่มีลอนข้างต่ำเพื่อ

ให้เห็น ลักษณะการนำไปใช้ลดการรั่วไหลของผลจากการตัดทอนในโดเมนเวลา ในรูปที่ 3.10(a) เราได้แสดงรูปคลื่นโคไซน์ที่คูณด้วยฟังก์ชันถ่วงน้ำหนักแฮนนิ่งที่แสดงไว้ในรูป 3.9(a) แล้ว รูปที่ 3.10(b) แสดงจุดสุ่มจาก FFT ของรูปที่ 10 (a) จากรูปจะเห็นว่าส่วนรั่วไหลลดลงไปอย่างมาก แต่ องค์ประกอบหลักของความถี่มีระยะกว้างขึ้น หรือ คลุมเคลือเมื่อเปรียบเทียบกับจุดของอิมพัลส์ฟังก์ชัน เพราะผลจากการคอนโวลูชันความถี่ของอิมพัลส์ฟังก์ชัน กับผลการแปลงฟูเรียร์ฟังก์ชันถ่วงน้ำหนัก

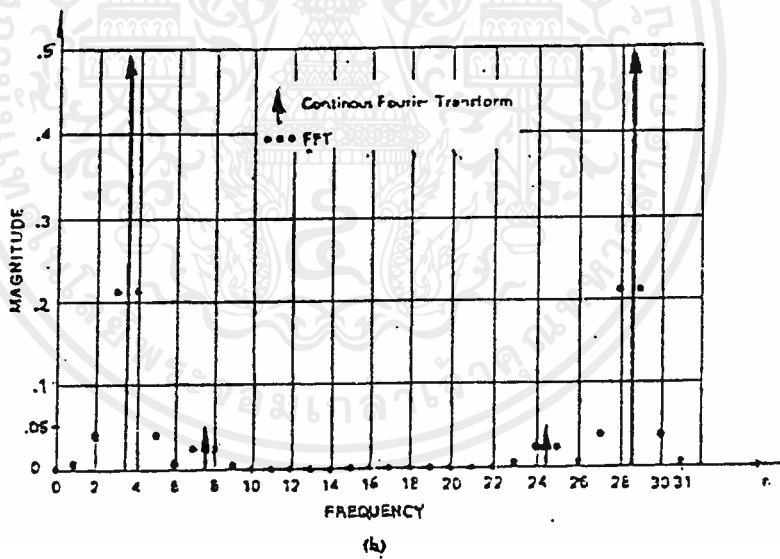
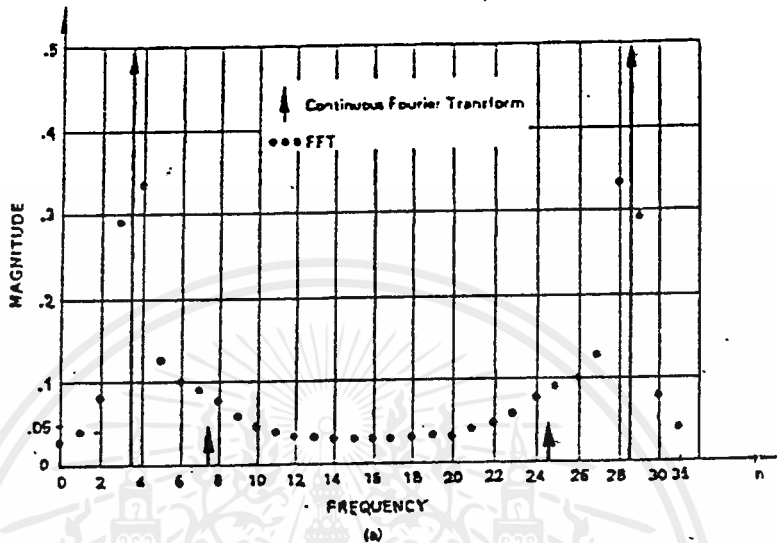


รูปที่ 3.9 แสดงฟังก์ชันถ่วงน้ำหนัก หรือ วินโดว์ของ FFT



รูปที่ 3.10 แสดงตัวอย่างการใช้ Hanning function เพื่อลดส่วนรั่วไหลของการคำนวณด้วย FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 (a) แสดงตัวอย่างที่คลุมเคลือเนื่องจากผลของส่วนรั่วไหลที่ลอนข้าง
(b) แสดงสัญญาณที่ตรวจจับได้หลังจากผ่าน Hanning function

บทที่ 4

ลักษณะของเสียงพูด

คนเราเปล่งเสียงพูดด้วยอวัยวะที่ใช้ในการออกเสียง (Organs of Speech) โดยเปล่งเสียงสำคัญตามที่มีอยู่ในระบบภาษาของตน แม้ว่าคนที่อยู่ในสังคมเดียวกันจะใช้ภาษาเดียวกัน แต่ถ้าพิจารณาเสียงที่เปล่งออกมาจริงๆ ในแต่ละครั้ง ก็อาจจะสังเกตเห็นลักษณะที่แตกต่างกันได้ เราจึงสามารถจำเสียง, จำผู้พูด, จำคนที่เราค้นเคยได้ เสียงพูดนี้มีลักษณะที่จะอธิบายได้ ด้วยหลักเกณฑ์ทางวิทยาศาสตร์ซึ่งเป็นสากลแม้ภาษาหนึ่งๆ จะมีเสียงที่แตกต่างกันไป แต่แต่ละเสียงก็สามารถที่จะนำมาพิจารณา และอธิบายให้รู้ลักษณะการออกเสียง และตำแหน่งที่เกิดของเสียงได้ คำอธิบายนี้จำทำให้เข้าใจลักษณะเสียงทุกเสียง ซึ่งวิชาที่ว่าด้วยเสียงพูดเรียกว่า "วิชาสัทศาสตร์ (Phonetics)"

ในการศึกษาเรื่องเสียงพูดจะสามารถแบ่งได้เป็นสองลักษณะคือ

1. สรีรศาสตร์ (Articulatory Phonetics) เป็นการศึกษาเสียงพูดจากอวัยวะและการเคลื่อนไหวของอวัยวะที่ทำให้เกิดเสียงพูด การอธิบายก็จะอธิบายโดยอาศัยลักษณะและการเคลื่อนไหวของอวัยวะที่เกี่ยวข้องในการเปล่งเสียงพูดนั้น
2. กลศาสตร์ (Acoustic Phonetics) เป็นการศึกษาเสียงพูดจากลักษณะคลื่นเสียงที่ผู้พูดเปล่งออกมาแล้ว และผู้ฟังได้ยิน ว่ามีลักษณะทางแลศาสตร์เป็นอย่างไร การศึกษาตามแนวนี้ต้องอาศัยความรู้ทางฟิสิกส์และคณิตศาสตร์เข้าช่วยอธิบายลักษณะของคลื่นเสียง

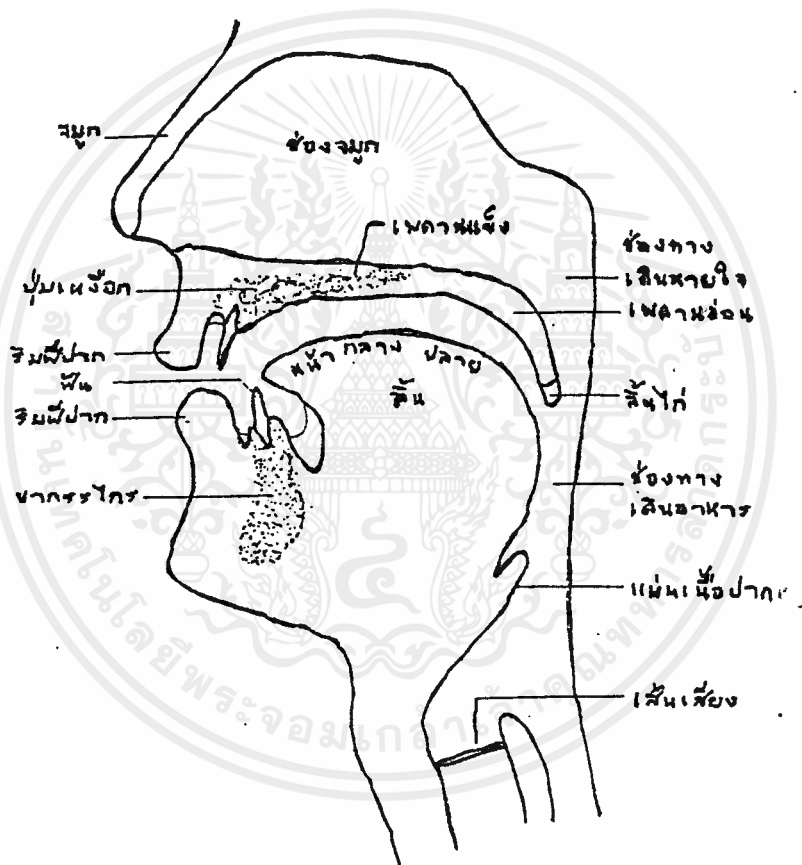
4.1 อวัยวะที่ใช้ในการออกเสียง (Organs of Speech)

อวัยวะที่ใช้ในการออกเสียงพูดมีอยู่หลายส่วน แต่ละส่วนสามารถทำให้เสียงพูดแตกต่างกันไป อวัยวะเหล่านี้ได้แก่ ปากและส่วนต่าง ๆ ในปาก, ช่องคอ, กล่องเสียง, ช่องปากและช่องจมูก อวัยวะที่ใช้ในการออกเสียงแบ่งได้เป็นสองพวก

ได้แก่

1. อวัยวะที่ใช้ในการกระทำอาการ(Articulator)หมายถึงอวัยวะส่วนที่เคลื่อนไหว เพื่อผลักลมไปยังส่วนต่าง ๆ อวัยวะตัวกระทำอาการที่สำคัญ คือ ลิ้น ซึ่งเป็นส่วนที่เคลื่อนไหวได้มากที่สุด

2. อวัยวะที่เป็นตำแหน่ง ที่เกิดเสียงต่าง ๆ (Point of Articulator) หมายถึง ตำแหน่งหรือฐานที่เกิดของเสียงต่าง ๆ เช่น ริมฝีปาก ฟัน เพดานส่วนต่าง ๆ เป็นต้น



รูปที่ 4.1 แสดงอวัยวะที่ใช้ในการออกเสียงพูด

4.2 การเกิดของเสียง (Speech Production)

ขั้นตอนการเกิดของเสียงแบ่งได้เป็น 3 ขั้นตอนใหญ่ ๆ คือ

1. ขั้นเริ่มต้น (Initiation) ขั้นตอนนี้เป็นขั้นตอนที่ ลมเริ่มถูกขับออกจากปอดเพื่อเข้าสู่ขั้นตอนที่สองต่อไป
2. ขั้นตอนดัดแปลงลมที่เส้นเสียง (Phonation) เป็นขั้นตอนที่ลมจากปอดจะผ่านมายังหลอดลมและกล่องเสียง ซึ่งที่กล่องเสียงนี้ เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิด, ปิด ทำให้เกิดเสียง ได้สองชนิด คือ เสียงก้อง (Voiced Sounds) และ เสียงไม่ก้อง (Unvoiced Sounds) อวัยวะที่ใช้ในตอนนี้คือ ส่วนที่ต่อจากปอดขึ้นมาจนถึงกล่องเสียง
3. ขั้นตอนเปลี่ยนแปลงลักษณะเสียง (Articulation) ในขั้นตอนนี้ ลมที่ผ่านออกมาจากกล่องเสียง จะถูกแปลงให้เกิดเสียงในลักษณะต่าง ๆ อวัยวะที่ใช้ในขั้นตอนนี้คือส่วนที่ต่อจากกล่อง เสียงจนถึงริมฝีปาก

เสียงที่เกิดจากขั้นตอนดังกล่าวจะถูกแบ่งเป็นสองประเภทใหญ่ ๆ คือ

1. เสียงก้อง (Voiced Sounds) เกิดเนื่องจากการออกเสียงในขณะที่เส้นเสียงถูกดึงเข้ามาใกล้กันจนเกือบปิดช่องทางลมเสียสนิท ลมที่ดันขึ้นมาจากปอดจะทำให้เส้นเสียงสั่น ลมที่ออกมาไม่สะดวก เพราะต้องบีบตัวผ่านช่องแคบเป็นจังหวะ จึงทำให้เกิดเป็นเสียงขึ้น เรียกว่า "เสียงก้อง"
2. เสียงไม่ก้อง (Unvoiced Sounds) เป็นการออกเสียงในขณะที่เส้นเสียงยังเปิดกว้าง โดยเปิดช่องระหว่างเส้นเสียงหรือช่องคอหอย (Glottis) ให้ลมหายใจผ่านเข้าออกสะดวก เรียกเสียงเหล่านี้ว่า "เสียงไม่ก้อง"

เราสามารถแบ่งวิธีการจดจำเสียงพูดออกได้เป็นสองวิธีคือ

1. แบบพิจารณา ทั้งหน่วยคำ ที่เปล่งออกมา (Isolated Words Recognition) หน่วยคำที่พิจารณาในวิธีนี้ อาจเป็นพยางค์เดี่ยว, คำ, กลุ่มคำ, วลี หรือ ประโยคก็ได้ สำหรับการวิจัยนี้จะเป็นการพิจารณาคำเดี่ยว (Word Based Recognition)
2. การจดจำโดยแยกแยะรายละเอียดของหน่วยเสียง (Phoneme Based Recognition) จะพิจารณาลักษณะของหน่วยเสียงที่มีขนาดเล็กลงไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น ระดับสระ พยัญชนะ หรือวรรณยุกต์ โดยจะใช้หน่วยเสียงย่อยเหล่านี้เป็น
หลักในการเปรียบเทียบ วิธีนี้จะสะดวกสำหรับระบบที่จะรับรู้คำจำนวนมาก โดยจะ
ใช้เป็นพื้นฐานในการวิเคราะห์ในขอบเขตที่กว้างขึ้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การวิเคราะห์เสียงพูด (Speech Analysis)

ในการเก็บหรือจดจำเสียงพูด เราต้องพยายามลดช่วงของสัญญาณเสียงพูดที่เกิดซ้ำซ้อนกัน โดยใช้ตัวแทนของเสียงพูดที่สามารถแทนลักษณะสำคัญของเสียงพูดนั้นในเทอมของค่าพารามิเตอร์ (Parameter) เพื่อให้จัดการกับข้อมูลได้ง่าย

ต่อไปเราจะได้อธิบายถึงวิธีการที่ใช้ในการวิเคราะห์เสียงพูด ทั้งในโดเมนเวลา (กระทำกับเสียงพูดที่เก็บไว้โดยตรง) และโดเมนความถี่ (ผ่านการแปลงเป็นความถี่ก่อน) เพื่อหาตัวแทนของสัญญาณเสียงพูดในรูปของพารามิเตอร์ของเสียงพูดที่เหมาะสมที่สุด สามารถนำไปใช้งานได้ และมีข่าวสารของข้อมูลครบถ้วน การวิเคราะห์ในโดเมนเวลาเราต้องการการคำนวณเพียงเล็กน้อยทำให้ถูกจำกัดให้วัดได้ในลักษณะง่าย ๆ เท่านั้น เช่น การวัดพลังงานและความเป็นคาบเวลาของสัญญาณ ในขณะที่การวิเคราะห์ในโดเมนความถี่จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพมากกว่า แต่การสุ่มสัญญาณเสียงพูด จะต้องมีความเที่ยงตรงสูง

เทคนิคที่ใช้ในการวิเคราะห์เสียงพูด อาจทำได้ทั้งแบบดิจิตอลและแบบอนาลอก การประมวลผลสัญญาณอนาลอกเป็นการใช้วงจรทางด้านอิเล็กทรอนิกส์ ซึ่งมีข้อดีอยู่ที่มีความเร็วสูง แต่จำเป็นต้องใช้วงจรเฉพาะอย่าง ทำให้ต้องต่อสายใหม่และปรับค่าใหม่ทุกครั้งที่น่าไปประยุกต์ใช้กับงานอื่น ในขณะที่เทคนิคทางดิจิตอลจะสร้างและเปลี่ยนแปลงได้ง่ายกว่า เพียงแต่ต้องการฮาร์ดแวร์หรือโปรแกรม และฮาร์ดแวร์พิเศษ (ไมโครโปรเซสเซอร์ และ ชิปต่าง ๆ) ถึงแม้ว่าอาจจะมีความเร็วในเรื่องความเร็วที่ไม่สามารถตอบสนองในเวลาจริงได้ แต่ในปัจจุบันเทคโนโลยีทางด้าน VLSI (Very Large Scale Integrated Circuit) ได้พัฒนาไปอย่างมากทำให้ลดข้อเสียเปรียบของเทคนิคทางดิจิตอลไปได้มาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1 การวิเคราะห์เสียงพูดในช่วงสั้น (Short-time Speech Analysis)

สัญญาณเสียงพูดเป็นสัญญาณที่เปลี่ยนไปตามเวลา โดยเกิดในลักษณะแบบสุ่ม (Random) โดยต้องขึ้นอยู่กับการควบคุมเสียงของผู้พูดด้วย เพราะเสียงที่เปล่งออกมาในช่วงระยะเวลาหนึ่งนั้น จะขึ้นอยู่กับรูปร่างของท่อกำทอนเสียง (Vocal Tract) และลักษณะการสั่นของเส้นเสียง (Vocal Cord) สัญญาณของเสียงพูดจึงเป็นสัญญาณที่เป็นคาบเวลาชั่วขณะ (quasi-periodic) หมายความว่าสัญญาณเสียงพูดมีคาบเวลาคงที่ในระยะเวลาสั้น ๆ และมีการเปลี่ยนแปลงในระหว่างระยะเวลาสั้น ๆ นั้น

ถ้าเราพูดช้ามาก ๆ เสียงพูดที่ได้ อาจเปลี่ยนแปลงอยู่ในช่วงมากกว่า 200 มิลลิวินาที ก็ได้ แต่การพูดโดยทั่วไปลักษณะการเปลี่ยนแปลงของเสียงพูดจะอยู่ในช่วงค่าเฉลี่ยประมาณ 80 มิลลิวินาที ในการวิเคราะห์เราจะใช้วินโดว์ (Window) วิเคราะห์สัญญาณเป็นช่วง ๆ หรือเรียกว่าการวิเคราะห์เฟรม (frame) ช่วงเวลาของวินโดว์จะมีค่าไม่แน่นอน ในบางครั้งอาจต้องใช้ถึง 100 มิลลิวินาที ในบางครั้งเมื่อทำการวิเคราะห์สัญญาณเสียงพูดที่เร็วมากและไม่มีช่วงหยุดเลย เราอาจจะต้องใช้ช่วงเวลาของวินโดว์ต่ำถึง 5-10 มิลลิวินาที

5.1.1 การใช้วินโดว์ (Windowing)

รูปแบบของการเฉลี่ย โดยปกติมีวิธีหลายอย่างเพื่อให้ได้เส้นของพารามิเตอร์เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้อง ทางเลือกทางหนึ่งที่ใช้คือขนาดของวินโดว์ที่เกี่ยวข้อง คือ

1. วินโดว์จะต้องเล็กพอ ๆ กับคุณลักษณะของคำพูด
2. วินโดว์จะต้องยาวพอที่จะใช้ในการคำนวณหาพารามิเตอร์
3. วินโดว์ที่ดีจะต้องไม่สั้นจนข้ามบางช่องของคำพูด การวิเคราะห์จะทำได้เป็นคาบ ๆ ไป ช้า ๆ กัน ตลอดที่ได้รับสัญญาณ

เงื่อนไขท้ายนี้จะขึ้นกับ frame rate (จำนวนครั้งต่อวินาทีที่กระทำต่อสัญญาณวิเคราะห์) มากกว่าขนาดของวินโดว์ ปกติ frame rate ประมาณ 2

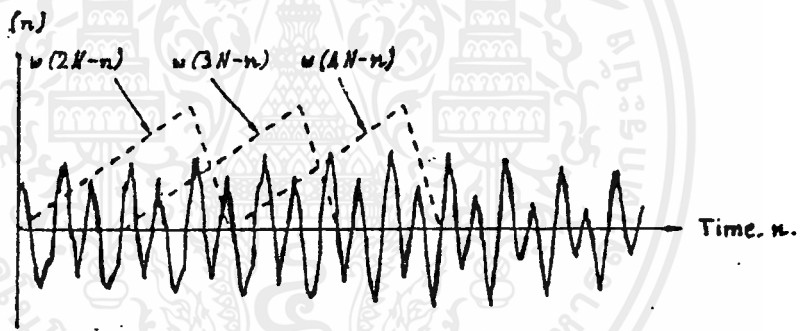
เท่าของความถี่เพื่อให้วินโดว์ทับกัน 50%

การทำวินโดว์ เป็นการคูณสัญญาณเสียง โดยวินโดว์ที่มีช่วงเวลาจำกัด (Finite-duration window) $w(n)$ ซึ่งกลุ่มของสัญญาณเสียงที่สุ่มมาจะถูกให้นำหนักโดยรูปของวินโดว์

วินโดว์อาจมีคาบเวลาแบบไม่จำกัด แต่ในทางปฏิบัติจะใช้แบบมีจุดสิ้นสุดเพื่อให้ง่ายในการคำนวณโดยการใชัพังก์ชัน $w(n)$ ตรวจสอบส่วนต่าง ๆ ของ $S(n)$ โดยการเคลื่อนวินโดว์ตามรูป 5.1 วินโดว์ที่มีรูปร่างง่ายที่สุดคือรูปแบบสี่เหลี่ยม $r(n)$

$$w(n) = r(n) = 1 \quad \text{เมื่อ } 0 \leq n \leq N-1 \quad \dots (1)$$

$$= 0 \quad \text{อื่น ๆ}$$



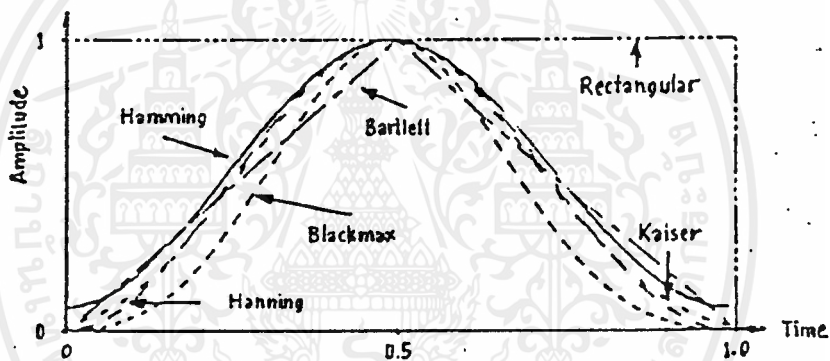
รูปที่ 5.1 แสดงสัญญาณเสียง $S(n)$ เมื่อวางทับสัญญาณวินโดว์ 3 ตัว โดยมีจุดเริ่มต้นของเวลาที่จุดสุ่ม $2N$, $3N$, $4N$

การวิเคราะห์แบบนี้เป็นการจำกัดช่วงจุดสุ่มให้เหลือ N โดยแต่ละจุดมีการถ่วงน้ำหนักเท่า ๆ กัน

วินโดว์ที่ใช้งานจริงนั้นต้องมีรูปร่างตามรูปที่ 2 สมมติว่า เสียงพูดประมาณว่าคงที่ในช่วง 10 ms เราต้องใช้วินโดว์ที่ช่วงเวลาเท่ากับ 20 ms โดยให้จุดตรงกลางถ่วงน้ำหนักมากกว่าจุดต้นและจุดปลาย เหตุผลที่ต้องถ่วงน้ำหนัก

จุดกลางมากกว่าจุดปลาย เพราะรูปร่างของวินโดว์มีผลต่อเอาท์พุท ต่อพารามิเตอร์ของเสียงพูด เมื่อเราเลื่อนวินโดว์ในโดเมนเวลาเพื่อวิเคราะห์เฟรมของสัญญาณเสียงพูด ผลของพารามิเตอร์อาจเปลี่ยนแปลงอย่างมากถ้าใช้ฟังก์ชัน $r(n)$ ตัวอย่างเช่น การหาพลังงานโดยการบวกผลกำลังสองของจุดลุ่มในวินโดว์ของสี่เหลี่ยมทำให้เกิดการกระเพื่อมรอบจุดความถี่ ดังนั้นเราจะใช้แฮนนิ่งวินโดว์ (Hanning window) แทนวินโดว์สี่เหลี่ยมที่มีรูปร่างเท่ากัน

$$w(n) = h(n) = 0.54 - 0.46 \cos(2\pi n / N - 1) \quad \text{เมื่อ } 0 \leq n \leq N-1 \\ = 0 \quad \text{เมื่ออื่น ๆ} \quad \dots \dots \dots (2)$$



รูปที่ 5.2 รูปแบบของวินโดว์ที่มีช่วงเวลาเท่ากับหนึ่ง

การทำให้ขอบวินโดว์ลาดลง ในการวิเคราะห์เฟรมโดยการเลื่อนตลอดแนวความยาวของสัญญาณทั้งหมด ไม่มีผลเสียต่อพารามิเตอร์เสียงพูด

5.2 พารามิเตอร์ในโดเมนเวลา (Time-domain parameter)

ในขบวนการของสัญญาณเสียงในฟังก์ชันเวลา จะมีความได้เปรียบในด้านความง่าย คำนวณเร็ว และ ง่ายในการแปลงความหมาย พารามิเตอร์ของเสียงสำหรับการเข้ารหัส และการจำ สามารถทำได้จากการวิเคราะห์ฟังก์ชัน

เวลา เช่น พลังงาน (amplitude), เสียง

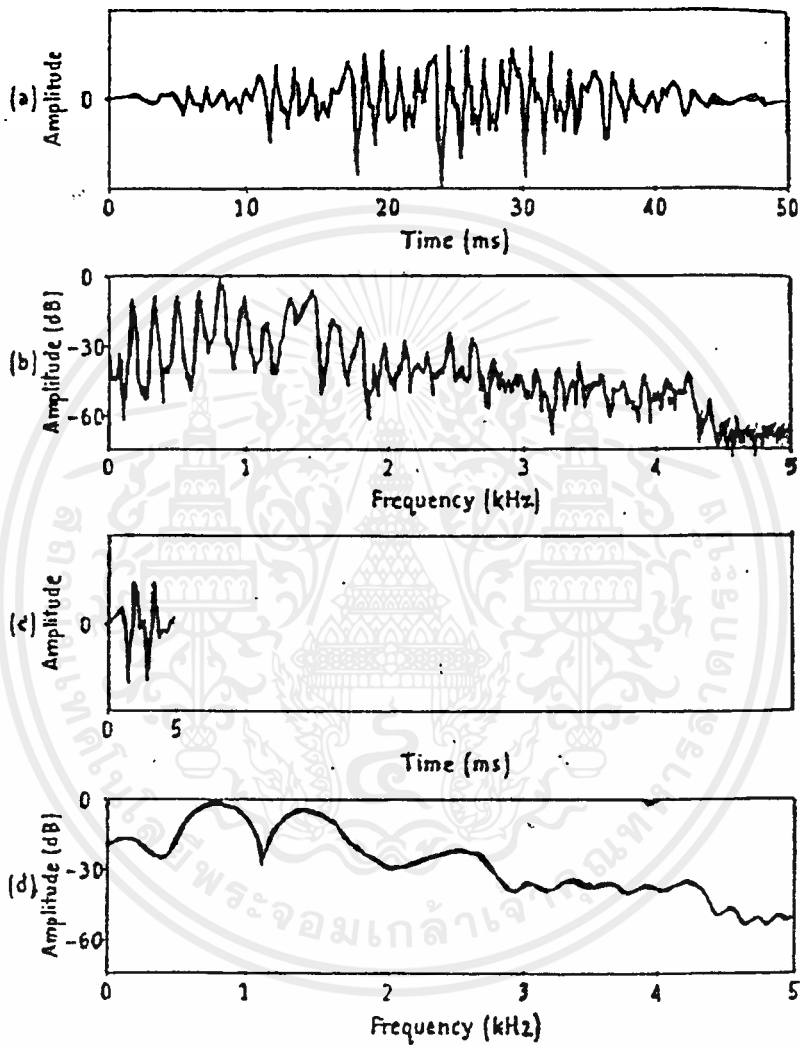
5.2.1 การวิเคราะห์ในโดเมนเวลา

การวิเคราะห์ฟังก์ชันเวลาที่เปลี่ยนจากสัญญาณเสียงสู่พารามิเตอร์หนึ่งตัวหรือมากกว่า ซึ่งปกติจะช้ากว่าสัญญาณเริ่มแรกมาก การเก็บและการ manipulate ในกรณีของสัญญาณเสียงจะมีประสิทธิภาพมากกว่าการเก็บสัญญาณเสียงโดยตรง ตัวอย่างเช่น คำพูดที่ถูกสุ่มมาที่ 6000-10000 ตัวอย่างต่อวินาที เพื่อที่จะเก็บช่วงความถี่ให้ถึง 3-5 kHz โดยปกติแล้วการแปลงเสียง 100 ms จะต้องการถึง 1000 ตัวอย่างเพื่อให้ได้ความถี่ที่ต้องการในการแสดงผลออกมา การสุ่มสัญญาณพารามิเตอร์โดยทั่วไปจะสุ่มที่ 40-100 ตัวอย่าง ต่อวินาที (มีบางระบบที่ต้องสุ่มถึง 200 ตัวอย่างต่อวินาที) ดังนั้นเมื่อเปลี่ยนคลื่นเสียงไปสู่กลุ่มพารามิเตอร์อัตราสุ่มสามารถลดลง

เทคนิคการประมวลผลแบบ Short-time ส่วนมากจะสร้างพารามิเตอร์อยู่ในรูปแบบ

$$Q(n) = \sum_m T[S(m)]w(n-m) \dots \dots \dots (3)$$

สัญญาณเสียง $S(n)$ ได้รับการเปลี่ยนเป็นฟังก์ชัน T ซึ่งถูกให้น้ำหนักโดยวินโดว์ $w(n)$ และถูกรวมกัน (Sum) เป็นพารามิเตอร์ $Q(n)$ ที่อัตราสุ่มเริ่มต้นซึ่งจะเป็นตัวแทนแสดงคุณสมบัติของคำพูด ที่ถูกเฉลี่ยภายใต้ช่วงของวินโดว์ $Q(n)$ ซึ่งได้มาจากการคอนโวลูชันของ $T[S(n)]$ กับ $w(n)$ เพื่อที่จะกระจายฟังก์ชันนั้น $w(n)$ ใช้เป็น low pass filter, $Q(n)$ จะถูกทำให้ลายเรียบลงไปในลักษณะของฟังก์ชัน $T[S(n)]$ เดิม เมื่อ $Q(n)$ เป็นเอาต์พุตของวงจรกรองความถี่ต่ำผ่าน low pass filter (วินโดว์) ในกรณีส่วนใหญ่แบนด์วิธของ $Q(n)$ กับของวงจรกรองความถี่ต่ำผ่าน $w(n)$ เพื่อการทำงานและการจัดเก็บมีประสิทธิภาพมากขึ้น



- รูปที่ 5.3 (a) แสดงสัญญาณที่ถูกคูณด้วยแฮนนิ่งวินโดว์ 50 ms
 (b) สเปกตรัมของรูป (a)
 (c) แสดงสัญญาณที่ถูกคูณด้วยแฮนนิ่งวินโดว์ 5 ms
 (d) สเปกตรัมของรูป (c)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ค่าพารามิเตอร์ในโดเมนความถี่

ค่าพารามิเตอร์สำคัญจำนวนมากถูกพบในโดเมนความถี่ เพราะท่อกำหนดเสียงให้กำเนิดสัญญาณ ที่สามารถวิเคราะห์เป็นความถี่ได้ง่ายกว่าทำในโดเมนของเวลาโดยตรง เสียงของคนพูดเพียงคนเดียวกันในแต่ละครั้ง เมื่อพิจารณาในโดเมนของเวลาจะเห็นผลของความแตกต่างของสัญญาณ แต่เมื่อพิจารณาผ่านในโดเมนความถี่สัญญาณที่ได้จะมีความคล้ายคลึงกันมาก

ระบบการได้ยินของมนุษย์จะตอบสนองต่อรูปร่างของสัญญาณเสียง (หรือขนาดที่กระจายอยู่ในโดเมนความถี่) มากกว่าที่เฟสของสัญญาณในโดเมนเวลา ด้วยเหตุผลเหล่านี้ การวิเคราะห์ความถี่จึงถูกใช้ในการดึงค่าพารามิเตอร์สำคัญจากเสียงพูด

ต่อไปจะได้กล่าวถึงวิธีการต่างที่จะนำไปใช้ในการวิเคราะห์ความถี่

5.3.1 การวิเคราะห์ด้วยฟิลเตอร์แบงก์ (filter bank)

การวิเคราะห์ด้วยฟิลเตอร์แบงก์ เป็นเทคนิคที่นิยมกันมาก เพราะสามารถตอบสนองได้ในเวลาจริง โครงสร้างง่ายไม่แพง โดยใช้ฟิลเตอร์แบงก์หรือกลุ่มของแบนด์พาสฟิลเตอร์ (bandpass filter) ซึ่งอาจจะเป็นอนาล็อกฟิลเตอร์หรือดิจิตอลฟิลเตอร์ก็ได้ แบนพาสฟิลเตอร์แต่ละตัวจะแยกวิเคราะห์ความถี่ของสัญญาณเสียงในช่วง ๆ ในทางการค้า วิธีฟิลเตอร์ มีความยืดหยุ่นมากกว่าการวิเคราะห์ด้วย (DFT) เพราะช่วงแบนวิธ (bandwidth) สามารถปรับช่วงตามขอบเขตการรับฟังเสียงของมนุษย์ได้มากกว่าการกำหนดตายตัวว่า ต้องมีแบนวิธกว้างแคบเท่านั้นเท่านั้น

ในการประยุกต์ใช้งานจำนวนมาก ต้องการกลุ่มของพารามิเตอร์ความถี่ไม่มากในการอธิบายการกระจายของพลังงานของความถี่ โดยทั่วไปใช้กลุ่มของแบนพาสฟิลเตอร์ 10-12 ตัว ก็สามารถให้ค่าตัวแทนของความถี่ที่ซับซ้อน และมีประสิทธิภาพมากกว่า การวิเคราะห์ด้วย DFT ซึ่งให้รายละเอียดปลีกย่อยของแต่ละความถี่ได้มากกว่า

ในวิธีการทั่วไป ฟิลเตอร์จะมีการวางระยะตาม bark scale คือให้มีระยะห่างความถี่ของฟิลเตอร์แต่ละตัวเท่า ๆ กัน และมีแบนวิธที่เพิ่มขึ้นแบบลอกการิทึม แต่ไม่ควรเกิน 1kHz โดยทั่วไปนิยมใช้ฟิลเตอร์ 1/3 ออกเตฟ (one-third-octave filter) ในระบบการจดจำเสียงพูดจริง ๆ นั้น อาจใช้ฟิลเตอร์แบ่งคี่ถึง 2 ครั้ง ครั้งแรกเป็นการจำแนกเสียงอย่างคร่าว ๆ ก่อน โดยใช้ฟิลเตอร์เพียงไม่กี่ตัว และ วิเคราะห์หารายละเอียดปลีกย่อยอีกครั้ง โดยใช้กลุ่มของฟิลเตอร์มากขึ้น

5.3.2 การวิเคราะห์ฟูรีเยร์ช่วงสั้น (Short-time Fourier Analysis)

การวิเคราะห์ฟูรีเยร์ ช่วงสั้นเป็นเทคนิคการหาความถี่ที่ใช้กันมานานแล้ว การวิเคราะห์ฟูรีเยร์ให้ตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชันความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งตลอดเวลา ดังนั้นจึงจำเป็นต้องใช้การวิเคราะห์ในช่วงสั้น โดยใช้วินโดว์

การใช้ฟูรีเยร์มีสมการว่า

$$S_{\omega}(e^{j\omega}) = \sum_{m} s(m) (e^{-j\omega m}) w(n-m) \dots \dots \dots (4)$$

ในการคำนวณเราต้องใช้ DFT แทนการแปลงฟูรีเยร์แบบต่อเนื่อง โดยใช้ฟังก์ชันวินโดว์ w ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว (N คือช่วงเวลาหรือขนาดของวินโดว์ที่ใช้ในการแปลง DFT) ข่าวดสารต่าง ๆ ใน $S_{\omega}(e^{j\omega})$ จะไม่สูญหายไปจากข้อมูลเดิม ถ้าการแปลงนั้นสุ่มมาด้วยความถี่สูงเพียงพอ (คือช่วงระยะห่างระหว่าง N) และ วินโดว์ $w(n)$ ไม่มีจุดสุ่มที่เป็น 0 ตลอดช่วง N ตัวแปร N เป็นตัวที่ต้องระวังมากเป็นพิเศษในการวิเคราะห์ความถี่ช่วงสั้น ถ้าค่าของ N ต่ำ (ใช้วินโดว์ช่วงสั้น) จะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะทำให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำในช่วงสั้น ๆ เท่านั้น ในทางตรงข้าม ถ้า N มีขนาดใหญ่จะให้ผลความละเอียดของเวลาที่ไม่มีดี แต่จะทำให้โดเมนความถี่มีความละเอียดสูงกว่า

บทที่ 6

การจดจำเสียงพูด

(Speech recognition)

การจดจำเสียงพูด เป็นการจดจำรูปแบบอย่างหนึ่ง เพื่อเป็นการเปรียบเทียบรูปแบบตรวจสอบ (test pattern) ซึ่งเป็นตัวแทนของสิ่งที่ไม่รู้ เพื่อแยกแยะ หรือจดจำกับรูปแบบอ้างอิง (reference pattern) ซึ่งได้รู้ลักษณะมาก่อนแล้ว

ในแต่ละรูปแบบจะอยู่ในรูปของเวกเตอร์ แต่ละสมาชิกของเวกเตอร์เป็นค่าที่วัดได้ของลักษณะเด่น (feature) ซึ่งเป็นลักษณะที่วัดได้ของสัญญาณอินพุท และสามารถนำไปใช้ประโยชน์ในการจดจำรูปแบบได้

ในการจดจำเสียงพูดแบบแยกคำ รูปแบบต่างๆจะเก็บอยู่ในรูปของกลุ่มฟังก์ชันเวลา เพราะค่าของแต่ละลักษณะที่วัดได้นั้นไม่ได้วัดที่จุดใดจุดหนึ่งของผู้พูดแต่มาจากการวัดทั้งคำ รูปแบบเหล่านี้เราเรียกว่า ต้นแบบ (template)

6.1 ประเภทของการจดจำเสียงพูด

การจดจำเสียงพูดโดยทั่วไปนิยมแบ่งออกเป็น ๔ กลุ่ม ดังนี้

1. การจดจำแบบแยกคำ (Isolated-word recognition) เป็นการจดจำโดยต้องมีช่วงหยุดของแต่ละคำ
2. การจดจำเฉพาะคำ (Word spotting) ตรวจจับคำเฉพาะใด ๆ ที่ปรากฏในประโยคนั้น
3. การจดจำแบบต่อเนื่อง (Connected speech recognition) จดจำคำพูดได้ทั้งประโยคโดยมิต้องมีช่วงหยุดแต่ละคำ
4. การเข้าใจเสียงพูด (Speech understanding) เป็นแบบสมบูรณ์ของกลุ่มที่ ๓ โดยสร้างแหล่งข้อมูลของการออกเสียงและหลักของภาษานั้น จุดประสงค์หลักก็คือ การพูดที่ไม่ต้องถูกต้องมากนัก แต่ให้เข้าใจความหมายของประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั้นได้ ในทางทฤษฎีระบบเข้าใจเสียงพูด จะตัดทอนเอาเฉพาะความหมายของประโยคโดยไม่คำนึงถึงเรื่องไวยากรณ์

การจำแบบแยกคำ (Isolated-word recognition)

การจำแบบแยกคำ เป็นจุดเริ่มต้นของการพัฒนาการจำเสียงพูด จนเป็นการจำเฉพาะคำและการจำแบบต่อเนื่องตามลำดับ ช่วงหยุดของแต่ละคำพูด ทำให้การจดจำคำง่ายขึ้น เพราะสามารถแยกแยะจุดสิ้นสุดของแต่ละคำ (หมายถึง จุดเริ่มต้นและจุดสิ้นสุดของคำนั้น) ทำให้ผลที่เกิดจากความไม่ชัดเจนของการพูดลดน้อยลงไปได้มาก การจำแบบแยกคำขณะที่ออกเสียงจะต้องระมัดระวังอย่างมาก เพราะต้องการช่วงหยุดระหว่างการพูดแต่ละคำ ทำให้การพูดขาดความไพเราะของภาษาไป ไม่เหมือนกับวิธีการอื่น ๆ ซึ่งสามารถพูดเป็นธรรมชาติได้มากกว่าและใช้ความระมัดระวังในการพูดน้อยกว่า

การจำแบบแยกคำ มีหลักการการจดจำโดยใช้ลักษณะของสัญญาณเสียงพูด ดังต่อไปนี้

1. ขนาดหรือกำลังต่อเวลา
2. อัตราตัดผ่านศูนย์ (zero-crossing rate)
3. สมดุลย์กลุ่มความถี่รวม (gross spectrum balance) เป็นการเปรียบเทียบพลังงานของกลุ่มความถี่สูงกับกลุ่มพลังงานความถี่ต่ำ
4. ความถี่สำคัญ อาจมาในรูปของ
 - ก. ความถี่จากการแปลง DFT
 - ข. F_1, F_2, F_3
 - ค. พารามิเตอร์ LPC หรือสัมประสิทธิ์ PARCOR
 - ง. เอาร์พุกของฟิลเตอร์แบงก์ (filter-bank) วิธีนี้เป็นวิธีที่นิยมกันที่สุดเพราะราคาถูกและความเร็วสูง

ขนาดนอกจากเป็นตัวบอกจุดสิ้นสุดของแต่ละคำ และยังเป็นตัวชี้บอกความแตกต่างของพยัญชนะและสระอีกด้วย

6.2 การตรวจจับจุดสิ้นสุด (Endpoint detection)

ปัญหาพื้นฐานของระบบจดจำเสียงพูด คือ เสียงที่เปล่งออกมาในขณะที่ตรวจสอบนั้น อาจจะไม่เหมือนกับตอนที่เรียนรู้เสมอไป เสียงที่เปล่งออกมา 2 ครั้ง จะมีระยะเวลาที่แตกต่างกัน และ ระยะระหว่างการออกเสียงที่ไม่แน่นอน หมายความว่า คุณสมบัติที่ขึ้นกับเวลา จะทำให้ การเปรียบเทียบไม่สามารถทำได้ เพราะคำอ้างอิงกับคำที่มาเปรียบเทียบเวลาจะต่างกัน ในกรณีเช่นนี้ ถึงแม้ว่าจะเปรียบเทียบกับคำที่ไม่รู้กับรูปแบบที่ถูกต้องของตัวเองแล้วก็ตาม แต่ก็อาจจะให้ผลแตกต่างได้พอๆกับที่เปรียบเทียบกับรูปแบบอ้างอิงของคำอื่น ซึ่งจะ เป็นปัญหา ถ้าคำที่ถูกเปรียบเทียบ เป็นการเปรียบเทียบทั้งคำต้นแบบ แทนที่จะเปรียบเทียบ บางส่วนของคำโดยการเปรียบเทียบส่วนต่อส่วน ความจริงนั้นแม้ในระหว่างเสียง สองเสียง ที่ใช้ในการเรียนรู้ รายละเอียดของเวลายังต่างกันเลย เมื่อเก็บ ข้อมูลเสียงจากหลาย ๆ คน จึงต้องทำการเฉลี่ยในระหว่างการเรียนรู้ก่อนนำไป เปรียบเทียบ

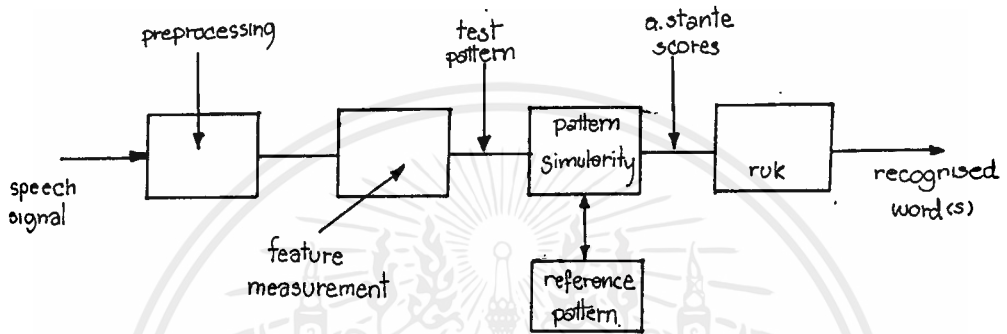
ในหลาย ๆ กรณี ความแม่นยำของการปรับ จะขึ้นอยู่กับความแม่นยำ ของการแสดงจุดสิ้นสุดของคำ ความผิดพลาดในการจดจำจุดเริ่มต้น และจุดสิ้นสุด ของคำ จะทำให้เกิดความยุ่งยากขึ้น โดยความผิดพลาดของจุดสิ้นสุด จะเกิดขึ้น ในคำที่เริ่มต้นและลงท้ายด้วยพยางค์ที่ออกเสียงเบา ผู้พูดบางคนอาจจบประโยคด้วย เสียงสั้น พยางค์เสียงสั้นนี้จะทำให้เกิดความผิดพลาดในการเปรียบเทียบคำ

ในห้องทดลอง การจดจำจุดสิ้นสุดอาจทำได้ง่าย เพราะ ข้อมูล เสียงพูดถูกรวบรวมภายใต้การควบคุมสภาวะ เพื่อหาจุดสิ้นสุด ยิ่งกว่านั้น ผู้พูดที่ บันทึกข้อมูลในห้องทดลอง จะทำด้วยความระมัดระวัง แต่นอกห้องทดลอง ไม่ได้ เป็นอย่างนั้น เสียงพูดจะมีการรบกวนจากรอบข้างเพราะไม่มีการควบคุมสภาวะ และผู้พูดไม่มีแรงจูงใจที่จะช่วยในการจดจำ

คุณสมบัติหลักของจุดสิ้นสุด คือ พลังงาน ดังนั้นวิธีง่ายๆ ที่ใช้ในการ ตรวจจับจุดสิ้นสุด ก็คือ การเปรียบเทียบพลังงานของสัญญาณกับค่าหรือระดับที่ตั้งไว้ โดยตรวจสอบจุดเริ่มต้นของคำจากตำแหน่งของคำที่มีพลังงานเริ่มมากกว่าระดับที่ตั้ง ไว้ และจุดสิ้นสุดจากตำแหน่งของคำที่มีพลังงานต่ำกว่าระดับที่ตั้งไว้ ค่าระดับที่ตั้ง

ไว้นี้ต้องมีความเข้าระดับหนึ่ง ถ้าความแรงของสัญญาณเสียงพูดอ่อน อาจจะต้องตั้งระดับต่ำ ๆ

ระบบจดจำเสียงพูดแบบแยกคำและไม่ขึ้นกับผู้พูด ด้วยเทคนิค DTW



รูปที่ 6.1

รูป 6.1 ได้แสดงแบบจำลองของการจดจำรูปแบบของสัญญาณเสียงพูด ซึ่งมีลักษณะการทำงาน ๓ อย่าง คือ การฝึก และการตรวจสอบ การฝึกเป็นการสอนศัพท์ให้กับระบบจดจำเสียงพูด โดยใช้ค่าเฉลี่ยของแต่ละคำ เป็นตัวแทนหรือต้นแบบอ้างอิง (reference template) ของคำนั้น การเฉลี่ยคำคำเดียวนั้นหลายครั้ง หลังจากการปรับแนวเวลา (time alignment) เป็นวิธีการอย่างหนึ่งเพื่อให้ได้ต้นแบบอ้างอิงตามที่ต้องการ

ในการตรวจสอบหรือการจดจำ เป็นการเปรียบเทียบคำที่ไม่รู้จักกับต้นแบบอ้างอิงที่เก็บไว้แล้ว เมื่อเปล่งเสียงออกมา คำพูดนั้นจะถูกบันทึกโดยผ่านตัวกรองความถี่ต่ำผ่าน (bandpass filter) ลดทอนสัญญาณความถี่ต่ำและจำกัดช่วงความถี่ เพื่อป้องกันความผิดพลาดของสัญญาณจากการรบกวนเชิงเลข (digitize) หลังจากนั้น คำที่เปล่งออกมาจะถูกแยกช่วงสัญญาณที่ไม่มีเสียงพูด (silence) ออกไปและมีการตรวจสอบจุดสิ้นสุดของแต่ละคำ เพื่อสะดวกต่อการประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนที่จะเก็บคำที่พูดออกมาทั้งหมด พารามิเตอร์เสียงพูด (speech parameter) ที่เป็นลักษณะเด่นของคำนั้นจะถูกแยกออกมา ในขั้นตอนนี้ต้องทำการลดขนาดของพารามิเตอร์ที่แยกได้ เพื่อทำการเปรียบเทียบกับรูปแบบอ้างอิง (reference -- pattern) หรือ รูปแบบที่ใกล้เคียงกับรูปแบบของคำที่ตรวจสอบมากที่สุด



บทที่ 7 ฮาร์ดแวร์

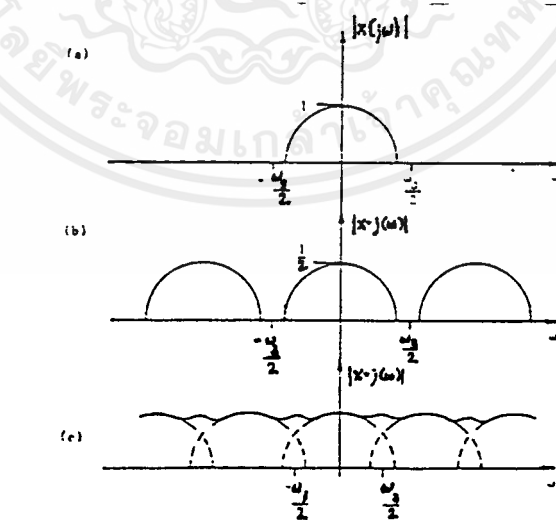
7.1 คุณสมบัติของฮาร์ดแวร์

- ความถี่ที่ใช้ในการส่งตัวอย่าง 10 KHz
- หน่วยความจำภายในฮาร์ดแวร์ 20 Kbyte
- เก็บข้อมูลเสียงได้ประมาณ 1 คำ หรือ 2 วินาที
- ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 8 bit
- ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอกขนาด 8 bit
- ตัวกรองผ่านความถี่ต่ำ สำหรับสัญญาณเข้า คัทออฟ 5 KHz
- แอมพลิไฟร์สำหรับขยายสัญญาณอินพุทและเอาต์พุท

7.2 แนวทางการออกแบบฮาร์ดแวร์

ฮาร์ดแวร์ที่ออกแบบจะนำมาใช้ในการเป็นส่วนเชื่อมโยง (interface) เครื่องคอมพิวเตอร์กับ เสียงพูด เพื่อให้สามารถนำข้อมูลของเสียงพูดเข้าไปใช้ในการคำนวณ และการวิเคราะห์โดยคอมพิวเตอร์ได้ การบันทึกข้อมูลเสียงพูดเข้าสู่คอมพิวเตอร์จะต้องแปลงเป็นสัญญาณเสียงพูดไปเป็นสัญญาณดิจิทัลแล้ว เก็บข้อมูลที่ได้ลงหน่วยความจำของฮาร์ดแวร์เมื่อหน่วยความจำบันทึกข้อมูลเต็มแล้ว จะถ่ายข้อมูลทั้งหมดเข้าสู่คอมพิวเตอร์เพื่อนำไปใช้ในการคำนวณ

หลักการออกแบบฮาร์ดแวร์ฮาร์ดแวร์ จะต้องสามารถแปลงข้อมูลเสียงพูดโดยแปลงจากสัญญาณอนาล็อก (analog signal) ไปเป็นสัญญาณดิจิทัล (digital signal) ใช้แซมปลิ่งเรทค่า 10KHz ซึ่งค่าที่ใช้มาจากทฤษฎีการสุ่มตัวอย่าง คือถ้าสัญญาณอนาล็อกที่ต้องการสุ่มตัวอย่างมี แถบความถี่ปฏิบัติการ (bandwidth) เท่ากับ f_0 แล้ว จะต้องใช้ความถี่ในการสุ่มตัวอย่าง (f) มากกว่าหรือเท่ากับ 2 เท่าของความถี่ f_0 เพื่อที่จะได้สัญญาณดิจิทัลโดยไม่สูญเสียเนื้อหาสำคัญ หรือ ข้อมูลสำคัญรวมทั้งป้องกันการเกิดผลการเอเลสซิ่ง (aliasing effect) ซึ่งเป็นการซ้อนทับของสัญญาณดิจิทัลที่ได้ ดังรูป 7.1



(a) Frequency spectrum of the continuous signal; (b) frequency spectra of the sampled signal; (c) aliasing of frequency spectra

รูป 7.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากความถี่เสียงพูดของคนทั่วไปมีค่าสูงสุดประมาณ 4-5 KHz ดังนั้นค่าความถี่ที่ใช้ในการส่งตัวอย่างควรจะมีค่าประมาณตั้งแต่ 8-10 KHz ขึ้นไป ซึ่งถ้าค่าความถี่ที่ใช้ในการส่งตัวอย่างมีค่าสูง ๆ ก็จะได้รูปสัญญาณที่มีคุณภาพดีขึ้นตาม แต่ก็จะมีข้อเสียคือจะต้องใช้เนื้อที่หน่วยความจำมากขึ้นตามไปด้วย รวมทั้งการคำนวณก็จะเสียเวลามากขึ้นเช่นกัน เพราะจำนวนข้อมูลที่ได้จากการส่งตัวอย่างมีจำนวนมากขึ้น ในขณะที่สัญญาณยังมีขนาด (ในฐานเวลา) เท่าเดิม ดังนั้นเราจึงต้องพิจารณาเลือกความถี่ในการส่งตัวอย่าง ที่สามารถส่งตัวอย่างสัญญาณให้ได้รายละเอียดเพียงพอต่อการคำนวณ แต่ไม่มากจนเกินไป เช่น การส่งตัวอย่างข้อมูลจำนวน 10 ครั้งต่อหนึ่งคาบ อาจจะได้ข้อมูลที่ได้รายละเอียดเพียงพอ ต่อการคำนวณไม่จำเป็นต้องส่งตัวอย่างข้อมูลถึงจำนวน 100 ครั้งต่อคาบ เป็นต้น

การเลือกขนาดหน่วยความจำ จะต้องคำนึงว่าจะสามารถมีขนาดพอเพียงที่จะเก็บจำนวนข้อมูลที่ต้องการใช้ เช่นในการบันทึกข้อมูลเสียงพูด 1 คำ โดยที่ปกติการพูด 1 คำพูดนี้จะใช้เวลาไม่เกิน 1 วินาที การออกแบบขนาดหน่วยความจำจึงต้องพิจารณาให้สามารถบันทึกข้อมูลเสียงพูดได้ทั้งหมดคือมากกว่า 1 วินาทีขึ้นไป แต่ก็ต้องพิจารณาเนื้อที่ที่ใช้ในการเก็บข้อมูลด้วย ถ้าใช้หน่วยความจำมากจนเกินไปก็จะเปลืองเนื้อที่ในการเก็บข้อมูล

โครงการนี้วิเคราะห์ข้อมูลเป็นคำ ๆ ดังนั้นฮาร์ดแวร์นี้จึงเลือกใช้หน่วยความจำขนาด 20 Kbyte และใช้ความถี่ในการส่งตัวอย่างที่ 10 KHz จะทำให้สามารถบันทึกข้อมูลเสียงพูดได้นาน 2 วินาที ซึ่งเพียงพอต่อการเก็บข้อมูลเนื่องจากเสียงคนพูดปกติจะใช้เวลา 1 วินาทีสำหรับคำหนึ่งคำ

เนื่องจากอุปกรณ์ฮาร์ดแวร์ที่ออกแบบไว้สำหรับใช้งานต่อเชื่อมกับเครื่องคอมพิวเตอร์โดยทำเป็นการ์ดที่ใช้ติดตั้งผ่านพอร์ตต่อขยาย (expansion slot) ของคอมพิวเตอร์ ดังนั้นการออกแบบวงจรและอุปกรณ์จะต้องคำนึงถึงด้านต่าง ๆ คือ

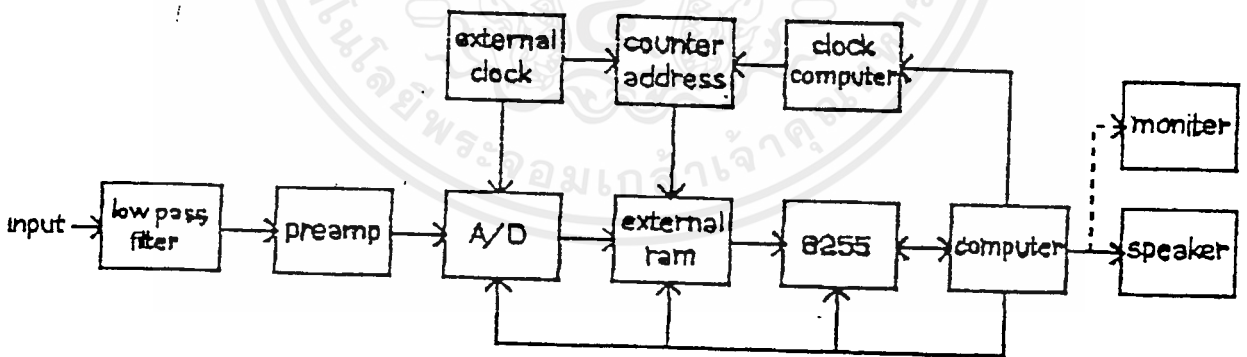
1. จะต้องป้องกันไม่ไห้ ฮาร์ดแวร์นี้ ไปรบกวนการทำงานของระบบคอมพิวเตอร์รวมทั้งการป้องกันไม่เครื่องคอมพิวเตอร์มารบกวนระบบของฮาร์ดแวร์ซึ่งส่วนนี้จะใช้การดีโค๊ดแอดเดรสของฮาร์ดแวร์ทางพอร์ตอินพุทเอาต์พุท (I/O port) เพื่อเอนเอเบิล(enable) หรือดิสเอเบิล(disable) การติดต่อใช้งานกับฮาร์ดแวร์

เมื่อคอมพิวเตอร์ต้องการเรียกใช้งานก็จะอ้างอิงถึงค่าแอดเดรสของฮาร์ดแวร์นี้จึงจะใช้งานได้เท่านั้น

2. จะต้องออกแบบวงจรอย่างรัดกุม และ รอบคอบ เพราะความผิดพลาดของฮาร์ดแวร์จะทำให้มีผลกระทบต่อเครื่องคอมพิวเตอร์ถึงขั้นเสียหายได้ เช่น การลัดวงจรของฮาร์ดแวร์ การส่งสัญญาณที่ไม่ถูกต้องเข้าสู่เครื่องคอมพิวเตอร์ เป็นต้น จึงควรป้องกันไว้ โดยการใส่บัฟเฟอร์ให้กับเส้นสัญญาณต่าง ๆ ที่เชื่อมต่อระหว่างคอมพิวเตอร์กับฮาร์ดแวร์ เมื่อเกิดการผิดพลาดขึ้นก็จะมีส่วนของบัฟเฟอร์กันเอาไว้ก่อน

3. เนื่องจากวงจรมีทั้งส่วนที่เป็นวงจรทางอนาลอก และส่วนที่เป็นวงจรทางดิจิทัล การออกแบบลายวงจรให้แยกกราวด์ของสัญญาณอนาลอก กับ กราวด์ของสัญญาณดิจิทัลออกจากกัน เพื่อป้องกันปัญหาของสัญญาณรบกวน (noise) ซึ่งวิธีนี้จะช่วยลดสัญญาณรบกวนลงได้อย่างมาก

7.3 โครงสร้างของฮาร์ดแวร์



รูปที่ 7.2 บล็อกไดอะแกรมแสดงโครงสร้างของฮาร์ดแวร์

โครงสร้างของฮาร์ดแวร์ทั้งหมดสามารถแสดงได้ดังภาพที่ 2

เมื่อเริ่มพูด สัญญาณจากไมโครโฟนก็จะผ่านสู่ตัวกรองผ่านความถี่ต่ำ (low pass filter) จากนั้นนำไปขยายสัญญาณให้ใหญ่ขึ้นที่ภาคปริแอมพลิไฟร์ เพื่อให้เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น มีอยู่ภายใต้ลิขสิทธิ์ของสถาบันฯ เพื่อไม่ให้เอกสารนี้ถูกเผยแพร่โดยไม่ได้รับอนุญาต หากท่านมีข้อสงสัยหรือต้องการข้อมูลเพิ่มเติม กรุณาติดต่อฝ่ายประชาสัมพันธ์ โทร. 0-2942-3000 หรือ 0-2942-3001 ในวันและเวลาราชการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณมีขนาดพอเหมาะที่จะนำเข้าสู่ตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล

สัญญาณนาฬิกาภายในฮาร์ดแวร์ จะทำหน้าที่ผลิตสัญญาณความถี่ ที่ใช้ในการสุ่มตัวอย่าง เพื่อป้องกันกับตัวแปลงสัญญาณอนาลอก เป็นดิจิตอลและตัวนับแอดเดรส

ตัวนับ (counter) แอดเดรสจะทำหน้าที่อ้างอิงแอดเดรสของหน่วยความจำตามจังหวะสัญญาณความถี่นาฬิกาจากสัญญาณนาฬิกาภายในฮาร์ดแวร์หรือ สัญญาณนาฬิกาจากคอมพิวเตอร์อย่างใดอย่างหนึ่ง ถ้าอยู่ในช่วงแปลงข้อมูลจากอนาลอกเป็นดิจิตอลแล้ว เขียนลงหน่วยความจำก็จะใช้สัญญาณนาฬิกาภายในฮาร์ดแวร์ แต่ถ้าเป็นการอ่านข้อมูลจากหน่วยความจำเข้าสู่คอมพิวเตอร์ ก็จะขึ้นกับสัญญาณนาฬิกาจากคอมพิวเตอร์ ทั้งนี้ เพื่อให้จังหวะการทำงานของการทำงานเขียนข้อมูลลงหน่วยความจำตรงกับอัตราการสุ่มข้อมูลและ จังหวะการอ่านข้อมูลจากหน่วยความจำตรงกับการเรียกข้อมูล (import) เข้าสู่คอมพิวเตอร์

ข้อมูลที่อ่านเข้าสู่คอมพิวเตอร์จะส่งผ่าน 8255 ซึ่ง 8255 ทำหน้าที่เป็นพอร์ตขนาน อินพุทเอาต์พุทสำหรับระบบสามารถโปรแกรมได้ 3 พอร์ต สัญญาณทั้งหมดจากคอมพิวเตอร์จะส่งผ่าน 8255 ตัวนี้

สำหรับคอมพิวเตอร์ จะเป็นตัวควบคุมการทำงานของฮาร์ดแวร์เกือบทั้งหมด ได้แก่ ควบคุมตัวแปลงสัญญาณอนาลอกเป็นดิจิตอล ควบคุมการอ่านและเขียนหน่วยความจำควบคุมการส่งข้อมูลอินพุทเอาต์พุทผ่าน 8255 และควบคุมการส่งข้อมูลเอาต์พุทออกตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก

สำหรับส่วนเอาต์พุทของระบบ ข้อมูลจะส่งมาที่ตัวแปลงสัญญาณดิจิตอลเป็นอนาลอก และ นำไปขยายผ่านแอมพลิไฟร์ เพื่อนำไปออกลำโพงอีกที

7.4 การออกแบบฮาร์ดแวร์ส่วนต่าง ๆ

7.4.1 การเชื่อมต่อกับคอมพิวเตอร์ เนื่องจากฮาร์ดแวร์เป็นอุปกรณ์ประเภทอินพุทเอาต์พุท จะต้องเชื่อมต่อกับคอมพิวเตอร์ผ่านทาง อินพุทเอาต์พุทพอร์ต (I/O port) ซึ่งการใช้งานจะต้องกำหนดแอดเดรสขึ้นมาเพื่อที่จะสามารถอ้างอิงถึงได้โดย

โปรแกรมแต่สำหรับแอดเดรสค่าต่าง ๆ จะมีกำหนดไว้ให้อุปกรณ์อื่นใช้งานอยู่แล้วตั้งไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 7.1

สำหรับการเลือกใช้อัดเตอร์สำหรับฮาร์ดแวร์นี้ จะเลือกใช้อัดเตอร์ที่อยู่ในช่วง 300-31F ซึ่งเป็นอัดเตอร์ที่เอาไว้สำหรับการดัดแบบ (prototype card) จึงสามารถใช้อัดเตอร์ในช่วงนี้ได้ แต่ต้องระวังมิให้มีการอ้างอัดเตอร์ซ้ำซ้อนในกรณีที่มีการดัดแบบมากกว่าหนึ่งการ์ดขึ้นไป

อัดเตอร์ที่ใช้งานมีค่าคือ 31C 31D 31E และ 31F จำนวน 4 อัดเตอร์เพื่อใช้งาน โดยเรียกผ่าน 8255 อีกที

อัดเตอร์ 31C ใช้เป็นพอร์ตอินพุต สำหรับรับข้อมูลเสียงที่แซมปลิ่งเข้ามา ซึ่งตรงกับพอร์ต A ของ 8255

อัดเตอร์ 31D ใช้เป็นพอร์ตเอาต์พุตสำหรับส่งข้อมูลเสียงไปแสดงผลทางลำโพงซึ่งตรงกับพอร์ต B ของ 8255

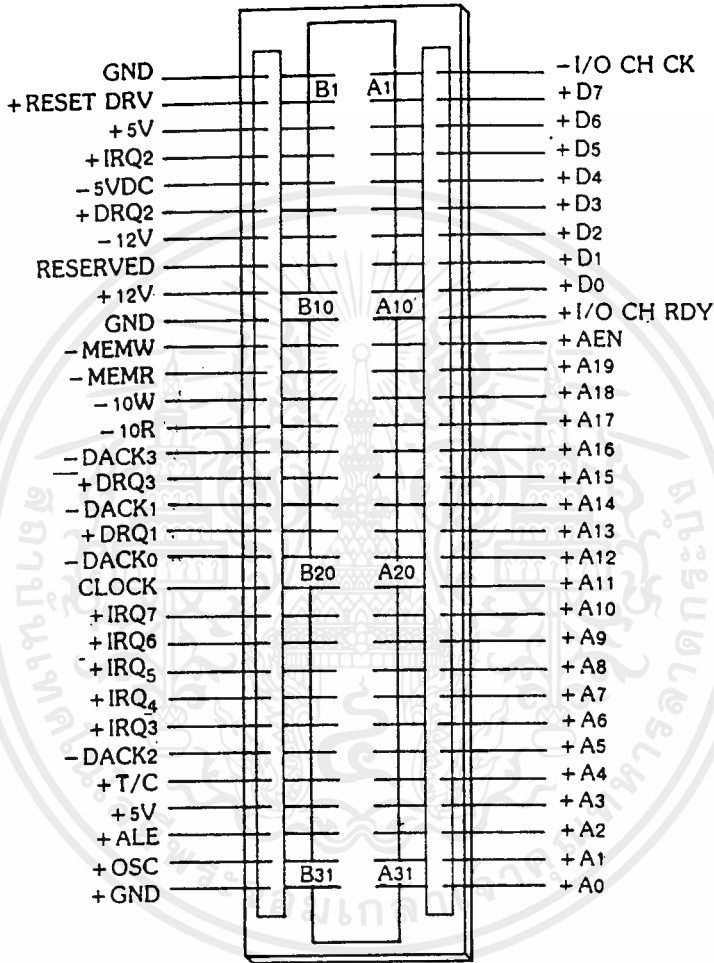
อัดเตอร์ 31E ใช้เป็นพอร์ตควบคุม คอมพิวเตอร์จะส่งและรับสัญญาณควบคุมต่าง ๆ ทางพอร์ตนี้ ได้แก่ สัญญาณอ่าน/เขียนข้อมูล, สัญญาณนาฬิกาจากคอมพิวเตอร์, สัญญาณเคลียร์เคาเตอร์, สัญญาณสิ้นสุดการอ่าน/เขียนหน่วยความจำ ซึ่งตรงกับพอร์ต C ของ 8255 โดยพอร์ต C บน เป็นเอาต์พุตพอร์ตที่ส่งสัญญาณควบคุมออกจากคอมพิวเตอร์และพอร์ต C ล่าง เป็นอินพุตพอร์ตที่รับสัญญาณสิ้นสุดการอ่าน/เขียนหน่วยความจำเข้าคอมพิวเตอร์

อัดเตอร์ 31F ใช้ส่งสัญญาณเพื่อที่จะโปรแกรมการใช้งาน 8255

Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Registers
0AX	NMI Mask Registers
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Sec)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C	SDLC Communications
380-389	Binary Synchronous Communications (Sec)
3A0-3A9	Binary Synchronous Communications (Sec)
3B0-3BF	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communications (Pri)

ตารางที่ 7.1 การใช้งานแอดเดรสสำหรับพอร์ตอินพุทเอาต์พุตต่าง ๆ

7.4.2 สัญญาณต่าง ๆ บนสลีตคอมพิวเตอร์



รูปที่ 7.3 แสดงขาสัญญาณต่าง ๆ บนสลีตคอมพิวเตอร์

เส้นสัญญาณต่าง ๆ ประกอบด้วย เส้นสัญญาณของบัสแอดเดรส (address bus), บัสข้อมูล (data bus), บัสควบคุมสำหรับการเขียน/อ่านข้อมูลจากหน่วยความจำ หรือ พอร์ตอินพุทเอาต์พุท, เส้นสัญญาณสำหรับการขออินเทอร์รัพท์ของวง

เอกสารนี้จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น เมื่ออนุญาตให้นำไปเผยแพร่โดยไม่เสียค่าใช้จ่าย

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

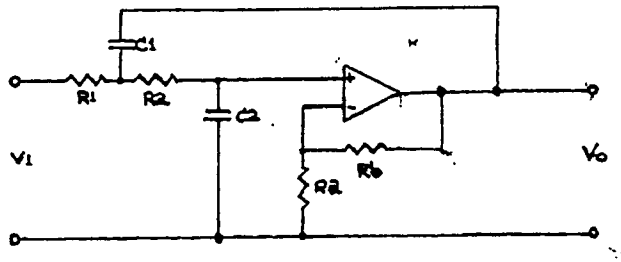
signal) ต่างๆ ที่ใช้ในระบบ, เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHECK)

นอกจากเส้นสัญญาณเหล่านี้ สล็อตบนเมนบอร์ดยังต่อกับแหล่งจ่ายไฟที่ใช้ในระบบอีก คือ +5Vdc, -5Vdc, +12Vdc และ -12Vdc

7.4.3 ตัวกรองผ่านความถี่ต่ำ (low pass filter) จะเป็นตัวกรองสัญญาณเสียงอินพุทโดยพิจารณาจากค่าความถี่คัทออฟ (cut off frequency) สัญญาณที่มีความถี่ต่ำกว่าค่าความถี่คัทออฟนี้จะสามารถผ่านได้โดยไม่ถูกลดทอน (attenuate) แต่สำหรับสัญญาณที่มีความถี่สูงกว่าค่าความถี่คัทออฟ จะถูกลดทอนลงทำให้ไม่สามารถผ่านไปได้ สำหรับค่าความถี่คัทออฟจะขึ้นอยู่กับแถบความถี่ปฏิบัติงาน (bandwidth) ของสัญญาณที่ใช้กันว่า เท่ากับเท่าไร ก็จะกำหนดค่าความถี่คัทออฟให้กรองสัญญาณที่อยู่นอกเหนือจากช่วงนั้นออกไป

ตัวกรองความถี่นี้มีประโยชน์ทั้งในด้านการป้องกันไม่ให้สัญญาณอื่น ๆ ที่ไม่ต้องการเข้ามารบกวนโดยการลดทอนสัญญาณเหล่านั้นไป รวมทั้งสัญญาณรบกวน (noise) ก็จะถูกกรองไปด้วย และในวงจรที่มีการแปลงสัญญาณจากอนาลอกไปเป็นดิจิทัล (analog to digital converter) การใช้ตัวกรองผ่านความถี่ต่ำจะสามารถช่วยป้องกันการเกิดผลการเอเลซซิ่ง (aliasing effects) อันเกิดจากความถี่ของสัญญาณที่ไม่ต้องการมีความถี่สูงกว่าความถี่ของสัญญาณที่ต้องการทำให้ข้อมูลที่ได้จากการสุ่มตัวอย่างข้อมูลมีค่าผิดเพี้ยนไปจากความเป็นจริง

ในที่นี้ใช้วงจรกรองผ่านความถี่ต่ำอันดับที่ 2 (second order) แบบบัตเตอร์เวิร์ท (Butterworth, maximally flat) ซึ่งจะ approximate ideal filter ดีกว่าแบบอื่นในช่วงแถบความถี่ผ่าน (pass band) จากวงจรจะเป็น low gain amplifier filter ซึ่งรูปแบบจะเป็นออปแอมป์ที่มีความต้านทานย้อนกลับภายนอก และความต้านทานที่อินพุทต่อกับขา noninverting input ของออปแอมป์ วงจรที่ใช้มีเกนขยายเป็น 1 ($x=1$) คือไม่ขยายสัญญาณอีก จะกรองสัญญาณเท่านั้น จึงต้องแทนความต้านทาน R_u ด้วยวงจรเปิด (open circuit) และแทนความต้านทาน R_b ด้วยวงจรปิด (short circuit) ตามรูป 7.4



รูป 7.4 วงจรตัวกรองผ่านความถี่ต่ำอันดับสอง

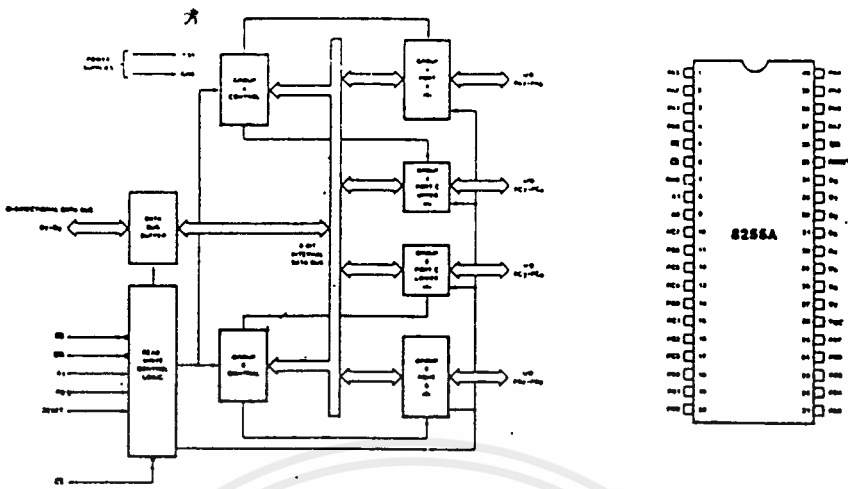
ในการกำหนดค่าของอุปกรณ์สำหรับวงจรตัวกรองผ่านความถี่ต่ำผ่านชนิดนี้ หาได้โดยเทียบค่าจากตารางที่ 7.2

Filter Class	R1	R2	C1	C2	K
Maximally flat	1.00000	1.00000	0.87403	1.14412	2.00000
(Butterworth)	1.00000	1.00000	1.41421	0.70711	1.00000
3.01 dB at w_H	1.00000	1.00000	1.00000	1.00000	1.58578
Equal ripple	1.00000	1.00000	0.77088	0.85555	2.00000
(Chebishev)	1.00000	1.00000	1.40259	0.47013	1.00000
0.5 dB ripple	0.81220	0.81220	1.00000	1.00000	1.84213
Equal ripple	1.00000	1.00000	0.93809	0.96688	2.00000
(Chebushev)	1.00000	1.00000	1.82192	0.49783	1.00000
1 dB ripple	0.95237	0.95237	1.00000	1.00000	1.95446

ตารางที่ 7.2 Component and gain for the low-pass filter of

figure 5.2 for $w_H=1$ (R in ohms, C in farads)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 7.5 แผนผังและการจัดขาของ 8255

สัญญาณต่าง ๆ ของ 8255

DO-D7 เป็นขาข้อมูล (data bus) แบบ 2 ทิศทาง

A0-A1 ใช้งานร่วมกับสัญญาณ RD และ WR เพื่อเลือกและควบคุม 1 ใน 3 พอร์ต หรือคอนโทรลเวิร์ดรีจิสเตอร์ (control word register)

PA, PB, PC พอร์ตทั้ง 3 ของ 8255

CS (chip select) สัญญาณเลือกชิปเพื่อใช้งาน

RD (read) เป็นสัญญาณการอ่านข้อมูลภายใน 8255

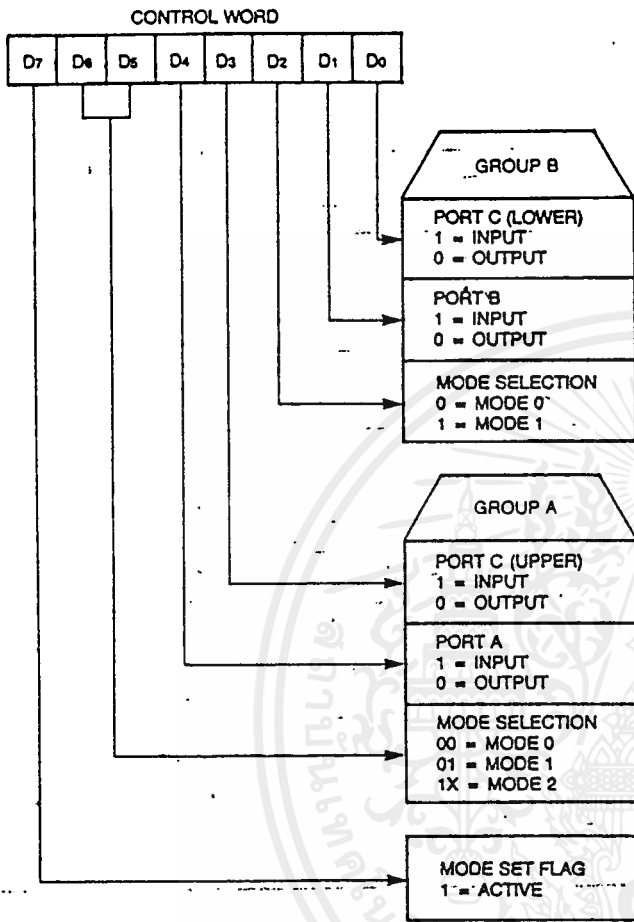
WR (write) เป็นสัญญาณในการเขียนข้อมูลลงใน 8255

RESET เป็นสัญญาณรีเซ็ตแก่ 8255 เพื่อเคลียร์รีจิสเตอร์

เราควบคุมการทำงานภายในของ 8255 โดยส่งสัญญาณไปโปรแกรมพอร์ตควบคุม (control port) ซึ่งจะต้องมีการโปรแกรมในตอนแรก ก่อนการใช้งาน การโปรแกรม 8255 แบ่งออกเป็น 3 โหมด คือ โหมด 0 โหมด 1 และ โหมด 2 สำหรับการใช้งานฮาร์ดแวร์จะใช้โหมด 0 ลักษณะความหมายของแต่ละบิตข้อมูลที่ใช้ในการโปรแกรมพอร์ตควบคุมจะแสดงได้ดังรูปที่ 7.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความหมายของบิตต่างๆมีดังนี้



บิต D7 เป็นบิตที่แสดงรหัสควบคุม ถ้าบิตนี้เป็น "1" หมายถึง รหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเชื่อมต่อต่างๆ ของ 82

บิต D6 และ D5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมี 3 โหมด คือ โหมด 0, 1 และ 2

บิต D4 ถ้ามีค่าเป็น "0" หมายถึง พอร์ต A เป็นพอร์ตเอาต์พุต ถ้ามีค่าเป็น "1" หมายถึงกำหนดพอร์ต A เป็นพอร์ตอินพุต

บิต D3 เป็นบิตที่บอกถึงการเชื่อมต่อพอร์ต C บน ถ้าเป็น "0" จะทำให้เป็นเอาต์พุต

รูปที่ 7.6 ความหมายของรหัสควบคุม

บิต D2 เป็นบิตที่เป็นการเชื่อมต่อของพอร์ต B ถ้าเป็น "0" หมายถึง พอร์ต B เป็นโหมด 0 แต่ถ้าเป็น "1" หมายถึง การเลือกโหมด 1

บิต D1 เป็นการกำหนดการเลือกอินพุตเอาต์พุตของพอร์ต B ถ้าเป็น "0" หมายถึง เอาต์พุต ถ้าเป็น "1" หมายถึง อินพุต

7.3 อธิบายส่วนต่าง ๆ ของวงจร

จากรูปวงจร สัญญาณแอดเดรสส์จากคอมพิวเตอร์ จะมาเข้า 74LS20 และ 74LS138 เพื่อทำการตีโค้ดการเรียกใช้งานของฮาร์ดแวร์ ให้สามารถติดต่อกับงานได้ในช่วงแอดเดรส 31C-31F สำหรับสัญญาณจากตาต้าบัส จะผ่านบัฟเฟอร์ 74LS245 เข้าสู่ 8255 เพื่อแยกเป็น 3 พอร์ตไว้ติดต่อกับอินพุทเอาต์พุทกับส่วนต่าง ๆ ของฮาร์ดแวร์

พอร์ต A เป็นพอร์ตอินพุท อินพุทสัญญาณจากการสุ่มตัวอย่าง เข้าสู่คอมพิวเตอร์

พอร์ต B เป็นพอร์ตเอาต์พุท สำหรับส่งข้อมูลจากคอมพิวเตอร์ ผ่านวงจรแปลงสัญญาณดิจิทัลเป็นอนาลอก แล้วขยายสัญญาณออกลำโพง

พอร์ต C บน เป็นเอาต์พุทพอร์ต จะส่งสัญญาณจากคอมพิวเตอร์มาควบคุมส่วนต่าง ๆ ของฮาร์ดแวร์ 3 สัญญาณ คือ

- สัญญาณ CLEAR จะใช้สำหรับการรีเซ็ตเคาท์เตอร์แอดเดรสเพื่อเริ่มต้นนับแอดเดรสหน่วยความจำใหม่ในการอ่านและเขียนข้อมูลแต่ละครั้ง
- สัญญาณ WR/RD เป็นสัญญาณที่แสดงถึงช่วงเวลาการเขียนหรืออ่าน การเขียน คือการเขียนข้อมูลที่ได้จากการสุ่มตัวอย่างลงหน่วยความจำ การอ่านคือการอ่านข้อมูลที่เก็บในหน่วยความจำเข้าคอมพิวเตอร์
- สัญญาณ CL RD เป็นสัญญาณนาฬิกาจากคอมพิวเตอร์ที่ใช้ในการอ้างแอดเดรสหน่วยความจำในการอ่านข้อมูล

พอร์ต C ล่าง เป็นอินพุทพอร์ต จะรับสัญญาณ END จากฮาร์ดแวร์ สัญญาณ END เป็นสัญญาณที่บอกถึงการสิ้นสุดการอ่าน/เขียนหน่วยความจำ เพื่อให้คอมพิวเตอร์ทำงานอื่นต่อไป

74LS125 ทั้ง 3 ตัวทำหน้าที่คล้ายสวิทช์ในการเลือกสัญญาณนาฬิกาที่ใช้ในการอ่าน หรือ เขียนข้อมูล โดยมีสัญญาณ WR/RD เป็นสัญญาณอ้างอิง สัญญาณนาฬิกาอ้างอิงการอ่านหรือเขียนข้อมูล ผ่านทางไบนารีเคาท์เตอร์ 74LS393 จากรูปเอกสารจะใช้ 74LS393 จำนวน 4 ตัว เพื่อนับให้ได้ถึง 15 บิต หรือ สามารถอ้างอิงหน่วยไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำจำนวน 20 Kb ได้ เมื่อเคอร์เตอร์นับถึง 20 Kbyte แล้วจะผ่าน 74LS00 และ 74LS04 เพื่อไปเป็น สัญญาณ END และไปใช้ในการเอนเอเบิล 74LS125 ด้วย

หน่วยความจำจะใช้ไอซีเบอร์ 62256 ซึ่งเป็น หน่วยความจำขนาด 32K x 8 bit คือ มีเนื้อที่หน่วยความจำ 32K ตำแหน่ง ตำแหน่งละ 8 บิต แต่ที่ใช้ในวงจรนี้จะอ้างถึงแค่จำนวน 20 K เท่านั้น

ADC 0808 เป็น ไอซีที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิตอล (Analog to digital converter) ขนาด 8 บิต อัตราสุ่มสัญญาณสูงสุด 10 KHz ซึ่งตั้งค่าโวลต์เดจ อ้างอิงที่ 0 - 5 โวลต์ ขาสัญญาณ START กับ EOC ต่อเข้าด้วยกันเพื่อใช้ในโหมด free running สัญญาณนาฬิกาที่ป้อนให้กับ ADC 0808 ใช้สัญญาณนาฬิกาที่ความถี่ 1.25 MHz แล้วจะได้ conversion times เท่ากับ 0.1 มิลลิเซค หรือ อัตราการสุ่มสัญญาณเป็น 10 KHz สัญญาณนาฬิกา 1.25 MHz ได้มาจากจากคริสตอลโมดูลสัญญาณนาฬิกา 10 MHz นำมาหารความถี่ โดย 74LS390 3 ตัว ก็จะได้สัญญาณนาฬิกา 1.25 MHz

สัญญาณอินพุทจะผ่าน LM 382 ซึ่งจะขยายสัญญาณจากไมโครโฟน แล้วยกระดับขึ้น 2.5 โวลต์แล้วผ่าน LF356 ที่ทำเป็นตัวกรองผ่านความถี่ต่ำที่ตั้งความถี่คัทออฟไว้ที่ 5 KHz จากนั้นก็นำสัญญาณป้อนเข้า ADC 0808

สัญญาณเอาต์พุทจากพอร์ต B จะต้องผ่านวงจรแปลงสัญญาณจากดิจิตอลเป็นอนาลอก (digital to analog converter) ก่อน ซึ่งในที่นี้จะใช้ไอซี DAC 0808 ที่มีขนาด 8 บิต แปลงสัญญาณดิจิตอลให้เป็นอนาลอก แล้วนำไปขยายสัญญาณโดย TL 082 แล้วจึงออกสู่ลำโพง

บทที่ 8

สรุปการทดลอง

1) การวิเคราะห์สัญญาณเสียงพูด ควรใช้กับเครื่องที่มีประสิทธิภาพ ยิ่งถ้าเป็นการใช้เทคนิคทางดิจิทัลแล้ว ควรที่จะต้องใช้เครื่องที่มีความเร็วสูง หรือใช้ชิพประมวลผลทางคณิตศาสตร์ (Math coprocessor) จึงจะสามารถทำการวิเคราะห์ได้อย่างรวดเร็ว

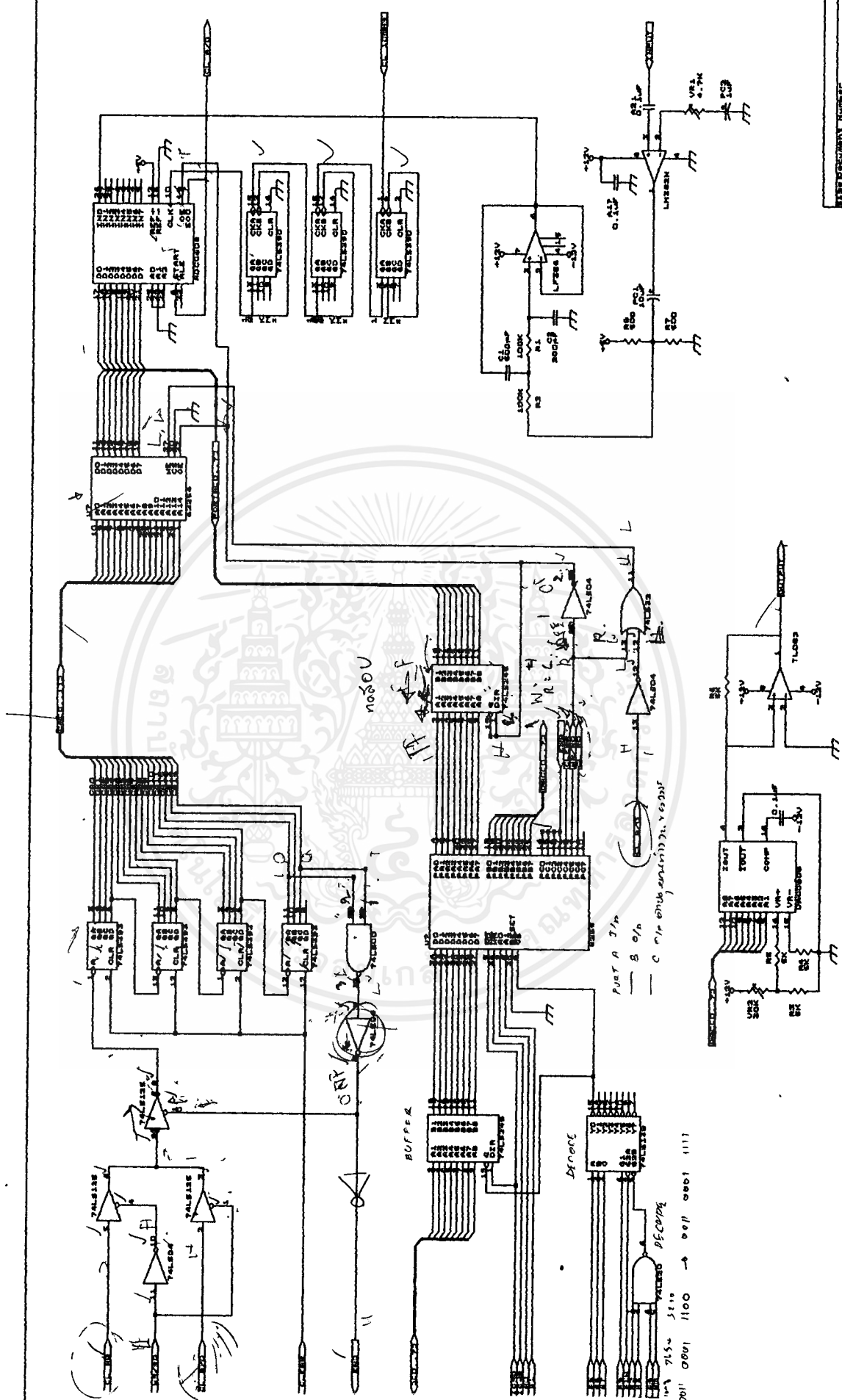
ในการทดลองต้องทำการ run กับเครื่องคอมพิวเตอร์ที่มีความเร็วต่ำ และไม่ได้ใช้ตัวประมวลผลร่วมทางคณิตศาสตร์ ทำให้ผลตอบสนองไม่เป็นเวลาจริง การวิเคราะห์ทำได้ลำบาก ซึ่งเราจะสามารถแก้ปัญหานี้ได้โดยการใช้โปรเซสเซอร์ที่ออกแบบสำหรับการประมวลผลสัญญาณโดยเฉพาะ เช่น TMS320C25 หรือเปลี่ยนไปใช้วงจรทางอนาล็อก โดยใช้เทคนิค filter bank แบ่งเป็นกลุ่มของแบนพาสฟิลเตอร์ หรือ แบ่งช่วงของความถี่ออกเป็นช่วงย่อย ซึ่งอาจฟิลเตอร์แบ่งเป็น 2 ชุด ชุดแรกเป็นการแบ่งกลุ่มของเสียงอย่างคร่าว ๆ ก่อน แล้วจึงซอยเป็นความถี่ย่อย ๆ ต่อไป

2) FFT เป็นเครื่องวิเคราะห์สัญญาณในโดเมนเวลาโดยเปลี่ยนให้เป็นสเปกตรัมในโดเมนความถี่ ที่ทำงานเร็วกว่าและใช้เนื้อที่หน่วยความจำ ที่ต้องใช้ในการคำนวณน้อยกว่าการคำนวณด้วย DFT โดยตรง

3) เมื่อใช้วินโดว์สี่เหลี่ยมวิเคราะห์ในช่วงที่ต้องการ จะมีผลของการรั่วไหลของลอนข้าง ซึ่งสามารถลดผลเสียนี้ได้โดยเปลี่ยนเป็นวินโดว์ที่มีการถ่วงน้ำหนักที่จุดกลางมากที่สุด และลาดลงมาจากจุดเริ่มต้นและจุดสิ้นสุด เช่น แฮมมิงวินโดว์ (Hamming Window), แฮนนิ่งวินโดว์ (Hanning Window)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข -

โปรแกรมควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#include <alloc.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <math.h>
#include <graphics.h>
#include <dos.h>
```

```
#include "zap.h"
#include "gprint.i"
```

```
#define VW 640
#define VH 340
#define TRUE 1
#define FALSE !TRUE
#define SPACE textwidth(" ")
#define HEIGHT textheight("H")
#define WIDTH textwidth("H")
#define BUFSIZE 140
#define NUMWINDOW 5
#define MAIN 0
#define SUB 1
#define LAST 0
#define NOW 1
#define CR 13
#define UP 72
#define LEFT 75
#define RIGHT 77
#define DOWN 80
#define PgUp 73
#define PgDn 81
#define ESC 27
#define Bsp 8
#define Spc 0
#define Tab 32
```

```
#define HUGE 10000
```



```

#define MAX(a,b) (((a) > (b)) ? (a):(b))
#define MIN(a,b) (((a) < (b)) ? (a):(b))

int Ordermenu[7]={5,3,2,2,3,1};
int Namewidth[]={4,8,3,9,8,8,9,9,4,4,9,8,6,9,4,8};
char *Name[]={"DATA","FINDWORD","LPC","RECOGNIZE","PLAYBACK",
  "Get Data","Load Data","Save Data","Find","Comp",
  "Calculate","Save LPC","Config",
  "Recognise","Auto","Playback"};
char *Namesub[]={"GET DATA","LOAD DATA","SAVE DATA","FIND WORD",
  "COMPARE","CALCULATE","SAVE LPC","CHANGE TEMP.",
  "RECOGNIZE","AUTO REC. ","PLAYBACK"};
int *data;
float RC[40][16];
int frame,M;
int REF=0;
int NAME=0;
float wr[260],wi[260],xr[260],xi[260];
float Rc[40][16],Mc[40][16],D[40][40],path[40][40];
int FRAME;
float cal[520];
double X,Y;
float zoom;
int N1,N2;
float en[400];
float mag[250];
float signal[260];
float En;
int NEn;
int START,STOP;
float GAB;
int Spec=0;
int LEN=128;
int speed=111;
int times=2;
int Key;

```

```
char *filename;
char *reffile[3]={"dir","mode","version"};
char *comm[3]={"dir","mode","ver"};
```

```
void Initialize()
{
    int GraphDriver = DETECT;
    int GraphMode,ErrorCode;

    initgraph(&GraphDriver,&GraphMode,"");
    ErrorCode = graphresult();
    if(ErrorCode !=grOk)
    {
        printf("Graphics System Error:%s\n",
grapherrormsg(ErrorCode));
        exit(1);
    }
}
```

```
void setFactor(void)
{
    X=(double)(getmaxx()+1)/VW; /* refer to MODE VGA */
    Y=(double)(getmaxy()+1)/VH;
}
```

```
int dx(int vx)
{
    return((int)X*vx);
}
```

```
int dy(int vy)
{
    return((int)Y*vy);
}
```

```
int gscanfxy(int xloc,int yloc,char *fmt,...)
```

```

{
int cnt=0;

settextjustify(CENTER_TEXT,CENTER_TEXT);
settextstyle(SANS_SERIF_FONT,HORIZ_DIR,2);
setcolor(BLACK);

outtextxy(xloc,yloc,"_");
for(cnt=0;cnt<20;cnt++)
{
fmt[cnt]=toupper(getch());
if(fmt[cnt]==0)
fmt[cnt]=getch();
switch(fmt[cnt]){
case CR: bar(xloc+WIDTH*(cnt)-WIDTH/2,yloc-1-HEIGHT/2,
xloc+WIDTH*(cnt+1),yloc+4+HEIGHT/2);
fmt[cnt]='\0';
return(0);
case Bsp: if (cnt > 0)
{
fmt[cnt-1]='\0';
bar(xloc+WIDTH*(cnt-1)-WIDTH/2,yloc-1-HEIGHT/2,
xloc+WIDTH*(cnt+1),yloc+4+HEIGHT/2);
cnt-=2;
outtextxy(xloc+WIDTH*(cnt+1),yloc,"_");
}
else cnt--;
break;
default : fmt[cnt+1]='\0';
bar(xloc+WIDTH*(cnt)-WIDTH/2,yloc-1-HEIGHT/2,
xloc+WIDTH*(cnt+2),yloc+4+HEIGHT/2);
outtextxy(xloc+WIDTH*(cnt),yloc,&(fmt[cnt]));
if(cnt+1 < 20)
outtextxy(xloc+WIDTH*(cnt+1),yloc,"_");
else cnt--;
break;
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
}

char *get_string(void)
{
    char *alpha;                /* result string pointer */
    int xxx,yyy;

alpha = (char *)·malloc(20);
    if(!alpha) {
        printf("\nString allocation error in get_string\n");
        exit(1);
    }
    xxx=getmaxx();
    yyy=getmaxy();
    setcolor(LIGHTGRAY);
bar(20+10*WIDTH,yyy-33,xxx-210,yyy-15);
gscanfxy(20+12*WIDTH,yyy-24,alpha);
return(alpha);
}

void BEEP()
{
    register int i;
    sound(1000);
    for(i=0;i<=300;i++)
        i=i;
    nosound();
}

void DELAY(int dl)
{
    delay(dl);
    BEEP();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/****** DATA PART *****/
```

```
void INPUT(void)
```

```
{
```

```
    int check,show;
```

```
    int i;
```

```
    int x,y,ty,tx;
```

```
/******
```

```
    SET INITIAL CONDITION FOR 8255
```

```
    MODE 0 ;; A & LOWER C ARE INPUT PORT
```

```
    B & UPPER C ARE OUTPUT PORT
```

```
*****/
```

```
    outportb(COMMAND_PORT,INIT_8255);
```

```
/******GET DATA*****/
```

```
    check= 0;
```

```
    settextstyle(3,0,1);
```

```
    setcolor(DARKGRAY);
```

```
    x=getmaxx();
```

```
    y=getmaxy();
```

```
    setfillstyle(SOLID_FILL,LIGHTGRAY);
```

```
    settextjustify(LEFT_TEXT,CENTER_TEXT);
```

```
    bar(100,y-45,x-220,y-15);
```

```
    tx=135;
```

```
    ty=y-40;
```

```
    gprintf(&tx,&ty,"PRESS ANY KEY WHEN READY ");
```

```
    getch();
```

```
    bar(100,y-45,x-220,y-15);
```

```
/*RESET COUNTER*/
```

```
    outportb(PORT_C,ENABLE_COUNT);
```

```
    outportb(PORT_C,RESET_COUNT);
```

```

outportb(PORT_C,ENABLE_COUNT);

/*WRITE EXTERNAL MEMORY*/
bar(150,y-33,x-220,y-15);
DELAY(500);
setfillstyle(SOLID_FILL,YELLOW);
bar(555,55,615,70);
DELAY(500);
setfillstyle(SOLID_FILL,GREEN);
bar(555,75,615,90);
/* DELAY FOR YOU */

outportb(PORT_C,WRITE_MEM);

/*DELAY FOR WRITE DATA */

while(check != CHECK)
  check=inportb(PORT_C);
setfillstyle(SOLID_FILL,RED);
bar(555,95,615,110);
/*READ DATA FROM EXTERNAL MEMORY*/

outportb(PORT_C,RESET_COUNT);
outportb(PORT_C,ENABLE_COUNT);
check = inportb(PORT_C);

for(i = 0; (i < 20000) && (check != CHECK_END); i++)
{
  data[i]=inportb(PORT_A);
  outportb(PORT_C,STROB_OFF);
  outportb(PORT_C,STROB_ON);
  check = inportb(PORT_C);
}
}

```

เอกสาร void Clear_Data(void) การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    register int i;

    for(i=0;i<20000;i++)
    data[i]=0;
}

void LOADDATA(void)
{
    int status;
    int count=0,num;
    int input,x,y;
    char *filename;
    FILE *data_file;

    y=getmaxy()-40;
    x=160;
    printf(&x,&y,"Enter File Name");
    filename=get_string();
    bar(100,getmaxy()-45,getmaxx()-220,getmaxy()-15);
    data_file=fopen(filename,"rb");
    if(data_file==NULL)
    {
        printf(&x,&y," can not open file ");
        exit(1);
    }
    else for(count = 0; (count < 20000) && (!feof(data_file)); count++)
    {
        status=fread(&input,sizeof(int),1,data_file);
        if(status!=1)
        {
            printf(&x,&y," \nread error ");
            exit(1);
        }
        data[count]=input;
    }
}

```

```

gprintf(&x,&y," can not open file ");
exit(1);
}
for(i=0;i<20000;i++)
{
dat=data[i];
status=fwrite(&dat,sizeof(int),1,data_file);
if(status!=1)
{
gprintf(&x,&y," write error ");
exit(1);
}
}
fclose(data_file);
}

```

```

void OPENFILE(int index)
{
int i,j,x=200,y=getmaxy()-40;
float K;
char *filename;
FILE *fp;

if(index==1)
{
gprintf(&x,&y,"Enter File Name");
filename=get_string();
bar(100,getmaxy()-45,getmaxx()-220,getmaxy()-15);
if((fp=fopen(filename,"rb"))==NULL)
{
gprintf(&x,&y,"Can't Open");exit(1);
}
}
else
if((fp = fopen(reffile[NAME],"rb")) == NULL)
{
gprintf(&x,&y,"Can't Open File : %s",reffile[NAME]);
exit(1);
}
}

```

```

}

fread(&frame,sizeof(int),1,fp);
for(i=1;i<=frame;i++)
for(j=1;j<=15;j++)
{
fread(&K,sizeof(float),1,fp);
if(index==1)
Rc[i][j]=K;
else
Mc[i][j]=K;
}
fclose(fp);
}

void Save_RC(void)
{
int status,x=200,y=getmaxy()-40;
register fr,order; /*fr=frame*/
char *filename;
FILE *data_file;

gprintf(&x,&y,"Enter File Name");
filename = get_string();
bar(100,getmaxy()-45,getmaxx()-220,getmaxy()-15);
data_file=fopen(filename,"wb");
if(data_file==NULL)
{
gprintf(&x,&y,"can not open file ");
exit(1);
}
status=fwrite(&FRAME,sizeof(int),1,data_file);
if(status!=1)
{
gprintf(&x,&y,"write error ");
exit(1);
}
for(fr=1;fr<FRAME;fr++)
for(order=1;order<=15;order++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

status=fwrite(&RC[fr][order],sizeof(float),1,data_file);
if(status!=1)
{
gprintf(&x,&y,"write error ");
exit(1);
}
}
fclose(data_file);
}

```

/****** SCREEN PART *****/

```

int Max(void)
{
register int i;
int max;

max=abs(data[N1]-128);
for(i=N1;i<=N2;i++){
if(abs(data[i]-128) > max) max = abs(data[i]-128);
}
return(max);
}

void Newdat()
{
register int i;
int g=0;
g = Max();
for(i = N1; i< N2; i++)
{
data[i] = (int)((float)(data[i]-128)*128.0/(float)g);
data[i] += 128;
}
}

void Plot(int EV,int start,int end,int MID)
{

```

```

int index;
int Rate,END=0;
float Zoom;

setfillstyle(SOLID_FILL,LIGHTGRAY);
if(Spec!=1)
bar(20,60,419,359);

Rate=1;
if((end-start)>400){
zoom=Zoom=(float)(end-start)/399;
Zoom--=(float)((int)(end-start)/399);
zoom=(int)zoom;
if(Zoom>0.5)zoom++;
}
else
{
Rate=399/(end-start);
zoom=1;
}
GAB=255.0;
if(GAB/zoom<=220 && zoom!=1)GAB=299/GAB;else GAB=1/zoom;
if(Spec==1)GAB/=3;

x1=x2=dx(20);
y1=y2=MID;

for(x=dx(20);x<=dx(419);x+=Rate)
{
index = (x-dx(20))*zoom;
y = ((128-data[index+start])*GAB)+MID;

if(y>=359)y=dy(359);
if(y<=61)y=dy(61);

if(MID==dy(210))
setcolor(YELLOW);
else
setcolor(BLUE);
}

```

```

if(index==0)
    moveto(x,y);
else
    if(EV==0)
        lineto(x,y);
    else
        if(y>MID)
        {
            line(x,y,x1,y1);
            x1=x;y1=y;
        }
        else
        {
            line(x,y,x2,y2);
            x2=x;y2=y;
        }
}

if(MID==dy(210))
{
    if(index+start<=N1 && N1<index+start+zoom)
    {
        setcolor(RED);
line(x-1,dy(60),x-1,dy(359));
    }

    if(index+start<=N2 && N2<index+start+zoom)
    {
        setcolor(RED);
line(x-1,dy(60),x-1,dy(359));
        END++;
    }
    else
if(END==0 && x==dx(419))
    {
        setcolor(RED);
line(x-1,dy(60),x-1,dy(359));
        N2=index+start;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    }
    START=N1;STOP=N2;
}

void PlotEn(int MID)
{
    int x,i;
    int j,y;

    if(GAB==1/zoom)j=1;else j=2;
    setcolor(MAGENTA);
    for(x=dx(20),i=0;x<=dx(419);x+=2,i++){
y=MID-j*GAB*en[i];
        if(y>dy(350))
            y=dy(350);
        if(y<dy(65))
            y=dy(65);
if(x==dx(20))
            moveto(x,y);
        else
            lineto(x,y);
    }
}

/***** FIND BOUND PART *****/

short int Sgn(unsigned int a)
{
    short int value;
    if(a>=128)value=1;
    else value =-1;
    return(value);
}

float zcr(int start, int win)
{

```

```

float zcrate;

zcrate = 0.0;
for(m = start; m < start+win; m++)
{
zcrate += abs(Sgn(data[m+1]-128)-Sgn(data[m]-128));
}
zcrate /= 2.0;
return(zcrate);
}

```

```

float av_mag(int start, int win)
{
register int m;
float mn;

mn = 0.0;
for(m = start; m < start+win; m++)
{
mn += abs(data[m]-128);
}
mn /= win;
return(mn);
}

```

```

void write_line(int x)
{
int far *fndline;
unsigned int size;

x += dx(20);
size = imagesize(x,dy(60),x,dy(359));
fndline = farmalloc(size);
getimage(x,dy(60),x,dy(359),fndline);
putimage(x,dy(60),fndline,NOT_PUT);
farfree(fndline);
}

```

```

#define ITL 3
#define IZCT 14

int find_end(int index, int start, int end, int win)
{
register int i,j,count=0;
unsigned int n;
int point,value,x=200,y=getmaxy()-40;

if(index == 0)
{
n = start;
while((av_mag(n,win) < ITU) && (n <= end))
{
write_line(2*n/win);
n += win/2;
}
if(n > end) gprintf(&x,&y,"Can't Find Begin-Point");
else
{
n -= win/2;
while((av_mag(n,win) > ITL) && (n >= 0))
{
write_line(2*n/win);
n -= win/2;
}
if(n < start) n = start;
else
{
value = point = n;
for(i = point;(i > point-25*win/2)&&(i >= start);i -= win/2)
{
if(zcr(i,win) > IZCT)
{
write_line(2*i/win);
count++;
value = i;
}
}
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

if(count > 3)
{
for(n = i; n < value; n -= win)
write_line(2*value/win);
value = point;
}
}
}
}
return(value);
}
#undef ITU
#undef ITL
#undef IZCT

void Word()
{
N1 = find_end(0,0,19999,100);
N2 = find_end(1,0,19999,100);
}

void Energy(int start,int stop,int win)
{
register int i,j=0;

for(i=start;i<=stop-win/2;i+= win/2)
{
en[j]=av_mag(i,win);
j++;
}
NEn=j-1;
}

/***** LPC PART *****/

void LPC()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น ยกเว้นให้ผมให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

float rc[16],R[22],S,A[22],B[22],ALPHA;
int MP,M=15,K,NP,NPK,NK,MIP,MINC,IP,J,JB,MF,N=250;

for(J=1;J<=21;J++){
R[J]=0;
B[J]=0;
A[J]=0;
rc[J]=0;
}
ALPHA=0.0;
MP=M+1;
for(K=1;K<=MP;K++)
{
R[K]=0.0;
NK=N-K+1;
NP=1;
for(NP=1;NP<=NK;NP++)
{
NPK=NP+K-1;
R[K]=R[K]+cal[NP]*cal[NPK];
}
}
ALPHA=R[1];
rc[1] = (-R[2])/R[1];
A[1]=1.0;
A[2]=rc[1];
ALPHA=ALPHA-rc[1]*rc[1]*ALPHA;
MF=M;
for(MINC=2;MINC<=MF;MINC++)
{
M=MINC-1;
for(J=1;J<=MINC;J++)
{
JB=MINC-J+1;
B[J]=A[JB];
}
M=M+1;
S=0;
for(IP=1;IP<=M;IP++)

```

```

{
    MIP=M-IP+2;
    S=S+R[MIP]*A[IP];
}
rc[M]=--S/ALPHA;

for(IP=2;IP<=M;IP++)
    A[IP]=A[IP]+rc[M]*B[IP-1];

A[M+1]=rc[M];
ALPHA=ALPHA-rc[M]*rc[M]*ALPHA;
if(ALPHA<=0.0)
    fprintf(&x,&y,"ERROR");
}
for(IP=1;IP<=15;IP++)
    RC[FRAME][IP]=rc[IP];
}

void Step_Lpc(int STEP)
{
    register int i;
    int kk, KK;
    int x=75,y=130,Y1=0;

    setcolor(MAGENTA);
    fprintf(&x,&y,"CALCULATE LPC");
    x=75;y=150;
    if(STEP==0){
        kk=38;
        KK=165+20*(int)((kk/20)+1);
        setfillstyle(SOLID_FILL,CYAN);
        bar(50,100,400, KK);
        setcolor(RED);
        rectangle(55,105,395, KK-5);
        rectangle(57,107,393, KK-7);

        for(i=0;i<=kk;i++){
            if((i)%20==0)

```

```

        if(i==0)Y1=0;else Y1+=20;
x=75;
    }
    y=150+Y1;
    setfillstyle(CLOSE_DOT_FILL,LIGHTGRAY);
bar(x,y,x+10,y+15);
    x+=15;
}
}else{
    y+=20*((FRAME-1)/20);
    Y1=FRAME;
    Y1-=(int)((FRAME-1)/20)*20;
    x+=15*(Y1-1);
    setfillstyle(SOLID_FILL,YELLOW);
    bar(x,y,x+10,y+15);
    BEEP();
}
setfillstyle(SOLID_FILL,LIGHTGRAY);
setcolor(DARKGRAY);
}
}

void CAL(void)
{
    int i,index,x=200,y=getmaxy()-40;
    float PI,arg,h[520];

    FRAME=1;
    Step_Lpc(0);
    PI=4.0*atan(1.0);
    KeY = (STOP - START)/39;
    arg=2.0*PI/(KeY);
    for(i=0;i<KeY;i++)
h[i]=0.54-0.46*cos(arg*i);

    for(index=START;index<=STOP-KeY;index+=KeY)
    {
        for(i=0+index;i<KeY+index;i++)
cal[i-index]=en[i/100]*h[i-index];

```

```

    LPC();
    Step_Lpc(1);
    FRAME++;
    if(FRAME>40) break;
}
    gprintf(&x,&y,"Calculation Already!");
}

```

/****** RECOGNITION PART *****/

```

float Average_En(int init,int end)
{
    register int i,j;
    float answer=0;

    for(i=init;i<=end;i++)
        answer+=en[i];
    answer=answer/(end-init+1);
    return(answer);
}

```

```

float ABS(float a)
{
    if(a>=0) a=a;
    else a=-a;
    return(a);
}

```

```

void Write_Speed(void)
{
    int x,y;

```

```

    settxtstyle(3,0,1);
    settxtjustify(LEFT_TEXT,CENTER_TEXT);
    setcolor(YELLOW);
    setfillstyle(SOLID_FILL,BLUE);

```

```

    bar(100,90,340,165);

```

```

setcolor(YELLOW);
rectangle(105,95,335,160);
x=120;y=110;
gprintf(&x,&y,"SPEED : %d",speed);
gprintf(&x,&y,"TIMES : %d",times);
setcolor(CYAN);
gprintf(&x,&y,"use <ENTER> to select");
)

```

```
PLAY_BACK(int xx,int yy)
```

```

{
register int i,j,k;
int KEY;
int out=1;
unsigned char DAT;

outportb(COMMAND_PORT,INIT_8255);
Write_Speed();
setfillstyle(SOLID_FILL,LIGHTGRAY);
do{
KEY=getch();
if(KEY!=0){
switch(KEY){
case UP :if(out!=2)speed++;else times++;
Write_Speed();
break;
case DOWN:if(out!=2)speed--;else times--;
if(times==0)times=1;
Write_Speed();
break;
case CR :out++;
BEEP();
break;
case Tab :
setfillstyle(SOLID_FILL,LIGHTGRAY);
if(out!=2)speed=atoi(get_string());else times=atoi(get_string());
if(times==0)times=1;
Write_Speed();
break;

```

เอกสารนี้ break; สารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}
}while(out!=3);
for(k=0;k<times;k++){
for(i=xx;i<=yy;i++){
for(j=0;j<=speed;j++)
i=i;
DAT=(unsigned char)(data[i]);
outportb(PORT_B,DAT);
}
BEEP();
}
}

```

```

void getref()
{
register int i;
for(i = 0; i < 2; i++)
{
reffile[i] = get_string();
comm[i] = get_string();
}
}
float DynamicPro(int M,int N)
{
register int i,j,k;
float tempreal,A,B,C,d;
int x,y;

for(i=1;i<=M;i++)
{
for(j=1;j<=N;j++)
{
tempreal=0.0;
for(k=1;k<=15;k++)
tempreal+=ABS(Rc[i][k]-Mc[j][k]);
D[i][j]=tempreal;
}
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

path[0][0]=0.0;

for(j=1;j<=N;j++)
path[0][j]=0.0;

for(i=1;i<=M;i++)
{
    path[i][0]=0.0;
for(j=1;j<=N;j++)
    {
        d=D[i][j];
A=path[i-1][j]+d;
B=path[i-1][j-1]+sqrt(2*d*d);
        C=path[i][j-1]+d;
if(A<B && A<=C) path[i][j]=A;
        if(B<=A && B<=C) path[i][j]=B;
if(C<A && C<B) path[i][j]=C;
    }
}
tempreal=path[M][N]/(M*N);

settextstyle(3,0,1);
settextjustify(LEFT_TEXT,CENTER_TEXT);
setcolor(YELLOW);
x=120;y=110+(NAME+3)*18;
fprintf(&x,&y," %s : %f",comm[NAME],tempreal);
return(tempreal);
}

```

```

void Clear_Rc(int Test)

```

```

{
    register int i,j;

for(i=0;i<40;i++)
    for(j=0;j<16;j++)
    {
        if(Test==1)
            Rc[i][j]=0;

```

```

    }
    for(i=0;i<40;i++)
        for(j=0;j<40;j++)
            {
                D[i][j]=path[i][j]=0;
            }
    }

void DIGIT(void)
{
    unsigned int i,j,k;
    int x=120,y=getmaxy()-40;
    int choose,N,num;
    float digit,word;

    Clear_Rc(1);
    M = FRAME-1;
    for(i = 1; i <= M; i++)
for(j = 1; j <= 15; j++)
Rc[i][j] = RC[i][j];
        setfillstyle(SOLID_FILL,BLUE);
        bar(100,90,320,300);
        setcolor(YELLOW);
rectangle(105,95,315,295);
for(NAME=0;NAME<2;NAME++){
    Clear_Rc(0);
    OPENFILE(2);

    N=39;
    digit = DynamicPro(M,N);
    if (NAME == 0)
    {
        num = NAME;
        word = digit;
    }
    if (digit < word)
    {
        num = NAME;
        word = digit;
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
if(word < 1.5)
{
getch();
free(data);
closegraph();
system(comm[num]);
exit(0);
}
else
    printf(&x,&y,"No one match,in my opinion");
    setfillstyle(SOLID_FILL,LIGHTGRAY);
}

/***** MENU PART *****/

void Light(void)
{
    register int i;

    for(i=55;i<=100;i+=20){
        setfillstyle(CLOSE_DOT_FILL,WHITE);
        bar(555,i,615,i+15);}
}

void Pop_Plate(int Up,int xleft,int ytop,int xright,int ybottom)
{
    int poly[14];
    int Gab;

    if(Up==1)
        {setcolor(DARKGRAY);setfillstyle(SOLID_FILL,DARKGRAY);Gab=4;}
    else
        {setcolor(WHITE);setfillstyle(SOLID_FILL,WHITE);Gab=3;}
    poly[0] = poly[12] = xleft;
    poly[1] = poly[13] = poly[11] = ybottom;
    poly[2] = xleft-Gab+1;
    poly[3] = poly[5] = ybottom+Gab;
    poly[4] = poly[6] = xright+Gab;

```

```

poly[7] = ytop-Gab;
poly[8] = poly[10] = xright;
poly[9] = ytop;
fillpoly(7,poly);
if(Up==1)
{setcolor(WHITE);setfillstyle(SOLID_FILL,WHITE);Gab=3;}
else
{setcolor(DARKGRAY);setfillstyle(SOLID_FILL,DARKGRAY);Gab=4;}
poly[0] = poly[10] = poly[12] = xleft;
poly[1] = poly[13] = ybottom;
poly[2] = poly[4] = xleft-Gab;
poly[3] = poly[6] = ybottom+Gab;
poly[6] = xright+Gab;
poly[5] = poly[7] = ytop-Gab;
poly[8] = xright;
poly[9] = poly[11] = ytop;
fillpoly(7,poly);
setcolor(BLACK);
}

void Write_Block(int status,int x,int y,int order)
{
int width;

if(order>=5)settextstyle(2,0,5);else settextstyle(3,0,1);
width=(Namewidth[order]+2)*WIDTH;
setfillstyle(SOLID_FILL,LIGHTGRAY); /*IF STATUS=0 MENU IS UNUSE*/
bar(x-10,y-10,x+width,y+30); /*IF STATUS=1 MENU IS USE */
setfillstyle(SOLID_FILL,BLACK);
bar(x,y,x+width,y+30);
setfillstyle(SOLID_FILL,LIGHTGRAY);
if(status==0)
{
bar(x-8,y-8,x-8+width,y+22);
setcolor(BLUE);
rectangle(x-8,y-8,x-8+width,y+22);
x+=WIDTH-8;
y+=7;

```

เอกสาร **setcolor(WHITE)**; สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
else
{
bar(x-1,y-1,x-1+width,y+29);
setcolor(BLUE);
rectangle(x-1,y-1,x-1+width,y+29);
x+=WIDTH;
y+=14;
setcolor(RED);
BEEP();
}
settextjustify(LEFT_TEXT,CENTER_TEXT);
gprintf(&x,&y,"%s",Name[order]);
}

void Do_Menu(int sub,int status,int pos,int LABEL)
{
int x=480,y;

if(sub==1)x=570;else y=145;
switch(pos){
case 1: if(sub==1)y=195; else y=140;
break;
case 2: if(sub==1)y=240; else y=145+5*WIDTH;
break;
case 3: if(sub==1)y=285; else y=145+5*WIDTH*2+5;
break;
case 4: if(sub==1)y=330; else y=145+5*WIDTH*3+10;
break;
case 5: if(sub==1)y=375; else y=145+5*WIDTH*4+15;
break;
}
Write_Block(status,x,y,LABEL-1);
}

```

```

void Write_Main_Menu(int index)

```

```

{

```

เอกสารนี้ **register** **int** **i**; ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int y=140,x=480;

for(i = 0; i < 5; i++)
{
if(i == index)
Write_Block(1,x,y,i);
else
Write_Block(0,x,y,i);
y += (WIDTH+1)*5;
}
}

int Order_Sub(int pos)
{
int order;
switch(pos){
case 1:order=Ordermenu[0];
break;
case 2:order=Ordermenu[0]+Ordermenu[1];
break;
case 3:order=Ordermenu[0]+Ordermenu[1]+Ordermenu[2];
break;
case 4:order=Ordermenu[0]+Ordermenu[1]+Ordermenu[2]+
Ordermenu[3];
break;
case 5:order=Ordermenu[0]+Ordermenu[1]+Ordermenu[2]+
Ordermenu[3]+Ordermenu[4];
break;
case 6:order=Ordermenu[0]+Ordermenu[1]+Ordermenu[2]+
Ordermenu[3]+Ordermenu[4]+Ordermenu[5];
break;
}
return(order);
}

```

```

void Write_Sub_Menu(int USAGE,int x,int y,int pos)
{

```

```

int order;

settextstyle(2,0,5);
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(x-15,y-15,x+75,y+205);
setfillstyle(SOLID_FILL,BLACK);
bar(x,y,x+75,y+205);
if(USAGE==1)Gab=-8;else Gab=-1;
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(x+Gab,y+Gab,x+75+Gab,y+205+Gab);
setcolor(BLUE);
rectangle(x+Gab,y+Gab,x+75+Gab,y+205+Gab);
order=Order_Sub(pos);
Write_Block(USAGE+1,x+23+Gab,(y+33+Gab),order);
if(USAGE==0)USAGE=2;
for(i=1;i<Ordermenu[pos];i++)
Write_Block(USAGE-1,x+23+Gab,(y+33+Gab)+45*i,order+i);
if(USAGE==2){y+=8;x+=5;}
setcolor(BLACK);
gprintf(&x,&y,"Sub Menu");
settextstyle(3,0,1);
}

```

```

void Write_Namesub(int order)
{
int x,y;
x=getmaxx();y=getmaxy();
bar(x-160,y-36,x-15,y-10);
setcolor(BLINK);
settextjustify(LEFT_TEXT,CENTER_TEXT);
settextstyle(3,0,1);
x=x-160+WIDTH;y-=26;
gprintf(&x,&y,"%s",Namesub[order-5]);
}

```

```

void Color(void)
{
register int i;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int x=100,y=215;

settextstyle(0,0,1);
setcolor(BLACK);
gprintf(&x,&y,"MAX");
x+=35;
for(i=0;i<13;i++){
switch(i){
case 0: color=14;break;
case 1: color=2;break;
case 2: color=10;break;
case 3: color=3;break;
case 4: color=11;break;
case 5: color=6;break;
case 6: color=13;break;
case 7: color=12;break;
case 8: color=5;break;
case 9: color=4;break;
case 10: color=8;break;
case 11: color=1;break;
case 12: color=0;break;
}
setfillstyle(SOLID_FILL,color);
bar(x-50,200,x-30,215);
x+=25;
}
x+=10;
y=215;
gprintf(&x,&y,"MIN");
settextstyle(3,0,1);
}

```

```

void Write_Spectrum_Menu(int SP)

```

```

{
Spec=SP;
setfillstyle(SOLID_FILL,LIGHTGRAY);
bar(5,45,440,375);
Pop_Plate(1,10,50,429,370);

```

เอกสาร Pop_Plate(0,20,60,419,360); งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

}

```
void Sub_Menu(int pos)
```

```
{
```

```
int KEY;
```

```
int out=0;
```

```
int SMENU,UPPER,LOWER;
```

```
int MID,N;
```

```
MID=dy(210);
```

```
UPPER=SMENU=Order_Sub(pos);
```

```
LOWER=Order_Sub(pos+1);
```

```
do{
```

```
KEY=getch();
```

```
if(KEY!=0){
```

```
switch(KEY){
```

```
case UP :
```

```
Do_Menu(SUB, LAST, SMENU-UPPER+1, SMENU+1);
```

```
SMENU--;
```

```
if(SMENU==UPPER-1)SMENU=LOWER-1;
```

```
Do_Menu(SUB, NOW, SMENU-UPPER+1, SMENU+1);
```

```
Write_Namesub(SMENU);
```

```
break;
```

```
case DOWN :
```

```
Do_Menu(SUB, LAST, SMENU-UPPER+1, SMENU+1);
```

```
SMENU++;
```

```
if(SMENU==LOWER)SMENU=UPPER;
```

```
Do_Menu(SUB, NOW, SMENU-UPPER+1, SMENU+1);
```

```
Write_Namesub(SMENU);
```

```
break;
```

```
case CR :
```

```
BEEP();
```

```
switch(SMENU){
```

```
case 5 :Light();
```

```
INPUT();
```

```
N1=0;N2=19999;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Plot(0,N1,N2,MID);
    Newdat();
Plot(0,N1,N2,MID);
Energy(N1,N2,200);
PlotEn(MID);
    BEEP();
    break;
case 6 :Clear_Data();
    LOADDATA();
    BEEP();
N1=0;N2=19999;
Plot(0,N1,N2,MID);
Energy(N1,N2,200);
PlotEn(MID);
    BEEP();
    break;
case 7 :Save_data();
    BEEP();
    break;
case 8:N1=0;N2=19999;
    Word();
    Plot(0,N1,N2,MID);
    START=N1;STOP=N2;
break;
case 9:
Write_Spectrum_Menu(1);
Clear_Data();
LOADDATA();
N1 = 0; N2 = 19999;
Plot(0,N1,N2,150);
Clear_Data();
LOADDATA();
N1 = 0; N2 = 19999;
Plot(0,N1,N2,250);
break;
case 10:
CAL();
Plot(0,N1,N2,MID);

```

```

        break;
case 11:
    Save_RC();
    break;
case 12:
    getref();
    break;
case 13:
    DIGIT();
    Plot(0,N1,N2,MID);
    break;
case 14:
    N1 = 0; N2 = 19999;
    Word();
    START = N1;
    STOP = N2;
CAL();
    DIGIT();
    break;
case 15:
    PLAY_BACK(N1,N2);
    break;
}
break;
case ESC:
out=3;
setcolor(LIGHTGRAY);
    bar(100,getmaxy()-45,getmaxx()-220,getmaxy()-15);
bar(480,130,659,399);
Write_Main_Meriu(pos-1);

    break;
}
}
}
while(out!=3);
}

```

```

main()
{
    register int i;
    int xx,yy,x,y;
    int KEY,MENU=1,out=0;

    data=(int *)calloc(20000,sizeof(int));
    Initialize();
    setFactor();

    x=getmaxx();
    y=getmaxy();

    setfillstyle(SOLID_FILL,LIGHTGRAY);
    setttextjustify(LEFT_TEXT,CENTER_TEXT);
    setttextstyle(3,0,1);
    bar(0,0,x,y);
    Write_Main_Menu(0);
    Pop_Plate(1,10,y-56,x-209,y-10);
    Pop_Plate(1,x-165,y-36,x-15,y-10);
    xx=20+WIDTH;
    yy=y-23;
    fprintf(&xx,&yy,"FILE      : ");
    yy=y-40;
    fprintf(&xx,&yy,"MESSAGE : ");
    Write_Spectrum_Menu(0);/*NO SPECTRUM IN MENU*/
    Light();

    do{
        KEY=getch();
        if(KEY!=0){
            switch(KEY){
            case UP :
                Do_Menu(MAIN, LAST, MENU, MENU);
                MENU--;
            if (MENU==0)MENU=5;
                Do_Menu(MAIN, NOW, MENU, MENU);
                break;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case DOWN :
    Do_Menu(MAIN, LAST, MENU, MENU);
        MENU++;
if (MENU==6) MENU=1;
    Do_Menu(MAIN, NOW, MENU, MENU);
    break;
case CR :
Write_Sub_Menu(1, 555, 170, MENU);
    Sub_Menu(MENU);
    break;
case ESC:
    out=3;
    break;
}
}
}
while(out!=3);

BEEP();
free(data);
closegraph();
}

```

