การรู้จำเสียงพูดเลขไทย

Speech Recognition of Thai Numberal

โดย

นายผดุงศิลป์ ล้อมพรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

การรู้จำเสียงพูดเลขไทย

Speech Recognition of Thai Numberal

โดย

นายผดุงศิลป์ ล้อมพรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

การรู้จำเสียงพูดเลขไทย

Speech Recognition of Thai Numberal

โดย

นายผดุงศิลป์ ล้อมพรม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญา
วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2536

ปริญญานิพนธ์ปีการศึกษา 2536

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง การรู้จำเสียงพูดเลขไทย

ผู้จัดทำ นายผดุงศิลป์ ล้อมพรม รหัส 33100226

.....อาจารย์ที่ปรึกษา

(อาจารย์ วัชระ ฉัตรวิริยะ)

การรู้จำเสียงพูดเลขไทย
SPEECH RECOGNITION OF THAI NUMBERAL

โดย นายผดุงศิลป์ ล้อมพรม 33100226

อาจารย์ที่ปรึกษา อาจารย์ วัชระ ฉัตรวิริยะ

บทคัดย่อ

การรู้จำเสียงพูดภาษาไทยในการวิจัยครั้งนี้เป็นการรู้จำตัวเลขสิบตัวคือ เลข 0 - 9 โดยระบบนี้เป็นระบบที่ทำงานกับเสียงพูดเฉพาะบุคคล ระบบที่สร้างขึ้นมาแบ่งออกเป็น 3 ขั้นตอนใหญ่ๆ คือขั้นตอนการหาค่าพารามิเตอร์,ขั้นตอนการฝึกอบรมและการรู้จำและขั้นตอนการประยุกต์ใช้งานในขั้นตอนแรกคือการหาค่าพารามิเตอร์จะเป็นการวิเคราะห์หาขอบเขตของคำโดยใช้การเปรียบเทียบค่าของพลังงานจะเป็นการเปรียบเทียบค่านี้ระหว่างช่วงที่มีเสียงพูดกับช่วงที่คิดว่าจะมีเสียงพูดในขั้นตอนที่สองเป็นขั้นตอนการฝึกอบรมให้เกิดการรู้จำเสียงพูดโดยใช้ค่าพารามิเตอร์ที่ได้จากขั้นตอนแรกเป็นข้อมูลในการฝึกอบรมนิวรอลเนตเวิร์คจนสามารถรู้จำเสียงพูดที่ให้เป็นตัวอย่างได้สมบูรณ์ขั้นตอนการประยุกต์ใช้งานเป็นการนำระบบที่ได้ไปทดสอบกับเสียงอื่นที่ไม่ได้ฝึกฝนมาโดยขบวนการคล้ายกับการฝึกฝนเนตเวิร์คซึ่งจะไม่มีการปรับค่าน้ำหนักและแสดงผลที่ได้จากการรู้จำ

ABSTRACT

Speech Recognition of Thai Number in this research is limited to number recognition from 0-9 and a speaker dependent system.The content of this research are devided into three main parts,parameter searching part,trainning and recognition for analyzed speech part and application part.Firstly,In the parameter searching part,it is a word boundary analysis by comparing energy level between the non-speech interval to the speech interval.In the second part,the parameter received from a first part is used to train computer to recognized speech by input to neural network until networks recognize exampled speech completely. Finally,this system will be applied to new speech that has never encountered during training.This process is the same as training process except adaptation weight between each layer and display recognized-output

สารบัญ

สารบัญรูปภาพ

บทที่ 1
บทนำ

1.1.ความเป็นมาของงานวิจัย

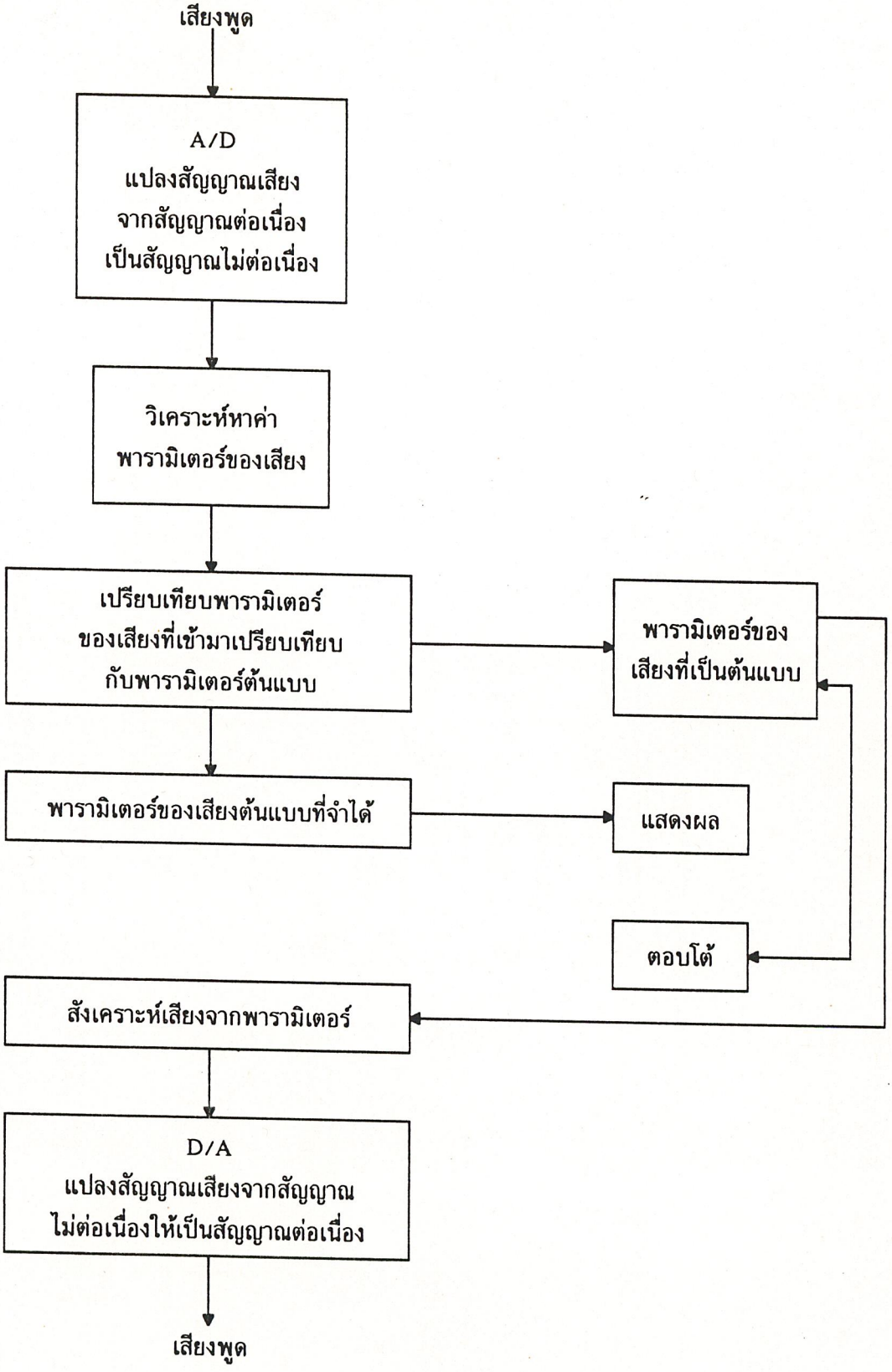
ในปัจจุบันคอมพิวเตอร์ได้เข้ามามีบทบาทเป็นอย่างมากไม่ว่าเป็นเทคโนโลยีสาขาใดๆ ล้วนแล้วแต่จำเป็นต้องพึ่งพาคอมพิวเตอร์ทั้งสิ้นอีกทั้งในด้านอุตสาหกรรมการผลิตหรือแม้แต่กระทั่งในชีวิตประจำวันของเราเอง ก็ได้ใช้ประโยชน์จากคอมพิวเตอร์ ไม่น้อยเลยทีเดียว

การที่คอมพิวเตอร์จะสามารถอำนวยประโยชน์แก่เราได้นั้นจำเป็นต้องมีการติดต่อกันระหว่าง ผู้ใช้กับเครื่องคอมพิวเตอร์ คือเครื่องคอมพิวเตอร์จำเป็นต้องรู้ว่าผู้ใช้ต้องการอะไรเสียก่อนผู้ใช้จะต้องใช้คำสั่งที่คอมพิวเตอร์สามารถเข้าใจได้ซึ่งคำสั่งนั้นก็จะอยู่ในรูปของสัญญาณไฟฟ้า โดยจะมีการส่งผ่านสัญญาณนั้นทางสวิทช์คอมพิวเตอร์จะรับสัญญาณเข้ามาแล้วดำเนินตามคำสั่งนั้นๆในการที่จะติดต่อคอมพิวเตอร์ให้ได้นั้นมนุษย์จะต้องเข้าไปใช้ภาษาของคอมพิวเตอร์คือสัญญาณไฟฟ้า

ในขณะที่ภาษาที่มนุษย์ใช้ติดต่อกันได้อย่างรวดเร็วที่สุดคือการใช้เสียงพูดคุยกันดังนั้นเพื่อที่จะสามารถใช้ภาษาของมนุษย์ติดต่อกับเครื่องคอมพิวเตอร์ให้ได้นั้นจำเป็นต้องมีการพัฒนาในส่วนที่จะรับรู้เสียงและส่วนที่จะพูดให้กับคอมพิวเตอร์คือทำอย่างไรให้คอมพิวเตอร์สามารถฟังและพูด ภาษาเดียวกับมนุษย์ได้เพื่อให้การติดต่อกันระหว่างมนุษย์กับคอมพิวเตอร์เป็นไปในลักษณะที่มนุษย์เข้าใจได้ง่ายและคุ้นเคย

ในการวิจัยครั้งนี้ เป็นการศึกษาวิธีการที่จะทำให้เครื่องคอมพิวเตอร์สามารถรับรู้และเข้าใจภาษาพูดของมนุษย์ได้เพื่อช่วยอำนวยความสะดวกในการติดต่อสื่อสารกันระหว่างมนุษย์กับเครื่องคอมพิวเตอร์

เนื่องจากสัญญาณเสียงที่มนุษย์ใช้ในการติดต่อสื่อสารกันนั้นเป็น สัญญาณของคลื่นเสียง (sound wave) ซึ่งโดยปกติทั่วๆไปแล้วจะมีรูปแบบของภาษาที่ใช้ในการติดต่อเหมือนกัน(ในภาษาเดียวกัน) และรูปแบบ(pattern)ของคลื่นเสียงที่คนเราพูดออกมานั้นจะคล้ายๆกันเนื่องจากสัญญาณเสียงเป็นสัญญาณที่ถูกทำให้เกิดผิดเพี้ยน(distort) ได้ง่ายจึงทำให้สัญญาณเสียงของแต่ละคนแตกต่างกันขึ้นอยู่กับ



รูปที่ 1.1 บล็อกไดอะแกรมสำหรับการติดต่อกับเครื่องคอมพิวเตอร์โดยใช้เสียงพูด

1. ความเร็ว

2. เสียงรบกวน

3. การออกเสียงที่ผิดหลักเกณฑ์ทางภาษา

แต่เราสามารถทราบลักษณะร่วมของสัญญาณเสียง จากมนุษย์ได้เราก็จะ สามารถอาศัยลักษณะร่วมกันนั้นเป็นตัวชี้ที่จะใช้ในการแปลงจากสัญญาณเสียงเป็นสัญญาณเครื่องได้

1.2. วัตถุประสงค์

1.เพื่อศึกษาและหาแนวทางให้คอมพิวเตอร์สามารถรู้จำเสียงมนุษย์ที่เป็นเสียงภาษาไทยได้

2.เพื่อทำการศึกษาถึงวิธีการและแนวทางในการหาพารามิเตอร์ที่จะมาใช้ในระบบการ รู้จำเสียงพูด

3.เพื่อทำการสร้างระบบรู้จำเสียงพูดที่ใช้วิธีใหม่มาช่วยในการรู้จำเสียงพูดคือการใช้นิวรอลเนตเวอร์ค(Neural Network)ที่เป็นเทคโนโลยีใหม่

4 สามารถนำระบบการรู้จำที่ได้ไปประยุกต์ใช้งานในชีวิตประจำวันได้เช่น ในระบบโทรศัพท์

1.3.ขอบเขตของงานวิจัย

1. เสียงพูดนั้นจะต้องเป็นเสียงภาษาไทยคือ เสียงเลข 0,1,2, ...9 เท่านั้น

2. ต้องพูดในที่ๆ มีเสียงรบกวนต่ำ

3. ผู้พูดจะต้องพูดดังพอสมควร

4. ผู้พูดจะต้องไม่พูดเร็วหรือช้าเกินไป

5. ระบบรู้จำที่สร้างขึ้นมาจะใช้กับเสียงพูดของเฉพาะบุคคล (speaker dependent)

1.4.ขั้นตอนและวิธีดำเนินงานวิจัย

1. หาหลักเกณฑ์ที่จะใช้เป็นเกณฑ์ในการรู้จำเสียงพูด

2. ออกแบบซอฟแวร์ที่จะใช้

3. ทดสอบและแก้ไขการทำงาน

4. สรุปงานวิจัย ปัญหาและข้อเสนอแนะ

1.5.ประวัติการค้นคว้าเกี่ยวกับระบบการตรวจรู้เสียง

การค้นคว้าพัฒนาเกี่ยวกับระบบการตรวจรู้เสียงนี้มีมานานกว่า 30 ปีและ ได้มีการสร้างระบบทดลองต่างๆมากมายบางส่วนของการวิจัยเหล่านี้ได้แก่

ในปี ค.ศ. 1952 Davis,Wiren และ Biddulph จาก Bell Laboratory ได้สร้างระบบตรวจรู้เสียงพูดตัวเลขได้ 10 ตัวสามารถจำคำพูดจากผู้พูด 1 คนได้ถูกต้อง 100 %โดยผู้พูดพูดผ่านโทรศัพท์และความถูกต้องลดลงต่ำกว่า 50 % ในกรณีเปลี่ยนคนพูด

ในปี ค.ศ. 1956 Wiron และ Stubbs จาก Northeastern University ได้สร้างเครื่องตรวจรู้แยกประเภทของการออกเสียงเช่น คำเสียงก้อง(voiced) คือเสียงไม่ก้อง (unvoiced) คำที่มีเสียงกัก(stop) และเสียงเสียดแทรก(fricative) เสียงแหลม(active) เสียงต่ำ(grave) เป็นต้น ระบบสามารถตรวจรู้ได้อย่างถูกต้อง 94 %สำหรับผู้พูด 24 คน โดยจำเสียงสระในปีเดียวกัน Olson และ Belar จาก R C A ได้สร้างระบบสามารถตรวจรู้คำ 1 พยางค์ได้ 10 คำ มีความถูกต้อง 98 % สำหรับผู้พูด 1 คนโดยผู้พูดจะต้องระวังการออกเสียงแต่ละพยางค์ในการพูดประโยค 1 ประโยค และจะต้องหยุดระหว่างคำระบบที่ใช้วงจรกรองแถบถี่ 8 วงจรและ แบ่งการวิเคราะห์แต่ละสมาชิกแบ่งออกเป็น 5 ช่วงเวลาต่อคำ ซึ่งจะได้เมตริกซ์ขนาด 8x5 ใช้ในการวิเคราะห์แต่ละสมาชิกของเมตริกซ์จะมีค่า 1 หรือ 0 ขึ้นอยู่กับค่าพลัง งานแต่ละแถบความถี่จะมีมากหรือน้อยกว่าระดับที่กำหนดไว้

ในปี ค.ศ. 1960 Danes และ Mathew ได้สร้างระบบตรวจรู้เลข 10 จำนวนโดยใช้เครื่องวิเคราะห์สเปคตรัมขนาด17 ช่องความถี่จับสัญญาณและบันทึกข้อมูลลงเทปแม่เหล็กจากนั้นข้อมูลเข้าสู่คอมพิวเตอร์เพื่อทำการวิเคราะห์ด้านความถี่และเวลาแล้วเก็บบันทึกไว้เป็นแบบอ้างอิงคำพูดใหม่ที่เข้ามาจะถูกวิเคราะห์สเปคตรัมแล้วเปรียบเทียบโดยขบวนการ ครอสโคริชัน(Crosscorrelation) กับแต่ละรูปแบบที่เก็บไว้ซึ่งระบบจะเลือกรูปแบบที่คล้ายกันมากที่สุดกับรูปแบบของคำพูดใหม่ การเปรียบเทียบทำโดยการนอมัลไลซ์(Normalization) และการ ไม่นอมัลไลซ์

ในปี ค.ศ. 1966 King และ Tunis ได้สร้างเครื่องตรวจรู้คำ (word recognittion) โดยใช้เครื่องวิเคราะห์สเปคตรัม 15 ช่อง และคอมพิวเตอร์ช่วยในการ วิเคราะห์โปรแกรมคอมพิวเตอร์ทำหน้าที่ตรวจจับยอดที่เกิดขึ้นในแต่ละแถบความถี่โดยหาตำแหน่งที่เกิดสร้างเป็นแมทตริกซ์ตามความถี่และ เวลา(frequency & time matrix) ซึ่งสมาชิกจะมีค่าเป็น 1 ถ้าในแถบความถี่ ณ เวลานั้นเกิดยอดคลื่นขึ้นและเป็น 0 ถ้าไม่มียอดคลื่นผลลัพธ์ให้ความผิดพลาด 0.2 %เมื่อตัวอย่างเสียงเก็บจากผู้พูดคนเดียวกันและผิดพลาดมากที่สุด 2.5 % การสอนให้ระบบรู้จักจะใช้คำพูด 15 คำ

คำ แต่ละคำจะใช้ตัวอย่าง 35 ถึง 50 ครั้ง การทดลองให้ผู้ทดลองพูดคำต่างๆ 90 ถึง 115 ครั้ง ผลลัพธ์จะผิดพลาดมากที่สุด เมื่อตัวอย่างเสียงพูดเก็บจากคนหนึ่งและใช้ผู้พูดเป็นอีกคนหนึ่ง ความผิดพลาดจะเกิดขึ้นประมาณ 46 % แต่เก็บตัวอย่างเสียง กระทำจากคนทั้งสองคนความผิดพลาดจะมีเพียง 0.8 %

ในปี ค.ศ. 966 Fraipont ได้สร้างเครื่องตรวจรู้คำ โดยใช้วงจรกรองแถบความถี่ 10 แถบ กรองความถี่ในช่วง 300 Hz ถึง 3000 Hz โดยการจับเฉพาะแนวยอดคลื่น (envilope) และใช้วงจรกรองความถี่อีก 2 วงจรสำหรับความถี่สูงกว่า 3000 Hz การสุ่มตัวอย่างสัญญาณใช้เวลา 500 ms.การสุ่มคำพูดกระทำ 192 ครั้งข้อมูลถูกบันทึกลงบนเทปกระดาษ แล้วป้อนเข้าสู่คอมพิวเตอร์ ซึ่งทำการวิเคราะห์โดยใช้ลิเนียร์ ดิซิชั่นไฮเปอร์เพลน (linear decision hyperplane) ซึ่งจะถูกปรับให้เหมาะสมระหว่างการสอนระบบ ระบบถูกสร้างให้สามารถตรวจ รู้วงจรได้ 10 ตัวและคำอื่นๆอีก 5 คำ ข้อมูลตัวอย่างถูกเก็บจากคน 24 คนแต่ละคนจะพูดคำหนึ่งคำเพียงครั้งเดียวหลังจากสอนระบบเรียบร้อยแล้วระบบสามารถตรวจรู้ได้ถูกต้อง 99 % สำหรับผู้พูดกลุ่มเดิมและถูกต้อง 87 % เมื่อเปลี่ยนกลุ่มผู้พูด

ในปี ค.ศ.1973 สุพงศ์ เภกะนันท์ ได้วิจัยการตรวจรู้เสียงพูดตัวเลขในภาษาไทย โดยใช้เทคนิคการวิเคราะห์สัญญาณตามความถี่ และได้ผลในการตรวจรู้ถูกต้อง 60 %แต่ถ้าผู้พูดได้รับการฝึกฝนเป็นพิเศษ จะสามารถตรวจรู้ได้ถูกต้องมากกว่า 90 %

ในปี ค.ศ. 1985 วีระ รั้วพิทักษ์ ได้วิจัยการตรวจรู้พยางค์ ในภาษาไทยโดยใช้สัมประสิทธิ์การสะท้อนกลับ เป็นพารามิเตอร์ที่ใช้ในการเปรียบเทียบได้ผลถูกต้อง 89 % และได้ใช้พารามิเตอร์ ความถี่มูลฐานในการแยกวรรณยุกต์ซึ่งได้ผลถูกต้องในการตรวจรู้ 92.8 %

บทที่ 2
เสียงพูดในภาษาไทย

2.1.ลักษณะร่วมของเสียงพูดในภาษาไทย

เสียงพูดในภาษาไทยถ้าพิจารณาในแต่ละเสียงจะมีลักษระที่แตกต่างกันในทำนองเดียวกันก็มีลักษณะคล้ายคลึงกันบ้างลักษณะเหล่านี้ได้แก่

2.1.1. ความก้องหรือไม่ก้องของเสียง(Voiced & Voiceless)

อวัยวะส่วนที่ทำให้เกิดความก้องหรือไม่ก้องของสียงคือเส้นเสียง(Vocal Cords) เสียงก้องเกิดจากการออกเสียงในขณะที่เส้นเสียงปิด ส่วนเสียงไม่ก้องเกิดจากการออกเสียงเมื่อเส้นเสียงเปิด

2.1.2. ความยาวเสียง(Length)

หมายถึงการที่เสียงใดเสียงหนึ่งจะเปล่งมานานเท่าใดในภาษาไทยนั้นความสั้นยาวก็หมายถึงเสียงสระเท่านั้น

2.1.3. ระดับเสียงสูงต่ำ(Pitch Of Voice)

เสียงจะมีระดับสูงหรือต่ำก็อยู่ที่ความถี่ของเสียง อวัยวะส่วนที่ทำให้เสียงมีระดับสูงหรือต่ำก็คือ เส้นเสียง ในการพูดเสียงที่จะมีระดับสูงต่ำได้ก็คือ เสียงก้องเท่านั้นเพราะมีการสั่นสะเทือนของเส้นเสียงซึ่งทำให้มีความถี่ระดับต่างๆได้

2.1.4. ความดังของเสียง(Loundness)

ขึ้นอยู่กับปริมาณของลมที่ผู้พูดเปล่งออกมารวมกับคุณลักษณะประจำคำของเสียงพูดการลงตัวการลงน้ำหนักและความยาวของเสียงประกอบกัน

2.1.5. การลงน้ำหนักของเสียง(Stress)

ความแรงที่ใช้ในการเปล่งเสียงของพยางค์แต่ละพยางค์นั้นจะต้องทำหน้าที่อย่างแข็งขันทำให้พยางค์นั้นดังกว่าพยางค์ที่อยู่ข้างเคียง

2.1.6. รอยต่อของเสียง(Juncture)

เสียงที่เรียงกันมาในคำพูดของคนนั้นมีรอยต่อระหว่างเสียงไม่เหมือนกันบางเสียงก็อยู่ชิดกันบางเสียงก็อยู่ห่างกันถ้าอยู่ห่างกันมากเราก็ก็อาจจะหยุดได้แต่ถ้าอยู่ชิดกัน ก็ไม่อาจจะหยุดได้เสียงที่รวมกันเป็นพยางค์หนึ่งจะเชื่อมกันสนิทจนเกือบจะไม่เห็นรอยต่อที่เรียกว่า Closure Juncture เสียงที่ปรากฏอยู่คนละพยางค์หรือคนละคำก็จะมีรอยต่อห่างกันอย่างสังเกตได้ชัดเรียกว่า Open Juncture

2.2.อักษรที่ใช้แทนเสียง

เพื่อช่วยให้การศึกษาเสียงพูดแต่ละเสียงเป็นไปได้สะดวกขึ้นนักสัทศาสตร์ได้ คิดอักษรชุดหนึ่งเป็นอักษรที่ใช้แทนเสียง(Phonetic Alphabet) โดยกำหนดให้แสดงลักษณะของการออกเสียงมากกว่าจะแสดงเสียงที่ปรากฏในภาษาใดภาษาหนึ่ง

2.3.ลักษณะของเสียงพยัญชนะภาษาไทย

ลักษณะของเสียงพยัญชนะไทยนั้นมีอยู่ 3 ลักษณะดังนี้

1.3.1.ความก้องหรือไม่ก้อง

1.3.2. ลักษณะของลมที่ผ่านเส้นเสียงขึ้นมาแล้วจะออกไปทางไหนอย่างไร

1.3.3. ตำแหน่งในช่องปากซึ่งทำให้ลมมีลักษณะตามข้อ 2.

2.4.เสียงพยัญชนะภาษาไทย

เสียงพยัญชนะในภาษาไทยมีลักษณะแตกต่างกันๆแบบเป็นหน่วยเสียงทั้งหมด 25 หน่วย ดังนี้คือ

2.4.1.พยัญชนะระเบิด(Plosive) คือ เสียงพยัญชนะที่เกิดจากลมซึ่งเปล่งออกมาแล้วถูกกักที่ใดที่หนึ่งในช่องปากชั่วขณะใดขณะหนึ่งแล้วเปิดช่องที่กักนั้นให้ลมพุ่งออกมาโดยแรง เสียงระเบิดนี้ถ้ามีลมหาย ใจเพิ่มขึ้นมาอีกก็จะเรียกว่า เสียงระเบิดที่มีลม(Aspirated Plosive) ส่วนเสียงระเบิดธรรมดาคือเสียงระเบิดซึ่งไม่มีลมหายใจเพิ่มมาอีกก็จะเรียกว่าเสียงระเบิดไม่มี(Unaspirated Plosive) เช่น "ผ้า" และ "บวช" ตามลำดับ

2.4.2.พยัญชนะกัก(Stop) คือ เสียงพยัญชนะที่เกิดในระยะต้นเหมือนกับพยัญชนะระเบิดแต่ลมที่ถูกกักนั้นมิได้ระเบิดออกมาเพราะช่องที่กักลมไม่เปิดออก ลมจึงอาจถูกกลืนกลับลงไปใหม่หรืออาจจะกลายเป็นลมหายใจออกธรรมดาไปหรือออกทางจมูกก็ได้เช่นคำว่า "อัด, ลัพธ์, นพ, สุข" เป็นต้น

2.4.3.พยัญชนะนาสิก(Nasal) คือ คล้ายกับการเปล่งเสียงพยัญชนะกักแต่เมื่อให้ลมมากักอยู่ในช่องปากแล้วก็ลดลิ้นไก่ลงเปิดช่องจมูกให้ลมออกไปทางจมูกได้เช่น พยัญชนะที่ออกเสียงในแม่ กม, กน, กง

2.4.4.พยัญชนะข้าง(Lateral) คือ เสียงพยัญชนะก้อง ซึ่งเปล่งเสียงโดยใช้ลิ้นปิดบริเวณปุ่มเหงือกตรงส่วนกลางปากไว้ปล่อยให้ลมออกทางข้างๆลิ้นเช่น คำว่า "เล่น, เลื่อย, จุฬา, หลาว" เป็นต้น

2.4.5.พยัญชนะรัว(Rolled) เสียงรัวเป็นเสียงพยัญชนะชนะก้อง ในขณะที่เปล่งเสียงต้องทำปลาย ลิ้นให้มีลักษณะอ่อนตัวที่สุดจนสามารถกระดกขึ้นไปแตะหลังฟันได้หลายๆครั้ง เช่น "ร่าย, ราม, หรูหรา" เป็นต้น

2.4.6.พยัญชนะเสียดแทรก(Fricative) คือ เสียงพยัญชนะซึ่งเกิดขึ้นเมื่อลมที่ทำให้เกิดเสียงซู่ซ่าขึ้นเสียงพยัญชนะเสียดแทรก ในภาษาไทยเป็นเสียงไม่ก้องทั้งหมด เช่น พยัญชนะที่เกิดจากริมฝีปากได้แก่ ฟ, ฝ (ไฟ, ฟ้อง, ฝา, ไฝ) ที่เกิดตรงฟันได้แก่ ซ, ศ, ษ, ส, ศ, ศร, สร (ซุง ศาล เสือ ศรี สร้าง) ที่เกิดจากช่องระหว่างเส้นเสียงได้แก่ ฮ ห (เฮฮา ห้าง)

2.4.7.พยัญชนะครึ่งสระ(Semi-Vowel) คือ เสียงเลื่อนที่เกิดขึ้นระหว่างเสียงสระ 2 เสียงตำแหน่งที่เกิดของพยัญชนะครึ่งสระทั้งสองให้ถือตามตำแหน่งของสระต้นเสียง เช่นคำว่า "วัน วาน หวาน ยา ญาติ วัย ใจ ไป" เป็นต้น

2.5.ระบบเสียง

เสียงที่ออกมาจากปากเป็นคำๆนั้นแหล่งกำเนิดมาจากหลอดเสียง(Vocal Tube) โดยปกติแล้วผู้ชายจะมีหลอดเสียงยาวประมาณ 17 ซม. ส่วนพื้นที่หน้าตัดไม่แน่นอนมีขนาดตั้งแต่ 0 ถึง 20 ตารางเซนติเมตร องค์ประกอบของจมูกเป็นสิ่งที่ช่วยในด้านคำพูดเสียงพูดของมมนุษย์ออกมาได้ 2 ทางคือ

2.5.1.จมูก เริ่มต้นด้วยลมจะเข้าสู่โพรงจมูกโดยผ่านลิ้นไก่(Velum) ของจมูกและจะมาสิ้นสุดที่ปลายจมูก

2.5.2.ปาก กรณีนี้ ลิ้นไก่(Velum) จะถูกดันขึ้นและจะมีผลทำให้ปิดช่องทางเข้าสู่โพรงจมูกดังนั้นลมส่วนใหญ่จะออกทางปาก ในบางครั้งเสียงของคำบางคำจะออกทั้งทางโพรงจมูกและปากในขณะเดียวกันทั้งนี้ขึ้นอยู่กับคุณสมบัติของคำนั้นๆ

2.6.อวัยวะที่ทำหน้าที่ในการออกเสียง

อวัยวะที่ใช้ในการออกเสียงมีหลายส่วน สามารถ ทำให้เสียงพูดแตกต่างกันไปดังจะกล่าวต่อไปนี้

2.6.1ริมฝีปาก เป็นอวัยวะส่วนที่เคลื่อนไหวได้มากในการออกเสียงพูดและมีอิทธิพลต่อการออกเสียงพูดและการทำเสียงให้แตกไปทั้งสิ้น

2.6.2.ฟัน เป็นอวัยวะที่เกิดของเสียงหลายชนิด เช่น ฟันบนกดริมฝีปากล่างหรือกดกับฝันล่างลมที่ผ่านออกมาโดยแรงจะเป็นเสียงชนิดที่เรียกว่า เสียงแทรก เกิดที่บริเวณฟันกับริมฝีปากข้าง นอกจากนี้เนื่องจากปลายลิ้นอยู่ใกล้ฟัน ปลายลิ้นจึงทำให้เกิดเสียงที่เรียกว่า ทันตะ (Dental Sound)

2.6.3. ปุ่มเหงือก เป็นส่วนนูนออกมาอยู่หลังฟันด้านบน จะทำให้เกิดเสียงปุ่มเหงือก(Alveolar Sound)เป็นตำแหน่งสำคัญอีกตำแหน่งหนึ่งในการอธิบายที่เกิดเสียง

2.6.4. เพดานแข็ง(Hard Palate)คือ เฉพาะส่วนเพดานที่โค้งเป็นกระดูกแข็งอยู่เป็นตำแหน่งสำคัญอีกตำแหน่งหนึ่งในการอธิบายที่เกิดเสียง

2.6.5. เพดานอ่อน คือ ส่วนของเพดานที่อยู่ต่อเพดานแข็งไปข้างในมีลักษณะเป็นกระดูกอ่อนที่ขยับขึ้นลงได้ เวลาพูดส่วนใหญ่ปลายเพดานอ่อนและลิ้นไก่จะถูกยกขึ้นลงได้ เวลาพูดส่วนใหญ่ปลายเพดานอ่อนและลิ้นไก่จะถูกยกขึ้นไปจดกับหลังคอนอกจากเวลาออกเสียงนาสิกเท่านั้น

บทที่ 3
การแปลงฟูเรียร์(Fourier Transform)

การแปลงฟูเรียร์เป็นเครื่องมือวิเคราะห์ในงานวิทยาศาสตร์สาขาต่างๆในปัจจุบันการประยุกต์ใช้งานที่นิยมใช้วิเคราะห์ระบบเชิงเส้นที่ไม่มีการแปรผันตามเวลา(linear time-invariant system) แต่การแปลงฟูเรียร์ที่กล่าวต่อไปนี้เป็นหลักหรือคุณสมบัติสำคัญซึ่งสามารถใช้วิเคราะห์งานได้ทั่วไป

3.1.ฟูเรียร์อินทิกรัล

ฟูเรียร์อินทิกรัลมีคำจำกัดความเป็นสมการว่า

$$H(f) = \int h(t)e^{j2\pi ft}\,dt \qquad \text{.....(3.1)}$$

h(t) เป็นเทอมของฟังก์ชันของตัวแปรเวลาและ H(f) เป็นเทอมของฟังก์ชันของตัวแปรความถี่การแปลงฟูเรียร์ของฟังก์ชันเวลาจะแทนด้วยดัวอักษรตัวใหญ่ ในรูปทั่วไปการแปลงฟูเรียร์มีรูปเป็นจำนวนเชิงซ้อน(complex number)

$$H(f) = R(f) + jI(f) = |H(f)|\, e^{j\theta(f)} \qquad \text{.....(3.2)}$$

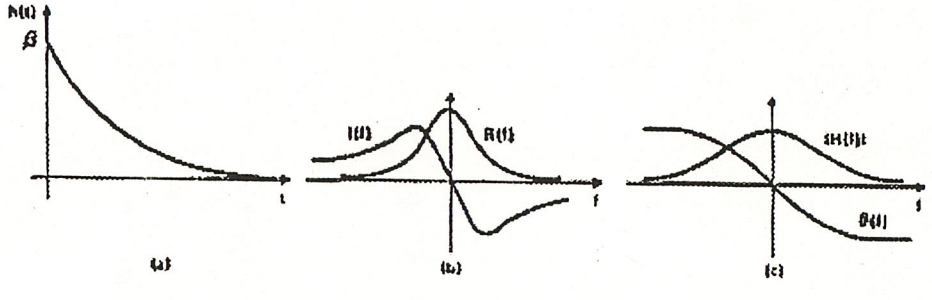
เมื่อ R(f) เป็นส่วนจริงของการแปลงฟูเรียร์

I(f) เป็นส่วนจินตภาพของการแปลงฟูเรียร์

H(f) เป็นแอมพลิจูดหรือฟูเรียร์สเปคตรัมของ h(t) ซึ่งได้จาก

$$\sqrt{R(f)^2 + I(f)2} \qquad \text{.....(3.3)}$$

(t) เป็นมุมเฟสของการแปลงฟูเรียร์ซึ่งได้ $\tan^{-1}[\, I(f)/R(f)\,]$



รูปที่ 3.1 (a) ตัวอย่างของฟังก์ชันในโดเมนเวลา

(b) ส่วนจริงและส่วนจินตภาพของการแปลงฟูเรียร์

(c) แอมพลิจูดและเฟสของการแปลงฟูเรียร์

3.2.การแปลงอินเวอร์สฟูเรียร์(inverse fourier transform)

การแปลงอินเวอร์สฟูเรียร์มีคำจำกัดความดังนี้

$$h(t) = \int H(f)\, e^{j2\pi ft}\, df \qquad \text{.....(3.4)}$$

3.3.การแปลงฟูเรียร์แบบไม่ต่อเนื่อง(discrete fourier transform หรือ DFT)

DFT เป็นลำดับของความถี่ไม่ต่อเนื่องที่ได้จากการสุ่มในหนึ่งคาบเวลาของการแปลงฟูเรียร์ เมื่อกำหนดให้จำนวนการสุ่มเท่ากับ N สุ่มบนคาบเวลา $0 < w < 2\pi$ ดังนั้น

$$w_k = 2\pi k/N \qquad \text{เมื่อ } 0 < k < N-1 \qquad \text{.....(3.5)}$$

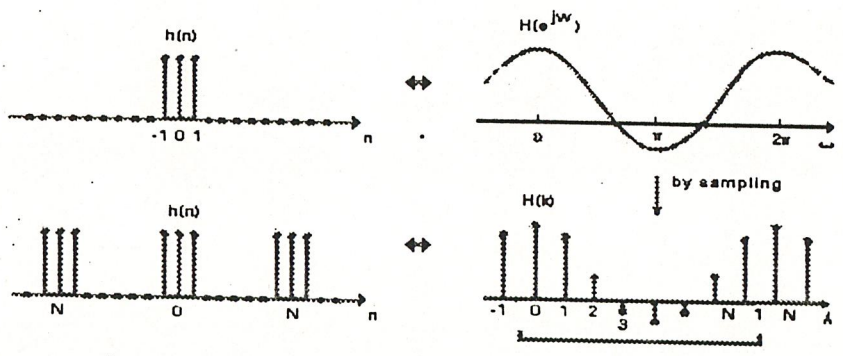
ถ้ากำหนดให้ {h(n)}เป็นลำดับของ discrete time และลำดับของ Fourier Transform $H(e^{j\omega})$ ให้เท่ากับ { H(k) } โดยที่

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

.....(3.6)

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

เราจะเห็นว่า $H(e^{j\omega})$ มีคาบเวลาเท่ากับ w หรือ 2π ซึ่งได้จากการกระจายอนุกรมฟูเรียร์และคำนวณค่าสัมประสิทธิ์จากลำดับ{ h(n) } แสดงว่าเมื่อสัญญาณเวลาต่อเนื่องถูกสุ่มด้วยคาบเวลาการสุ่ม T_s สเปคตรัมของผลลัพธ์จากการแปลงฟูเรียร์จะมีฟังก์ชันคาบเวลาของความถี่เท่ากับ $2\pi/Ts$ จากรูปที่ 1 เมื่อ $H(e^{j\omega})$ ถูกสุ่มด้วยคาบเวลาการสุ่ม $w_s = 2\pi/N$ เมื่อแปลงกลับเป็นลำดับของฟังก์ชันเวลาไม่ต่อเนื่อง { h(n) } จะมีคาบเวลา $2\pi/w_s = N$



รูปที่ 3.2 แสดงผลลัพธ์ในลำดับเวลาจาก sampled spectrum

ในกรณี N=8 และอยู่ในช่วง $0 < w < 2\pi$

ดังนั้นลำดับ discrete time ที่อยู่ในฟังก์ชันคาบสามารถหาในเทอมของ{ x(n) } ได้

$$h(n) = \sum h(n+mN) \qquad \text{.....(3.7)}$$

ลำดับ{ h(n) } เรียกว่า periodic extention ของ{ h(n) } จำนวนจุดสุ่มในหนึ่งคาบของ spectrum N จะมีค่าเท่ากับคาบของ { h(n) }

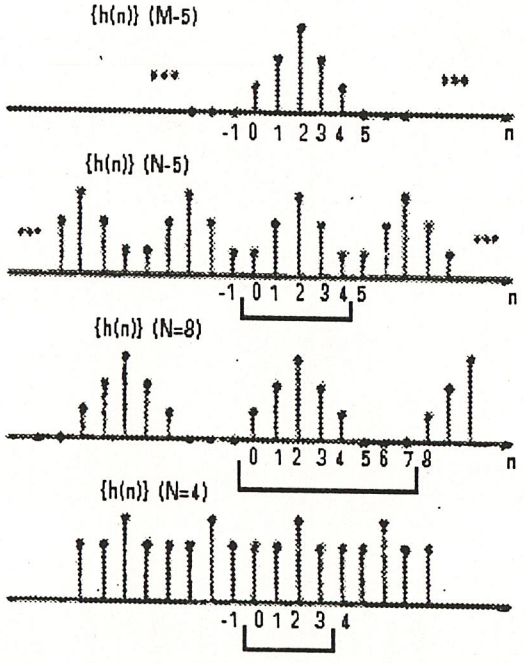
เราสามารถเลือกค่า N เท่าใดก็ได้ซึ่งเป็นจำนวนการสุ่มของ $H(e^{j\omega})$ ภายใต้ $0 <= \omega < 2$ เมื่อ N เป็นคาบของ{ h(n)} เราต้องเลือกค่าที่ไม่น้อยเกินไปในลักษณะที่{ h(n)} เป็นฟังก์ชันคาบเวลาแบบไม่จำกัดทำให้เกิดลักษณะสัญญาณที่ถูกสุ่มทับกันสัญญาณที่ได้จะผิดพลาดไปสามารถแสดงจากตัวอย่างข้างล่าง

กำหนดให้

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

$$= 0 \qquad \text{เมื่อ } M <= n <= N-1 \qquad \text{.....(3.8)}$$

จะได้จุดที่เกิดจากการสุ่มเมื่อ N มีค่าต่างๆ ดังรูป



รูปที่ 3.3 แสดงความสัมพันธ์ของช่วงเวลาของลำดับ M และจำนวนจุดในการสุ่มสเปคตรัม N

จากรูปจะเห็นว่าถ้าค่า M > N รูปสัญญาณจะทับกัน(overlap) หรือเกิด time-aliasing ซึ่งสามารถป้องกันได้โดยเลือกค่า N >= M

3.4.การแปลงฟาสท์ฟูเรียร์(Fast fourier Transform หรือ FFT)

เนื่องจากปัญหาที่เกิดขึ้นกับการแปลงฟูเรียร์เต็มหน่วย (dicrete fourier transform หรือ DFT) คือ ในการคำนวณนั้นมีการใช้จำนวนลำดับข้อมูลมากซึ่งทำให้การคำนวณของคอมพิวเตอร์ใช้เวลามากตามไปด้วยเช่น การคำนวณ DFT สำหรับลำดับสัญญาณเข้ายาว N ลำดับหรือ จุดนั้น คอมพิวเตอร์ต้องใช้การคำนวณจำนวนเชิงซ้อนถึง NxN ครั้งและการบวกจำนวนเชิงซ้อนอีก N(N-1) ครั้ง ซึ่งคอมพิวเตอร์ทั่วไปแล้วไม่มีคำสั่งภาษาเครื่องที่ใช้ในการคูณตัวเลข จึงต้องงเขียนโปรแกรมย่อยหรืออเพิ่มเติมวงจรคูณโดยเฉพาะเข้าไป ส่วนการบวกตัวเลขของคอมพิวเตอร์นั้นทำได้ง่ายและรวดเร็วจึงสามารถกล่าวได้ว่ากระบวนการคูณตัวเลขนั้นใช้เวลาในการคำนวณมากกว่าการบวกตัวเลขมาก จากผลอันนี้ทำเห็นได้ชัดว่า ความเร็วในการคำนวณ DFT จึงขึ้นอยู่กับความเร็วและจำนวนครั้งในการคูณตัวเลขเป็นสำคัญ ดังนั้น J.W. Cooley และ J.W. Turkey จึงได้พัฒนาลำดับการหรืออขั้นตอนวิธีในการคำนวณ DFT ให้รวดเร็วขึ้นซึ่งเรียกว่า การแปลงฟาสท์ฟูเรียร์ ซึ่งการคำนวณโดยใช้ FFT จะใช้การคูณจำนวนเชิงซ้อนเพียง $N \log_2 N$ ครั้งเท่านั้นหรืออจำนวนครั้งในการคูณลดลงไปถึง $N/(\log_2 N)$ เท่า และผลดีอีกประการหนึ่งคือทำให้การสร้างวงจรเฉพาะเพื่อการคำนวณเฉพาะเพื่อการคำนวณ DFT ทำได้ง่ายและคำนวณได้เร็วขึ้น ถึงแม้ว่า FFT จะมีชื่อเรียกว่าการแปลงฟาสท์ฟูเรียร์ แต่ตัว FFT เองนั้นไม่ใช่การแปลงฟูเรียร์แท้จริงแต่เป็นวิธีการหรือลำดับการในการคำนวณที่ช่วยในการคำนวณ DFT ซึ่งเป็นการแปลงฟูเรียร์ได้รวดเร็วขึ้น ในปัจจุบันมีการดัดแปลง และคิดค้นเสนอผลงานเกี่ยวกับ FFT มากมายหลายแบบซึ่งแต่ละแบบมีทั้งข้อดีและข้อเสียต่างกันออกไปโดยทั่วไปแบ่งออกได้เป็น 2 ชนิดใหญ่คือ ชนิดลดทอนทางเวลา(decimation in time หรืออ DIT) และ ชนิดลดทอนทางด้านความถี่(decimation in frequency หรือ DIF) ทั้งสองชนิดนี้โดยหลักการแล้วมีความคล้ายคลึงกันดังนั้นจึงจะขออธิบายแค่วิธีเดียวคือ DIT

3.4.1.การแปลงฟาสท์ฟูเรียร์แบบฐานสอง(Radix 2 FFT)

3.4..1.1.หลักการเบื้องต้นของ FFT

มีหลักการนิยามมาจากการแปลงฟูเรียร์แบบเต็มหน่วย(dft)โดยมีการนิยามดังนี้

$$X(k) = \sum_{m=0}^{N-1} x(m).\ W^{mk} \qquad \text{.....(3.9)}$$

โดยดรรชนี k,m = 0,1,...,N-1 และจำนวนเชิงซ้อน W = exp(-j2π/N) โดยที่ลำดับ x(m) มักจะเป็นสัญญาณเกี่ยวกับในโดเมนเวลา ส่วน X(k) มักเกี่ยวข้องกับสัญญาณในโดเมนความถี่ หรืออเรียกว่าสเปคตรัมของสัญญาณโดยในที่นี้ในสมการได้ยกเว้นการเติม 1/N ไว้เพื่อที่จะให้สะดวกและง่ายต่อการอธิบายสำหรับหลักสำคัญประการหนึ่งของ FFT ที่ลดจำนวนครั้งในการคูณเลขจำนวนเชิงซ้อนโดยอาศัยคุณสมบัติ ความเป็นคาบ ของจำนวนเชิงซ้อน W คือ

$W^{mk} = W^{[\,mk \bmod (N)]}$

3.4.1.2.ขั้นตอนการลดทอนทางเวลา(Decimation-In- Time หรือ DIT)

เป็นวิธีในการแบ่งกลุ่มลำดับของสัญญาณในโดเมนเวลา x(m) ที่มีขนาด N จุดออกเป็นสองลำดับสัญญาณที่มีความยาว N/2 จุดเท่ากัน ทั้งสองลำดับนี้ให้ชื่อว่า ลำดับสัญญาณคู่และลำดับสัญญาณคี่ โดยที่ลำดับสัญญาณคู่เกิดจากการเอาลำดับในตำแหน่งเลขคู่มาเรียงกันโดยส่วนที่เหลือเป็นลำดับสัญญาณคี่ ถ้าเรานิยามให้ $x_E(m)$ เป็นลำดับคู่ และลำดับคี่เป็น $x_O(m)$ ตามลำดับเพราะฉะนั้น

$$x_E = x(2m) \quad ; \quad m = 0,1,...,(N/2)-1$$

$$x_O = x(2m+1) \quad ; \quad m = 0,1,...,(N/2)-1 \qquad \text{.....(3.10)}$$

ถ้าให้ W_N แทน W ของลำดับยาว N จุด จะทำให้การคำนวณการแปลง DFT ของลำดับ x(m) ที่ยาว N จุด เขียนใหม่ได้เป็น

$$X(k) = \sum_{m=0}^{N-1} x_E(m)\,(W_N)^{km} + \sum_{m=0}^{N-1} x_O(m)\,(W_N)^{km}$$

$$X(k) = \sum_{m=0}^{(N/2)-1} x_E(m)\,(W_N)^{km} + \sum_{m=0}^{(N/2)-1} x_O(m)\,(W_N)^{km} \quad \text{.....(3.11)}$$

ให้พจน์ $(W_N)^2$ จะได้ว่า

$$(W_N)^2 = \{\exp(j2\pi/N)\}^2 = \exp(j2\pi/N/2) = W_{N/2}$$

ซึ่งเมื่อจัดพจน์ใหม่จาก $W_{N/2}$ หรืออ W ของลำดับยาว N/2 จุด จะได้

$$X(k) = \sum_{m=0}^{(N/2)-1} x_E(m)\,(W_{N/2})^{km} + (W_N)^k \sum_{m=0}^{(N/2)-1} x_O(m)\,(W_{N/2})^{km}$$

$$X(k) = X_1(k) + (W_N)^k\, X_2(k) \quad \text{.....(3.12)}$$

โดยที่ $X_1(k)$ และ $X_2(k)$ แทนการแปลง DFT ขนาด N/2 จุดของลำดับ $x_E(m)$ ตามลำดับ สมการที่ (3.12) แสดงให้เห็นว่าการคำนวณ DFT ขนาด N จุด นั้นสามารถแบ่งออกเป็นการคำนวณ DFT ขนาด N/2 จุดสองอันดับได้และข้อสำคัญก็คือ การคูณจำนวนเชิงซ้อนจะลดลงเหลือ $2(N/2)^2 = N^2/2$ ครั้ง ซึ่งจะเห็นว่าการลดเวลาการคำนวณลงไปได้ถึง 50 % โดยอาศัยหลักการเดียวกันถ้าเราแบ่งทอนลำดับ $x_E(m)$ และ $x_O(m)$ ออกเป็นลำดับคู่ลำดับคี่ลงไปอีกตามลำดับจนในที่สุดเหลือเป็นลำดับขนาด 2 จุดหรืออกล่าวได้ว่า การคำนวณการแปลง DFT ขนาด N จุดทำได้โดยการแปลง DFT ขนาดสองจุดจำนวน N/2 ตอน ด้วยกัน ซึ่งในการแบ่งหรืออซอยลำดับ x(n) ออกทีละครึ่งจนเหลือการคำนวณ DFT ขนาด 2 จุดนี้สำหรับสัญญาณขนาด N ลำดับจะทำให้การแบ่งออกเป็น $\log_2 N$ ดังแสดงตามรูปที่ 3.5

หลังจากที่มีการแบ่งย่อยก็ต้องมีการรวมประกอบคืนเหมือนเดิมเพื่อให้ได้เป็นการคำนวณ DFT ขนาด N จุดซึ่งการนำมาประกอบคืนจะต้องมีหลักเกณฑ์ที่แน่นอนไม่เช่นนั้นจะทำให้ผลการคำนวณ DFT ที่ได้มีค่าผิดพลาด เพื่อให้การประกอบกันของลำดับสัญญาณ N ลำดับเป็นไปอย่างถูกต้องจึงต้องทำการนิยามค่าของสมการ (3.12) สำหรับค่า k>N/2 ดังนี้

ในการคำนวณ DFT นั้นลำดับสัญญาณเข้า x(n) จะมีการสลับตำแหน่งหรือสลับลำดับอย่างมีกฎเกณฑ์ที่แน่นอนซึ่งเรียกวิธีการนี้ว่า การผันกลับบิท นั้นคือการแทนดรรชนี n ของลำดับ x(n) ด้วยเลขฐานสองโดยจำนวนของเลขฐานสองต้องเพียงพอที่จะแทนค่า N ได้เช่น ในกรณี N=8 ต้องแทนด้วยเลขฐานสอง 3 บิท จากนั้นการจัดลำดับ x(n) ใหม่จะได้จากการผันบิทของเลขฐานสองที่แทนด้วยดรรชนี n ดังในรูปที่ 3.3 คือ x(001) จะถูกแทนด้วย x(100) และ x(110) จะแทนด้วย x(011) เป็นต้นดังรูปที่ 3.6

โดยวิธีที่อธิบายอาจกล่าวโดยย่อได้ว่าเป็นการคำนวณ DFT ขนาด N จุดเดิมโดยถูกแบ่งออกเป็น DFT ขนาด N/2 จุดจำนวน 2 ตอนแล้วนำมารวมกันโดยใช้ตัวประกอบหมุนและลำดับนี้จะทำจนกระทั่งผลสุดท้ายเป็นการแปลง DFT ขนาด 2 จุดตามรูปที่ 3.7

3.5.การประยุกต์ FFT

3.5.1.ความละเอียด(resulotion) ของ FFT

ผลลัพธ์ของ FFT ในรูปที่ 3.8(b) และ (c) แต่ละความถี่มีระยะห่าง f = 1/NT ดังนั้น จุดสุ่มในโดเมนความถี่จึ่งมีค่าตั้งแต่ 0/NT,1/NT, 2/NT,...,(N/2)/NT สำหรับด้านความถี่บวก ระยะห่างของความถี่ f = 1/NT เป็นเทอมที่แสดงถึงความละเอียดของ FFT แต่ละสมาชิกของความถี่ที่ได้เรียกว่า สมาชิกความละเอียด(resolution element) หรืออเซลความละเอียด(resolution cell) เราอาจจะคิดว่า ความละเอียดเป็นเทอมที่ทำให้เราสามารถแยกความแตกต่างของแต่ละความถี่โดยประมาณค่าความถี่ที่ได้จากการแปลงเป็นความถี่ตั้งแต่ 0/NT,1/NT, 2/NT,...,(N/2)/NT ซึ่งเราอาจจะลดระยะห่างระหว่างแต่ละความถี่ได้โดยการเพิ่มจำนวนข้อมูล N ซึ่งก็คือการเพิ่มระยะการตัดทอน(truncation) ของฟังก์ชันที่ต้องการแปลงถ้าค่าของ N เพิ่มขึ้นเป็นสองเท่าระยะห่างของความถี่ก็ลดลงสองเท่าเช่นกัน

จากรูปที่ 3.9 ระยะห่างของความถี่(ความละเอียด) ของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ถูกกำหนดจากความกว้างของสี่เหลี่ยมที่ไปคูณหรืออตัดทอนฟังก์ชันก่อนทำการแปลง การตัดทอนในโดเมนเวลาก็คือการทำคอนโวลูชั่นของฟังก์ชัน [sin(f)/f] กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้นการทำคอนโวลูชันกับฟังก์ชัน [sin(f)/f] ทำให้ผลการแปลงคลุมเครือและเลือนลางถ้าฟังก์ชันที่ตัดทอนในโดเมนเวลามีความกว้างมากขึ้นจะทำให้ผลของฟังก์ชัน [sin(f)/f] แคบลงและความคลุมเครือของความถี่ลดน้อยลงด้วย ความคลุมเครือของความถี่ยิ่งน้อยเท่าใดก็ยิ่งทำให้ประสิทธิภาพของการใช้ FFT ในการแก้ปัญหาในแต่ละงานเป็นไปได้มากยิ่งขึ้น

ความเข้าใจผิดทั่วๆไปของผู้ใช้ FFT คือการเพิ่มจำนวนข้อมูล N โดยการต่อเติมศูนย์และตัดทอนฟังก์ชันและตีความหมายของฟังก์ชันความถี่ที่ได้เป็นฟังก์ชันความถี่ที่มีความละเอียดสูง ตอนนี้เรามาเปรียบเทียบรูปที่ 3.10 กับรูปที่ 3.11 ในรูปที่ 3.11(a) เป็นการแสดงผลการแปลงดิสครีตฟูเรียร์เช่นเดียวกับรูปที่ 3.10(g) เราต้องการตรวจสอบผลของการเพิ่มศูนย์ให้กับฟังก์ชันเวลาของรูปที่ 3.11(a) สมมติว่าจำนวนของศูนย์ที่เพิ่มเข้าไปนั้นเท่ากับ N ซึ่งทำได้โดยการคูณด้วยฟังก์ชันเวลาในโดเมนเวลา

ในรูปที่ 3.11(b) ผลของความถี่ที่ได้แสดงในรูปถัดมาซึ่งในการคูณทำให้ฟังก์ชันมีคาบเวลาเท่ากับ 2N โดยจุดที่ไม่ใช่ศูนย์เท่ากับจุดสุ่ม N จำนวนในรูปที่ 3.11(a) ผลของฟังก์ชันความถี่ที่ได้ก็คือการทำคอนโวลูชั่นของฟังก์ชันความถี่ของรูปที่ 3.11(a) และรูปที่ 3.11(b) แต่ความละเอียดของความถี่ที่ได้ถูกกำหนดไว้ในรูปที่ 3.11(a) เรียบร้อยแล้ว การทำคอนโวลูชั่นเป็นการเพิ่มจำนวนจุดสุ่มในโดเมนความถี่โดยการเฉลี่ยค่า (Interplation) ด้วยฟังก์ชัน [sin(f)/f] กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้นดังนั้นถึงแม้ว่าระยะห่างของผลลัพธ์จาก FFT จะมีระยะใกล้ขึ้นโดยการต่อเติมศูนย์ก็ตามแต่ความละเอียดก็ไม่เปลี่ยนแปลง ความละเอียดของ FFT ไม่สามารถทำให้ความละเอียดของข้อมูลเพิ่มขึ้นโดยการต่อเติมศูนย์นอกเสียจากว่า ฟังก์ชันที่จะทำการต่อเติมศูนย์มีค่าศูนย์ครอบคลุมช่วงนั้นอยู่แล้ว

จากหัวข้อนี้จึงสรุปได้ว่า ความละเอียดนั้นต้องพิจารณาจากช่วงของเวลาในโดเมนเวลาในสัญญาณและในการประยุกต์ใช้ FFT ช่วงเวลาของสัญญาณถูกตั้งใหม่เป็นช่วงระยะห่างของการตัดทอนข้อมูล

3.5.2. ผลของความผิดเพี้ยนจาก FFT(FFT aliasing)

ปัญหาหนึ่งที่เราพบในการคำนวณการแปลงฟูเรียร์จาก FFT คือความผิดเพี้ยน(aliasing) จากที่เคยกล่าวมาแล้วในการสุ่มข้อมูลความผิดเพี้ยนจะเกิดขึ้นเมื่อจุดสุ่มของฟังก์ชันเวลามีระยะห่างของการเก็บมากเกินไปผลของฟังก์ชันความถี่ที่ผิดเพี้ยนนี้เรียกว่า การพับ(fold)หรืออการซ้อน(overlap) ในตัวมันเองซึ่งแสดงตัวอย่างไว้ในรูปที่ 3.12

ในรูปที่ 3.12 เรามีจุดสุ่มของฟังก์ชัน h(t) = e , t > 0 โดยช่วงของเวลาการสุ่ม T = 1.0,0.5 และ 0.25 ตามลำดับ จำนวนข้อมูลตั้งให้เท่ากับ 32 สำหรับแต่ละกรณีขนาดของการแปลงฟูเรียร์ที่ได้คำนวณโดยใช้ FFT ซึ่งได้แสดงรวมไว้ในรูปที่ 3.12(a) ถึง 3.12(c) แล้วจากผลที่ได้จาก FFT จะเห็นว่าเมื่อ T= 1.0 ผลที่ได้มีความผิดเพี้ยนมากที่สุด(ขนาดของการแปลงฟูเรียร์แบบต่อเนื่องได้แสดงไว้ในรูปที่ 3.12(d)) จะเห็นว่าความผิดเพี้ยนมีค่าลดลงเมื่อมีคาบเวลาเท่ากับ 0.5 และเมื่อคาบเวลาเท่ากับ 0.25 ผลที่ได้จะมีความคล้ายคลึงกับผลที่ได้จากการคำนวณแบบต่อเนื่องมากที่สุด

ในรูปที่ 3.12 ได้แสดงหลักการลดความผิดเพี้ยนโดยการลดช่วงของการสุ่มในกรณีนี้จะเห็นว่าไม่มีผลต่อการตัดทอนเนื่องจาก T มีค่าต่ำจึงทำให้ NT มีค่ามากกว่าความกว้างของช่วงไม่เป็นศูนย์ (nonzero interval) ของฟังก์ชันมาก

3.5.3. การตัดทอนของ FFT ในโดเมนเวลา(FFT Time-Domain Truncation)

ปัญหาความผิดพลาดอื่นที่มักพบจากการประยุกต์ใช้ FFT ในการแปลงฟูเรียร์คือความผิดพลาดเนื่องจากการตัดทอนข้อมูลในโดเมนเวลา(time-domain truncation) ความผิดพลาดที่เกิดขึ้นจากการเลือกกลุ่มข้อมูลจุดสุ่มเพื่อหาลักษณะของฟังก์ชันเวลาไปตัดทอนลักษณะบางส่วนของรูปคลื่นเริ่มต้น รูปที่ 3.13(a) และรูปที่ 3.13(b) แสดงตัวอย่างของจุดนี้เมื่อเราตัดทอนฟังก์ชัน h(t) ที่ NT = 1,2,5 second ตามลำดับขนาดของการแปลงฟูเรียร์ที่ผ่าน FFT ได้แสดงในรูป

การตัดทอนที่ 1 วินาทีทำให้ผลที่ได้จาก FFT มีการกระเพื่อม(rippling) เมื่อใช้ช่วงเวลาของการตัดทอนนานขึ้นเท่ากับ 2 วินาที่ผลของ FFT ที่ได้มีการกระเพื่อมน้อยลงและเมื่อเพิ่มช่วงเวลาการตัดทอนให้เป็น 5 วินาทีผลของ FFT ที่ได้ไม่ปรากฏผลของการเกิดการกระเพื่อมเหมือนกับรูปที่ถูกต้องในรูปที่ 3.13(c) ส่วนในรูปที่ 3.13 แสดงผลของการทดลองโดยการพิจารณาหาช่วงเวลาการตัดทอนที่ดีที่สุดโดยการเพิ่มช่วงเวลาการตัดทอนซึ่งเราก็ได้เห็นผลของการลดลงของการกระเพื่อม

3.5.4.FFT ของฟังก์ชันคาบเวลา

การคำนวณ FFT ของฟังก์ชันคาบเวลาเราจะต้องเลือกช่วงการสุ่มและช่วงของการตัดทอน จากที่กล่าวมาแล้ว การเลือก T ต้องเลือกให้เหมาะสมเพื่อไม่ให้เกิดความผิดเพี้ยน การตัดทอนของฟังก์ชันเวลา คาบเวลาเป็นปัญหาใหม่ที่ไม่ได้อธิบายไว้ในหัวข้ออื่นมาก่อน ถ้าเราเลือกจำนวนจุดสุ่ม N ของการแปลงฟูเรียร์เท่ากับ 1 คาบเวลาพอดีหรืออเท่ากับจำนวนเท่าของคาบเวลา ผลของความถี่ที่ได้จะตรงกับสัญญาณเริ่มต้น

เพื่ออธิบายตรงจุดนี้เราทำการคำนวณ FFT ของฟังก์ชันโคซายน์ในรูปที่ 3.14(a) โดยมีช่วงการสุ่ม T = 1.0 S และจำนวนจุดสุ่มเท่ากับ 32 และแสดงผลลัพธ์ที่ได้ในรูปที่ 3.14(b) โดยแสดงเป็นขนาดที่ได้จาก FFT ของจุดเหล่านี้จากรูปจะเห็นว่า จุดอื่นเป็นศูนย์หมดนอกจากจุดที่เป็นความถี่จริง

3.5.5.ฟังก์ชันถ่วงน้ำหนักข้อมูล(data weighting function)

จากหัวข้อที่แล้ว การตัดทอน(truncation) อาจจะทำให้ผลการแปลงฟูเรียร์คลาดเคลื่อนการประมวลผลข้อมูลจำนวนจำกัด N ของฟังก์ชันคาบเวลาที่ไม่รู้คาบเวลาแน่นอนจำเป็นต้องใช้วินโดว์ของข้อมูลหรือที่เรียกว่าฟังก์ชันถ่วงน้ำหนักข้อมูลในหัวข้อนี้เราจะอธิบายเทคนิคการใช้ฟังก์ชั่นถ่วงน้ำหนักเพื่อลดผลเสียจากการตัดทอนให้เหลือน้อยที่สุด

3.5.5.1 ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม(rectangular weighting funtcion)

จากรูปที่ 3.15 เราสุ่มสัญญาณซายน์โดยการคูณด้วยอิมพัลส์ฟังก์ชัน(impulse function) ซึ่งแสดงไว้ในรูปที่ 3.15(b) นำผลที่ได้ (รูปที่ 3.15(c)) ไปคูณด้วยฟังก์ชันตัดทอนสี่เหลี่ยมซึ่งแสดงไว้ในรูปที่ 3.15(d) เพื่อจำกัดจำนวนของจุดสุ่มให้เหลือเท่ากับ N ในที่นี้คือการตัดทอนของโดเมนเวลา(time domain) ก็คือการถ่วงน้ำหนักข้อมูลโดยคูณด้วยฟังก์ชันสี่เหลี่ยมผลของการตัดทอนโดเมนเวลาได้แสดงไว้ในรูปที่ 3.15(e) เรารู้ว่าฟังก์ชันโดเมนความถี่(frequency domain) ของฟังก์ชันสี่เหลี่ยมคือฟังก์ชัน[sin(f)/f] ดังนั้นการแปลงเป็นโดเมนความถี่ของฟังก์ชันเวลาที่ตัดทอนแล้วจึงเป็นการทำคอนโวลูชัน(convolution) ของฟังก์ชันเวลากับฟังก์ชันสี่เหลี่ยมทำให้ผลที่ได้มีองค์ประกอบของความถี่เพิ่มขึ้นมาซึ่งก็คือลอนข้าง(side lobe) ของฟังก์ชันสี่เหลี่ยมองค์ประกอบที่เพิ่มขึ้นมานี้เรียกว่า ส่วนรั่วไหล(leakage) ทำให้อิมพัลส์ของความถี่ของฟังก์ชันรั่วออกไปที่ลอนข้างของฟังก์ชัน[sin(f)]/f

ถึงแม้ว่าฟังก์ชันในโดเมนเวลาเริ่มต้นเป็นสัญญลักษณ์ซายน์แต่เมื่อผ่านการสุ่มแล้วรูปคลื่นของสัญญาณที่ถูกสุ่มในโดเมนเวลาจะไม่เป็นสัญญาณซายน์อีกเพราะว่าช่วงของการตัดทอน(truncation interval) ไม่เท่ากับคาบเวลาหรืออจำนวนเท่าของคาบเวลา(เป็นจำนวนเต็ม) ทำให้คอนโวลูชั่นในโดเมนเวลารูปที่ 3.15 (e) และ 3.15(f) ไม่ตรงกับฟังก์ชันคาบเวลาเริ่มต้นเนื่องจากการทำคอนโวลูชันของฟังก์ชันเวลาเป็นแบบไม่ต่อเนื่อง (discontinuous) จึงได้แสดงผลของการเกิดการกระเพื่อมที่เพิ่มขึ้น(rippling effect) ไว้ในแบบไม่ต่อเนื่องในรูปที่ 5.2(g) ด้วย ต่อไปเราสาธิตตัวอย่างของผลที่เกิดจากการใช้ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมดังนี้

สมมติว่าเราต้องคำนวณ FFT ของฟังก์ชันโคซายน์(cosine function) ซึ่งแสดง 5.8(a) โดยมีคาบเวลา T = 1.0 และ N = 32 ในรูปที่ 3.16(b) เราได้แสดงขนาดของการแปลง ดีสครีตฟูเรียร์(discrete fourier transform) ของจุดสุ่มในรูปที่ 3.16(a)FFT จะทำให้ผลเป็นความถี่ที่ไม่ต่อเนื่องทางด้านบวกทั้งหมดจากที่ได้อธิบายในหัวข้อที่แล้วองค์ประกอบของความถี่เพิ่มขึ้นมาที่เรียกว่าส่วนรั่วไหลเป็นผลมาจากลักษณะการมีลอนข้างของฟังก์ชัน [sin(f)]/f เพื่อลดผลจากการรั่วไหล เราจำเป็นต้องใช้วิธีการตัดทอนโดเมนเวลาหรืออ ฟังก์ชันถ่วงน้ำหนักที่มีลักษณะของลอนข้างในโดเมนของความถี่เล็กกว่าของฟังก์ชันสี่เหลี่ยมลอนข้างที่เล็กกว่านี้ทำให้การรั่วไหลของผลลัพธ์จาก FFT ลดน้อยลงเพื่อให้เข้าใจจุดนี้ได้ชัดเจนยิ่งขึ้นขอให้พิจารณารูปที่ 3.15 อีกครั้งหนึ่งถ้าเปลี่ยนฟังก์ชันตัดทอนในรูปที่ 3.15(d) เป็นฟังก์ชันที่มีลอนข้างต่ำจะทำให้การประกาศค่าของการแปลงฟูเรียร์ดีขึ้น การใช้ฟังก์ชันถ่วงน้ำหนักข้อมูลต้องทำกับจุดสุ่ม N ก่อนคำนวณ FFT

3.5.5.2. ลักษณะของฟังก์ชันถ่วงน้ำหนักชนิดต่างๆ

ในรูปที่ 3.17(a) ได้แสดงฟังก์ชันตัดทอนหรืออถ่วงน้ำหนักที่นิยมใช้กับ FFT ผลของการตอบสนองความถี่ของฟังก์ชันได้แสดงไว้แล้วในรูปที่ 3.17(b) และในตารางที่ 5.1 แสดงข้อมูลเปรียบเทียบลักษณะของฟังก์ชันถ่วงน้ำหนักแต่ละตัวทั้งในโดเมนเวลาและโดเมนความถี่โดยมีจุดศูนย์กลางอยู่ที่จุดศูนย์เพื่อให้ง่ายต่อการสังเกต

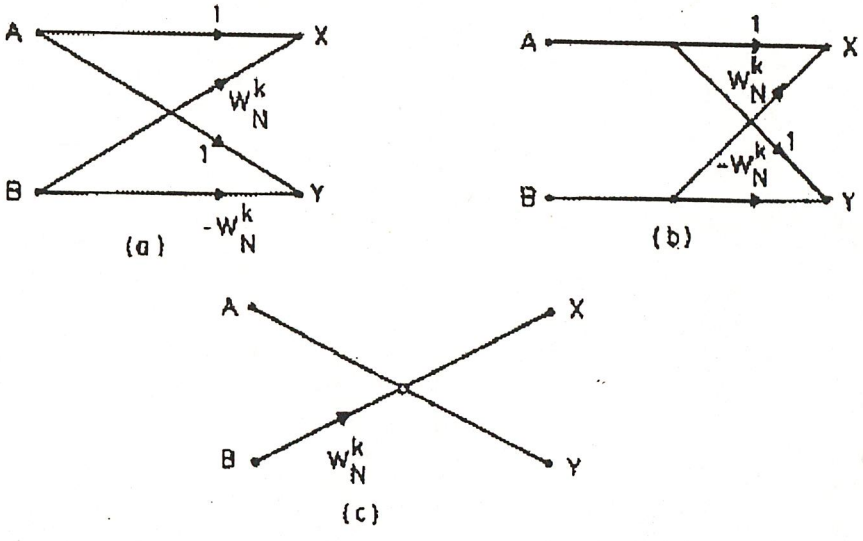
จากรูปที่ 3.17(b) จะเห็นว่าฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมทั้งหมดมีลอนข้างในโดเมนความถี่ที่มีขนาดเล็กกว่าฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมทั้งนั้นซึ่งจะช่วยให้ผลของการรั่วไหลลดน้อยลงแต่อย่างไรก็ตามฟังก์ชันถ่วงน้ำหนักทั้งหมดนี้มีลักษณะของลอนหลัก(main lobe) ที่กว้างกว่าลองดูรูปที่ 3.15(d) และรูปที่ 3.15(e) อีกครั้ง ลองคิดเปรียบเทียบผลที่ได้เมื่อใช้ฟังก์ชันถ่วงน้ำหนักตามรูปที่ 3.17(b) จะเห็นว่ายิ่งลอนหลักมีความกว้างมากขึ้นเท่าใด ผลที่ได้จาก FFT ยิ่งมีความคลุมเครือมากยิ่งขึ้นนั่นก็หมายความว่าถ้าลอนหลักมีขนาดกว้างจะทำให้แยกความแตกต่างของแต่ละความถี่ได้ยาก

ข้อได้เปรียบเสียเปรียบระหว่างส่วนรั่วไหล(ระดับของการรั่วไหล) และความละเอียด (แบนด์วิดธ์ของลอนหลัก) เป็นที่รู้จักกันดีในงานวิทยาศาสตร์สาขาต่างๆในตารางที่ 5.2 แสดงระดับสูงสุดของลอนข้างและแบนวิดธ์(bandwidth) 3-dB สำหรับแต่ละฟังก์ชันถ่วงน้ำหนัก สำหรับงานการทดลองทั่วไป นิยมใช้ฟังก์ชันแฮนนิ่ง(Hanning function) เพราะว่าโครงสร้างของมันง่ายกว่าแบบอื่นการเลือกใช้ฟังก์ชันถ่วงน้ำหนักที่ให้ผลดีที่สุดต้องขึ้นกับลักษณะของงานที่เรานำไปใช้งานด้วยเป็นสำคัญ

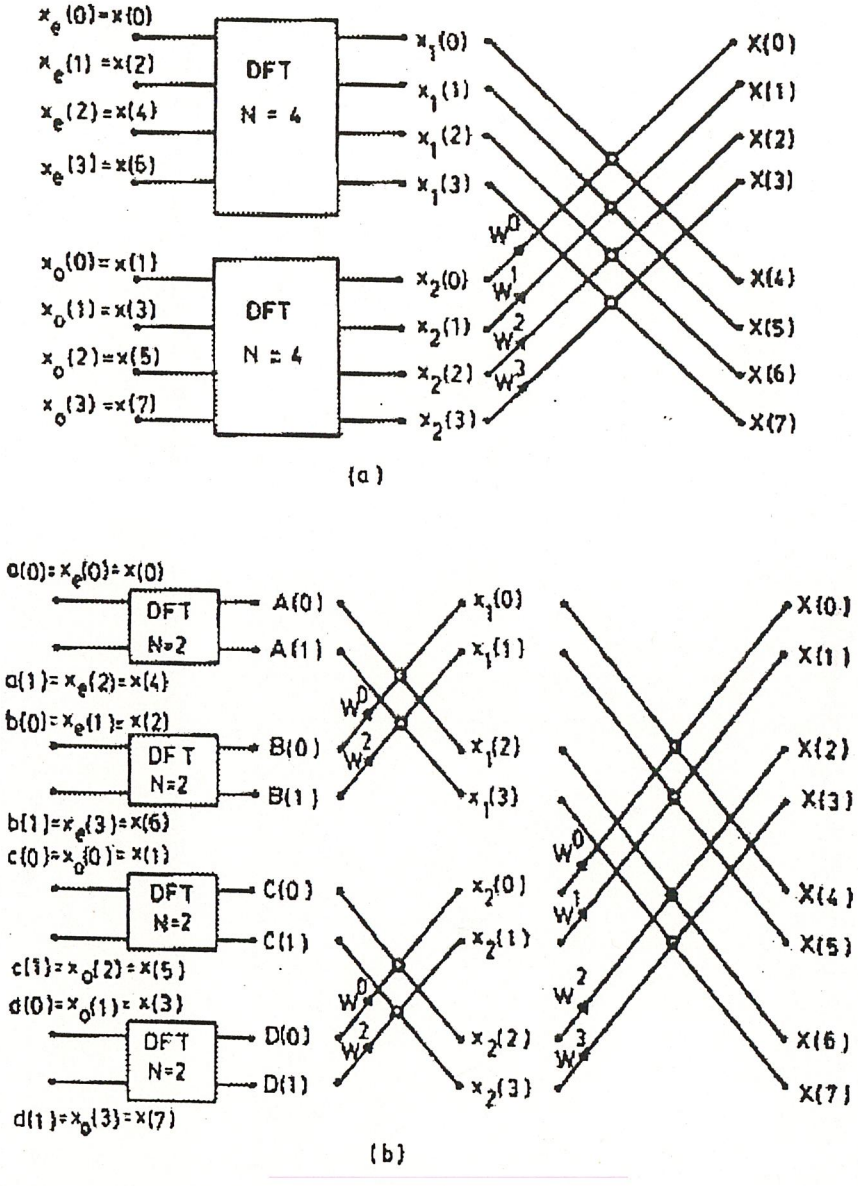
ผลที่ได้จาก FFT จะมีช่วงความถี่ $f_0 = 1/NT$ ซึ่งอาจทำให้เกิดความสับสนกับค่า 1/NT ของฟังก์ชันถ่วงน้ำหนักที่เลือกใช้ผลของความละเอียดของความถี่ของผลลัพธ์จาก FFT เป็นเพียงฟังก์ชันของแบนวิดธ์ของฟังก์ชันถ่วงน้ำหนักแต่ละตัวเท่านั้น(ดูรูปที่ 3.17(b)) ดังนั้นการใช้คำนิยามของความละเอียด 1/NT ต้องทำอย่างระมัดระวังและรำลึกอยู่เสมอว่ามันเป็นเพียงช่องว่างระหว่างความถี่ที่เกิดจากผลที่ได้จาก FFT เท่านั้นและไม่เกี่ยวกับ 1/NT ของวินโดว์ที่ใช้

ต่อไปเราจะแสดงผลจากการใช้ฟังก์ชันถ่วงน้ำหนักที่มีลอนข้างต่ำเพื่อให้เห็นลักษณะการนำไปใช้ลดการรั่วไหลของผลการตัดทอนในโดเมนเวลาในรูปที่ 3.18(a) เราได้แสดงรูปคลื่นโคซายน์ที่แสดงไว้ในรูปที่ 3.16(a) ที่คูณด้วยฟังก์ชันถ่วงน้ำหนักแฮนนิ่งที่แสดงไว้ในรูปที่ 3.17(a) แล้ว

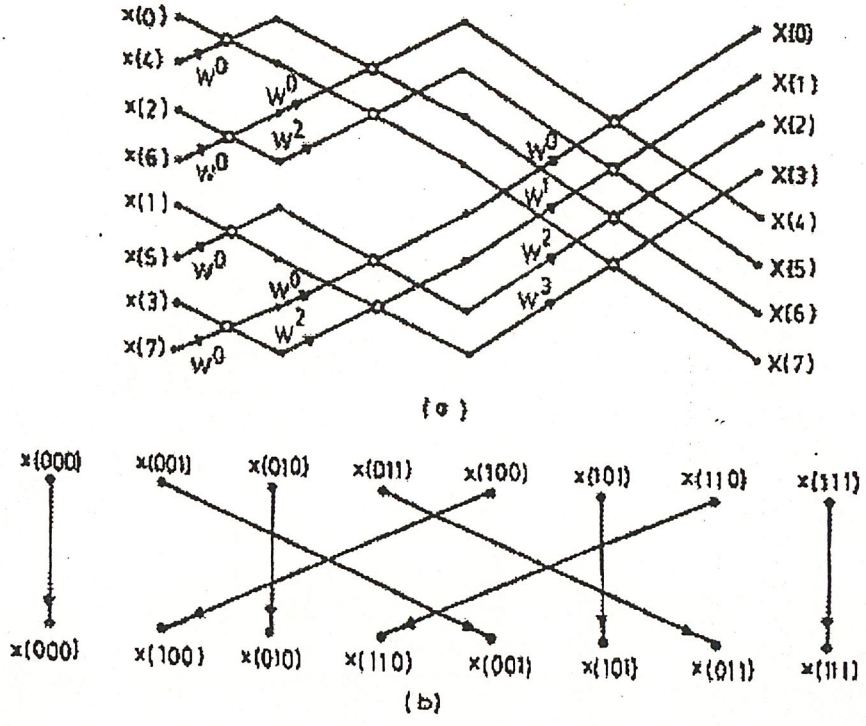
รูปที่ 3.18(b) แสดงจุดสุ่มจาก FFT ของรูปที่ 3.18(a) จากรูปจะเห็นว่าส่วนรั่วไหลลดลงไปอย่างมาก แต่องค์ประกอบหลักของความถี่มีระยะกว้างขึ้นหรือคลุมเครือเมื่อเปรียบเทียบกับจุดของอิมพัลส์ฟังก์ชันเพราะผลจากการคอนโวลูชั่นความถี่ของอิมพัลส์ฟังก์ชันกับผลการแปลงฟูเรียร์ของฟังก์ชันถ่วงน้ำหนัก



รูป 3.4 หน่วยผีเสื้อของการคำนวณแบบ DIT

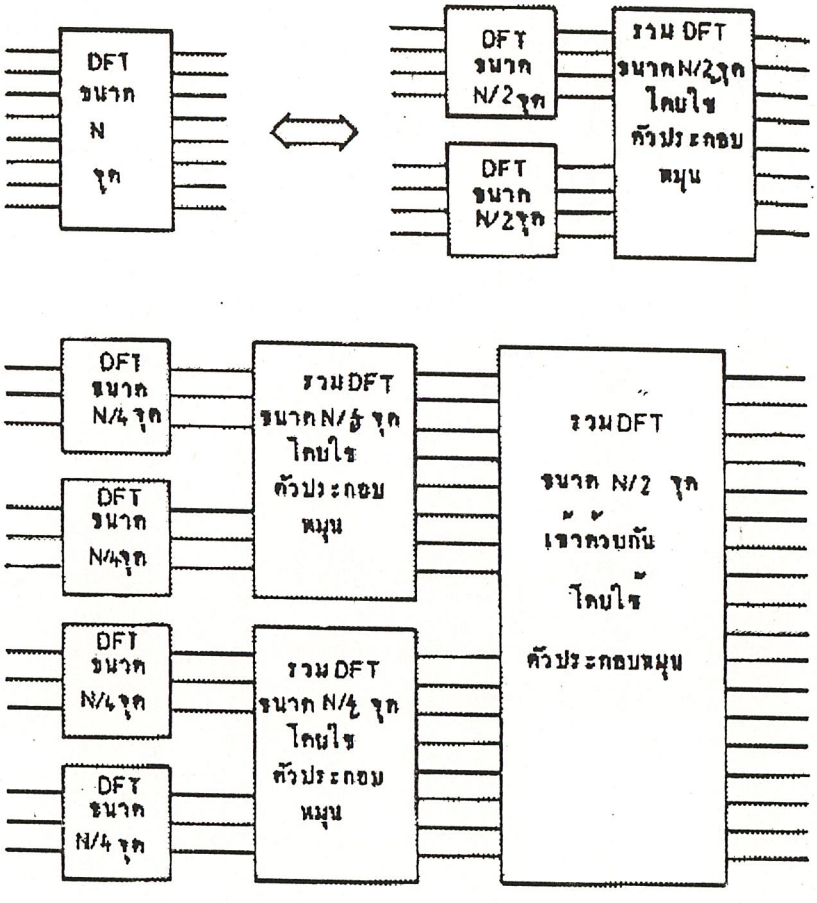


รูป 3.5(a) และ (b) แสดงวิธีแบบ DIT สำหรับ DFT แบบ 8 จุด

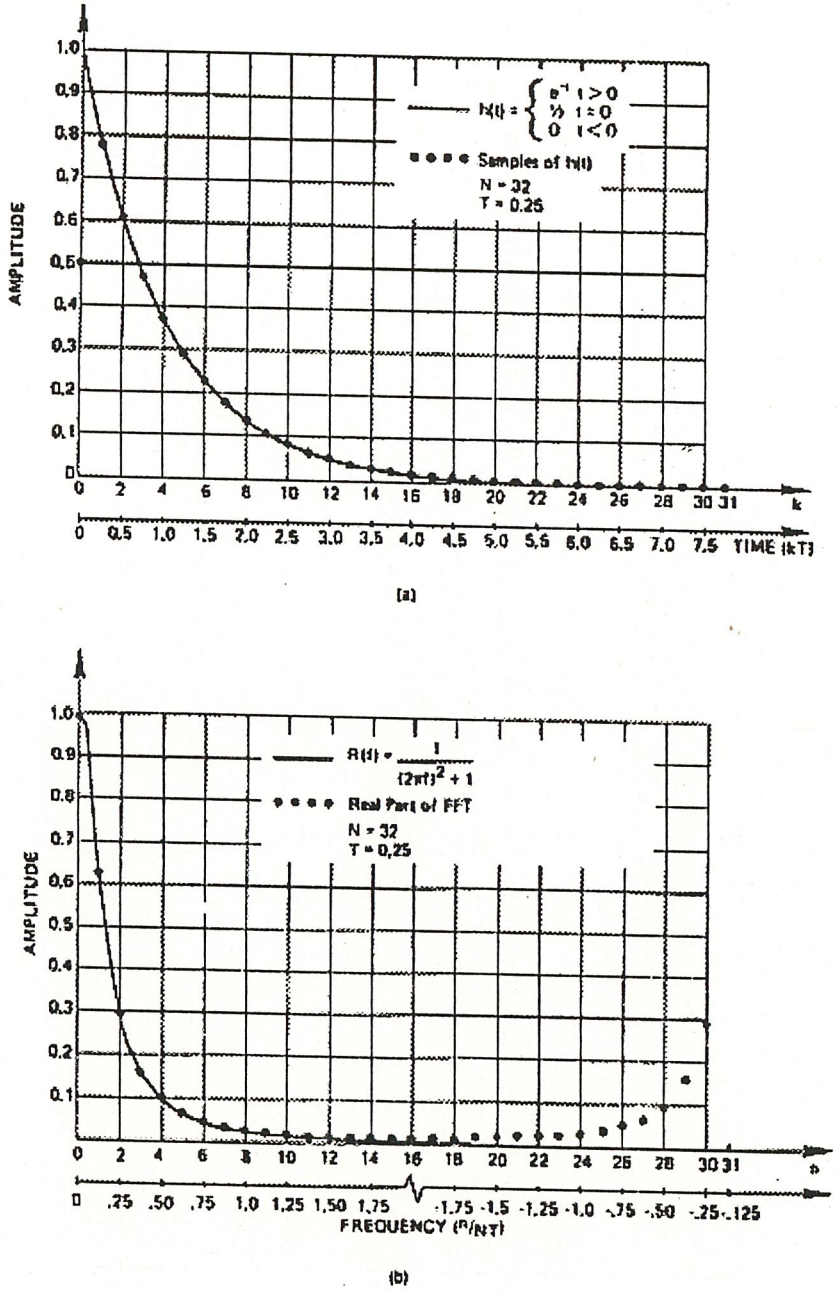


รูป 3.6 (a) กราฟการไหลสัญญาณแสดงการคำนวณตามรูป 3.5

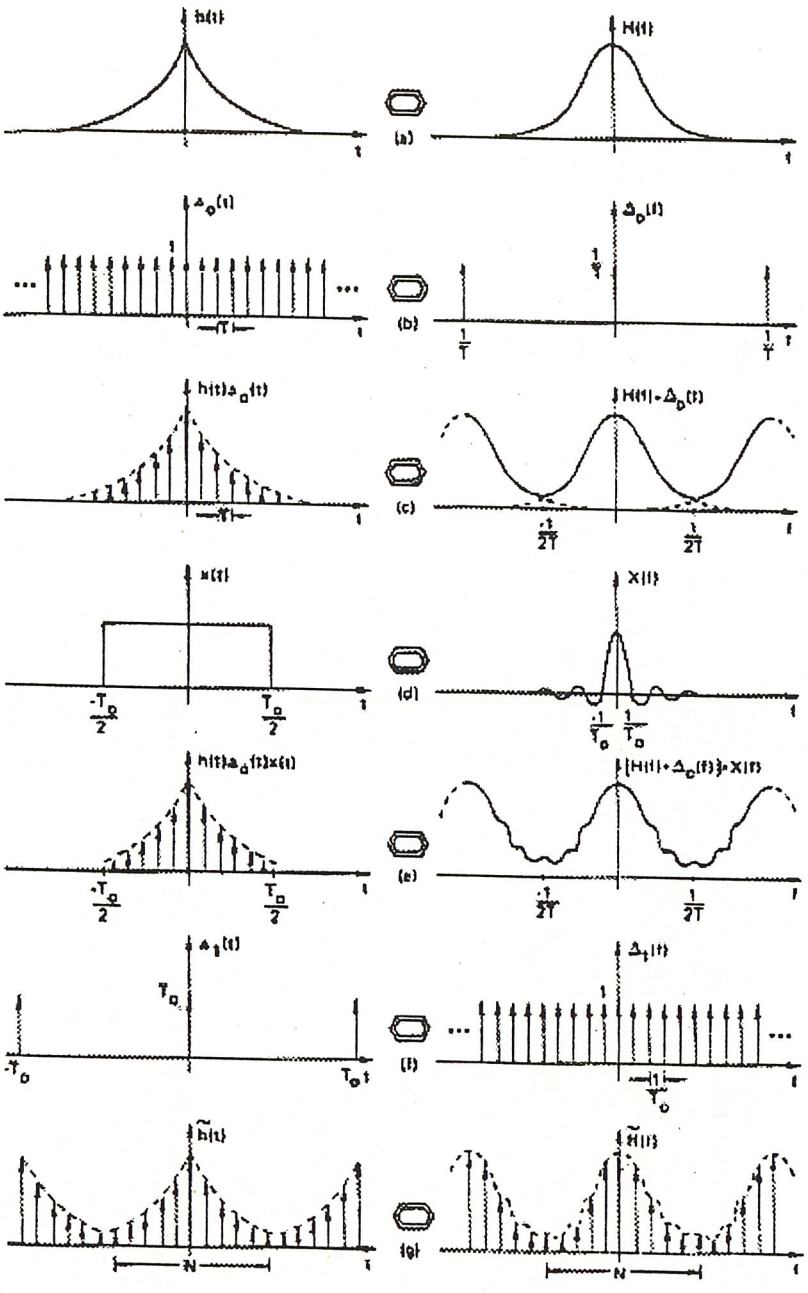
(b) แสดงการสลับตำแหน่งของลำดับ x(n) ด้วยการผันบิท



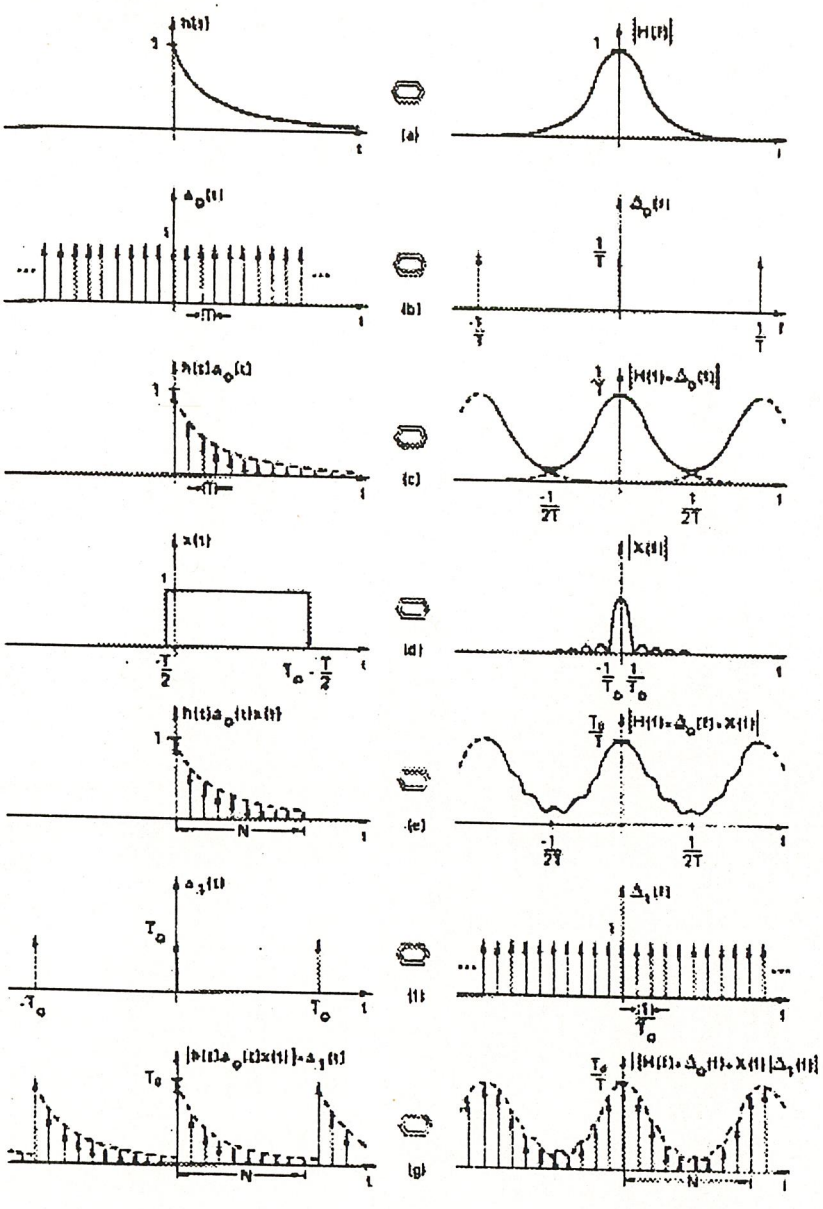
รูป 3.7 ภาพรวมแสดงขั้นตอนวิธีการคำนวณ DFT ขนาด N จุด แบบ DIT



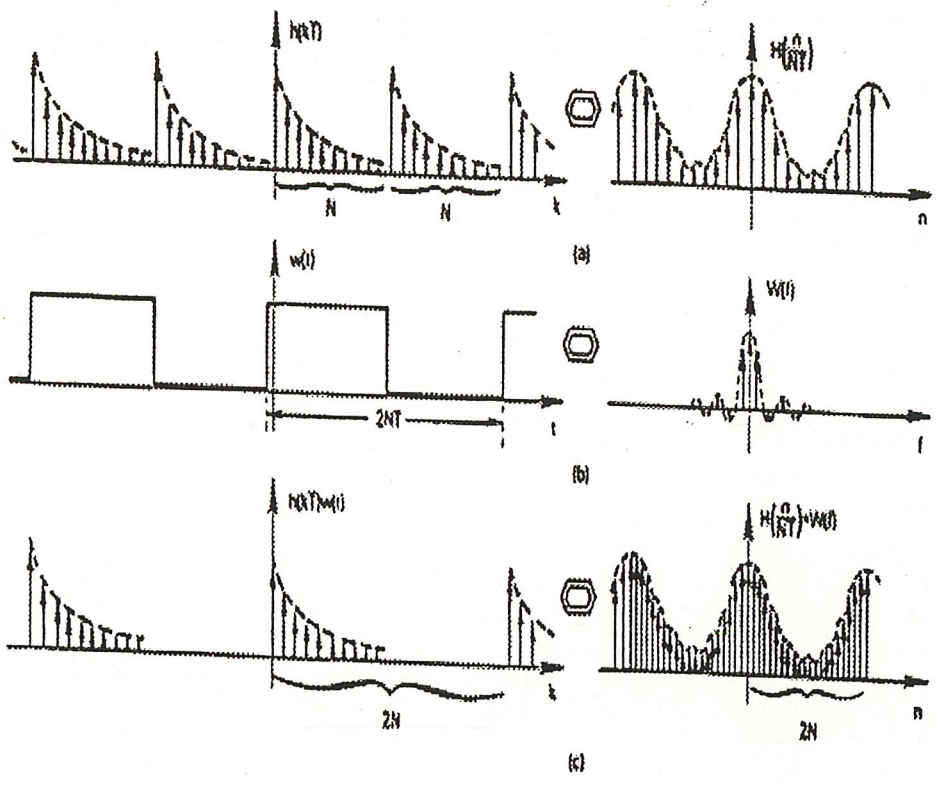
รูป 3.8 แสดงตัวอย่างการแปลงฟูเรียร์ โดยผ่าน FFT



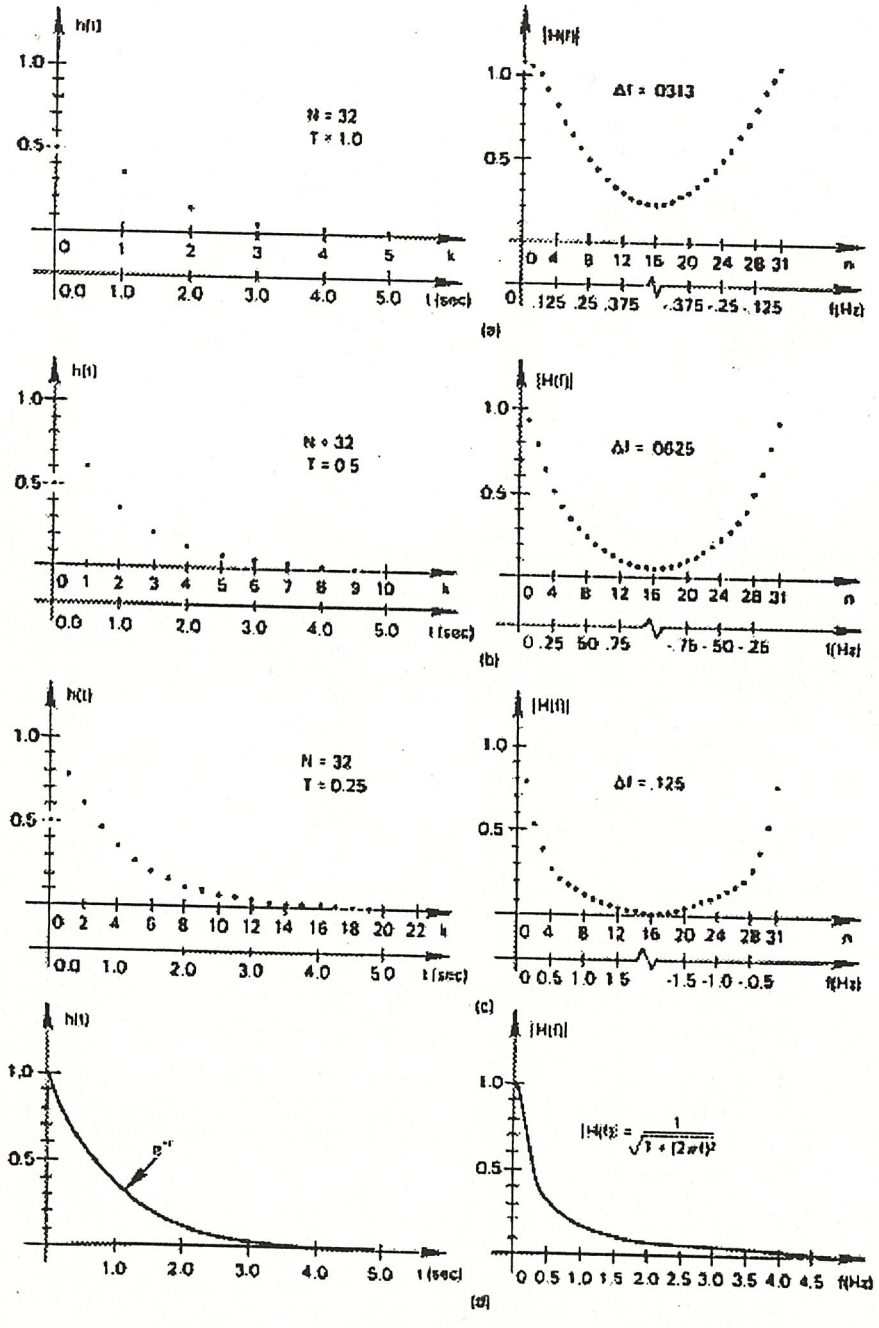
รูป 3.9 แสดงรูปของการแปลงฟูเรียร์ (DFT)



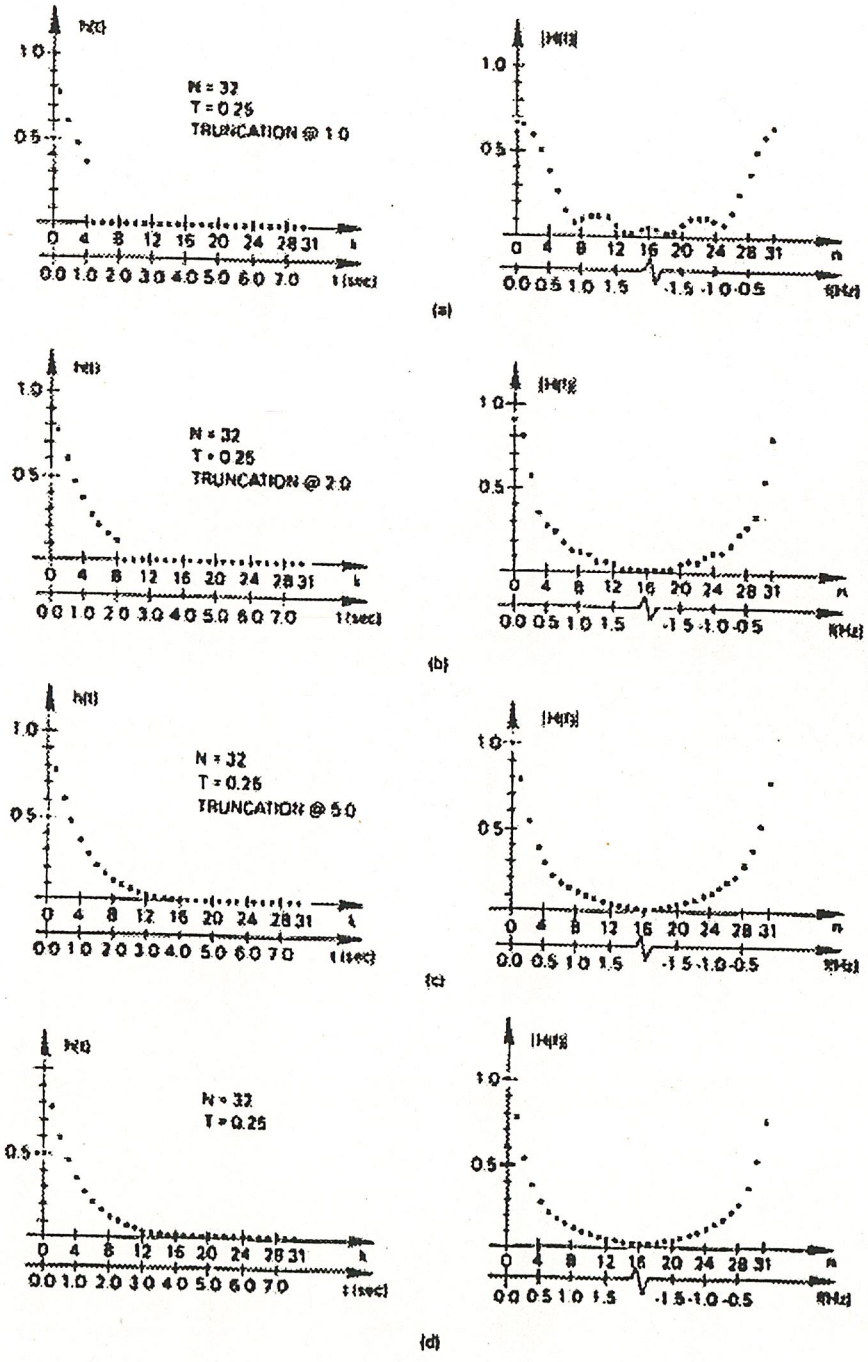
รูป 3.10 แสดงตัวอย่างการแปลงฟูเรียร์ (DFT)ในโดเมนเวลาและโดเมนความถี่



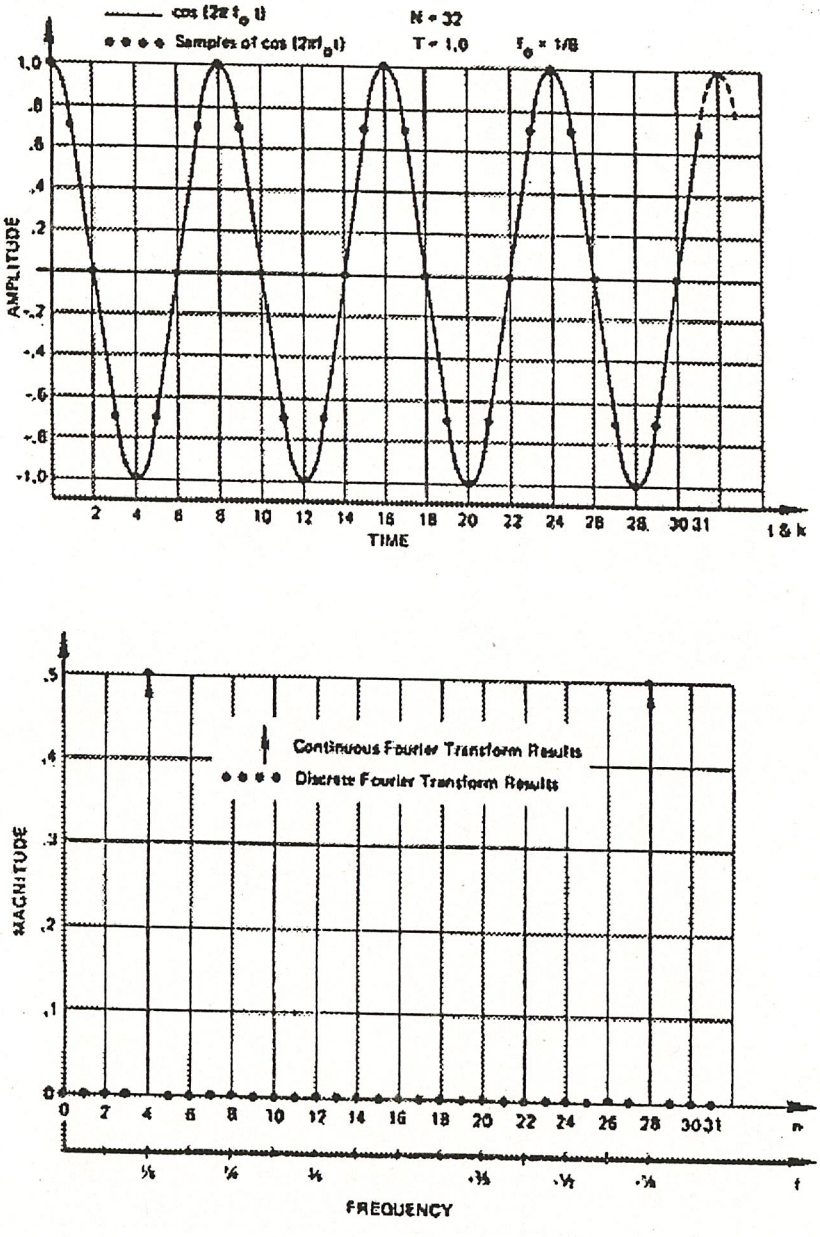
รูป 3.11 แสดงตัวอย่างของการเพิ่มความละเอียดของ FFT โดยต่อเติมศูนย์



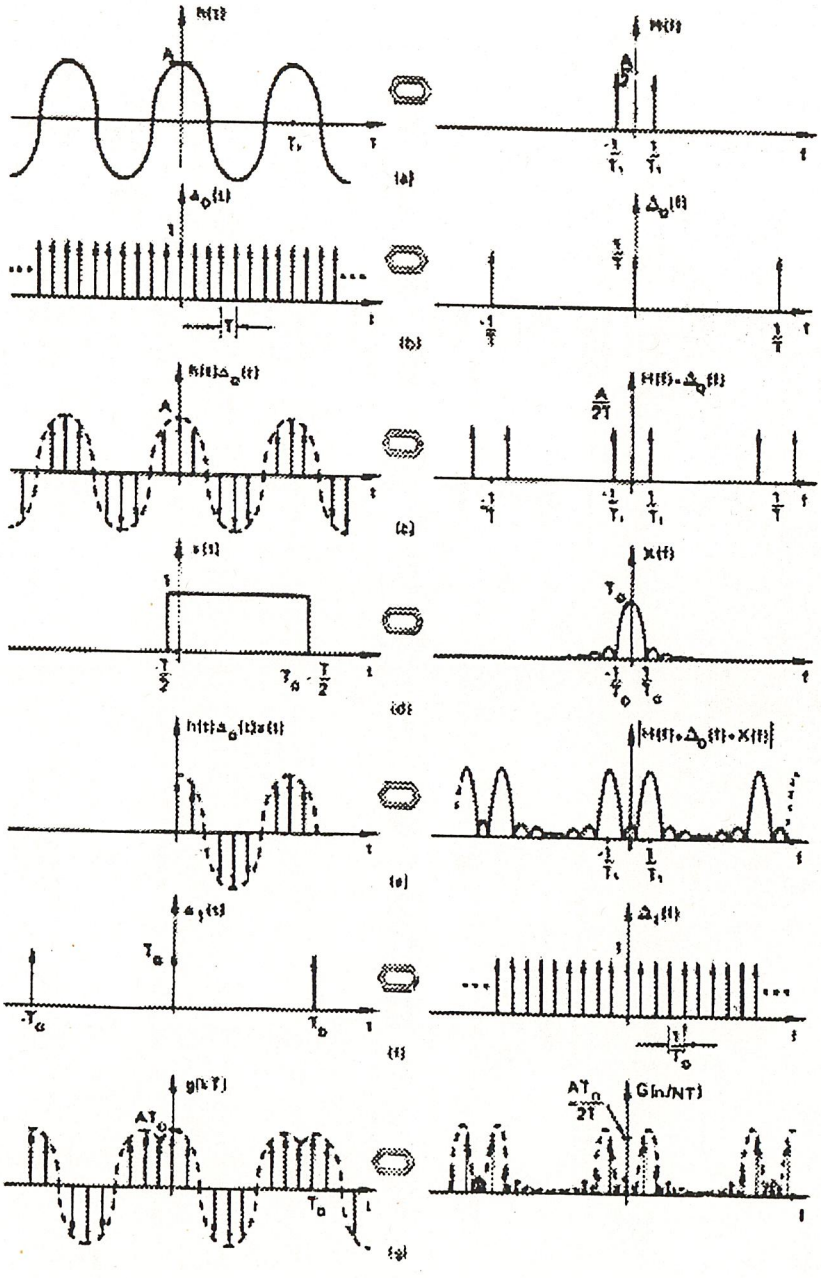
รูป 3.12 แสดงตัวอย่างของความผิดเพี้ยนในโดเมนความถี่จากอัตราการสุ่ม



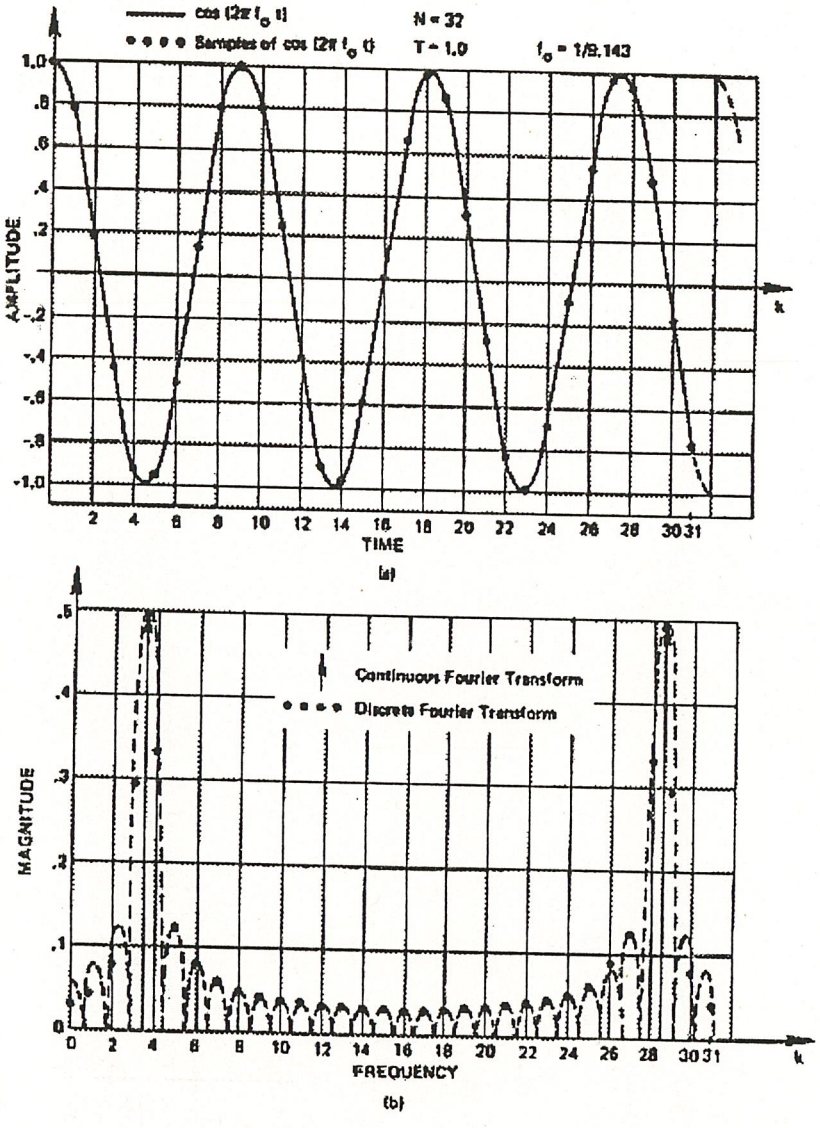
รูป 3.13 แสดงการตัดทอนในโดเมนเวลา



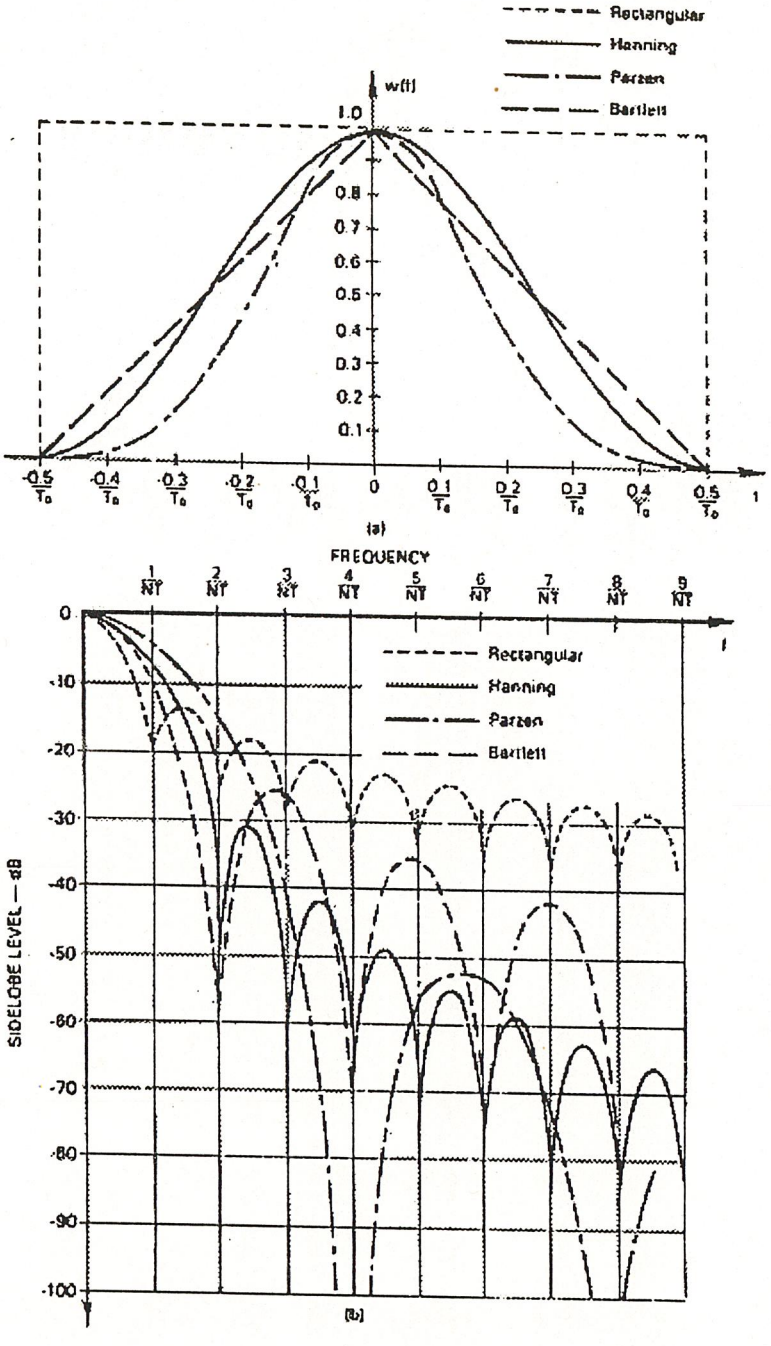
รูป 3.14 แสดงผลจาก FFT ของฟังก์ชันคาบเวลาโดยมี
ช่วงการตัดทอนเท่ากับจำนวนเท่าของคาบเวลา



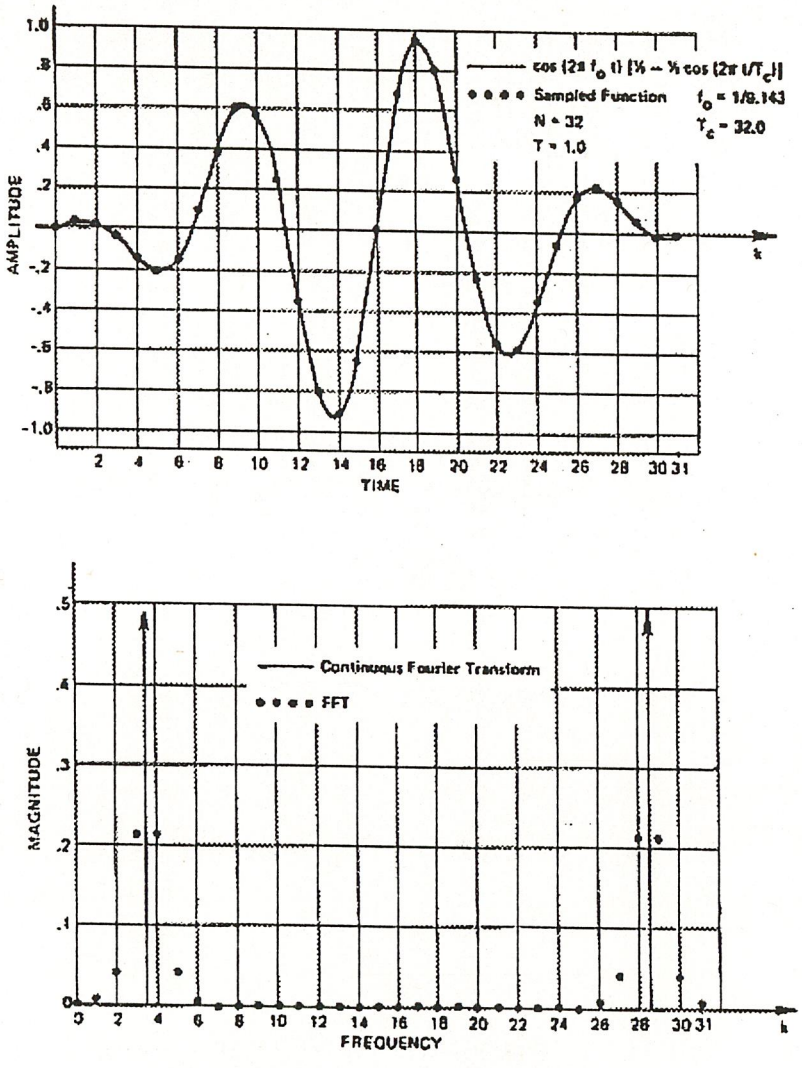
รูป 3.15 แสดงตัวอย่างการแปลงฟูเรียร์ โดยผ่าน FFT



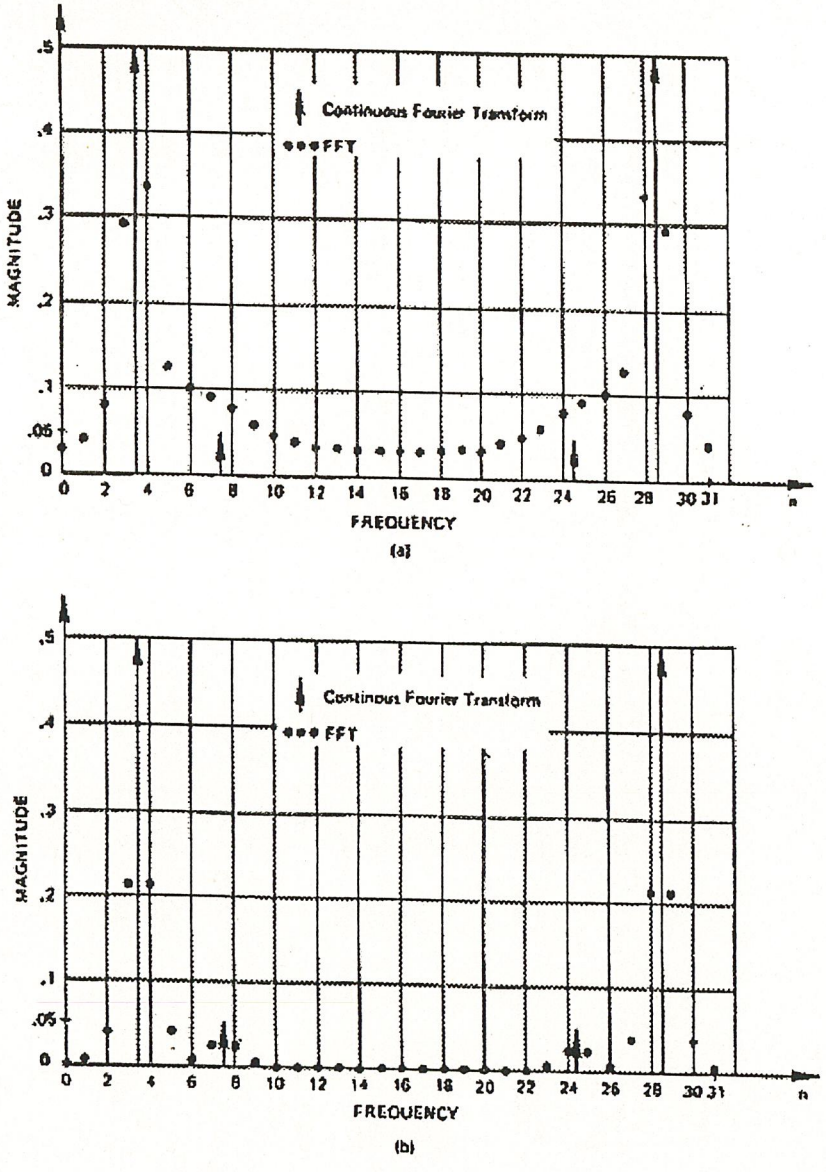
รูป 3.16 แสดงผลจาก FFT ของฟังก์ชันคาบเวลาโดยมีช่วงการตัดทอนไม่เท่ากับจำนวนเท่าของคาบเวลา



รูป 3.17 แสดงฟังก์ชันถ่วงน้ำหนักหรืออวินโ ดว์ของ FFT



รูป 3.18 แสดงตัวอย่างการใช้แฮนนิ่งฟังก์ชันเพื่อลด
ส่วนรั่วไหลของการคำนวณด้วย FFT



รูป 3.19 (a) แสดงตัวอย่างที่เลือนลางเนื่องจากผลจากการรั่วไหลของลอนข้าง
(b) สัญญาณที่ตรวจจับได้หลังจากผ่านแฮนนิ่งฟังก์ชันแล้ว

Weighting Function Nomenclature	Time Domain	Frequency Domain	Highest Side-Lobe Level (db)	3-dB Bandwidth	Asymptotic Rolloff (dB/Octave)
Rectangular	$w_R(t) = 1 \quad \lvert t\rvert \le \frac{T_0}{2}$ $= 0 \quad \lvert t\rvert > \frac{T_0}{2}$	$W_R(f) = \frac{T_0 \sin(\pi f T_0)}{\pi f T_0}$	13	$\frac{0.89}{T_0}$	6
Bartlett (triangle)	$w_B(t) = \left[1 - \frac{2\lvert t\rvert}{T_0}\right] \quad \lvert t\rvert < \frac{T_0}{2}$ $= 0 \quad \lvert t\rvert > \frac{T_0}{2}$	$W_B(f) = \frac{T_0}{2}\left[\frac{\sin\left(\frac{\pi}{2} f T_0\right)}{\frac{\pi}{2} f T_0}\right]^2$	26	$\frac{1.28}{T_0}$	12
Hanning (cosine)	$w_H(t) = \cos^2\left(\frac{\pi t}{T_0}\right)$ $= \frac{1}{2}\left[1 + \cos\left(\frac{2\pi t}{T_0}\right)\right] \quad \lvert t\rvert \le \frac{T_0}{2}$ $= 0 \quad \lvert t\rvert > \frac{T_0}{2}$	$W_H(f) = \frac{T_0}{2} \frac{\sin(\pi f T_0)}{\pi f T_0 (1 - (f T_0)^2)}$	32	$\frac{1.44}{T_0}$	18
Parzen	$w_P(t) = 1 - 24\left(\frac{t}{T_0}\right)^2 + 48\left\lvert\frac{t}{T_0}\right\rvert^3 \quad \lvert t\rvert < \frac{T_0}{4}$ $= 2\left[1 - \frac{2\lvert t\rvert}{T_0}\right]^3 \quad \frac{T_0}{4} < \lvert t\rvert < \frac{T_0}{2}$ $= 0 \quad \lvert t\rvert \ge \frac{T_0}{2}$	$W_P(f) = \frac{3T_0}{8}\left[\frac{\sin(\pi f T_0/4)}{\pi f T_0/4}\right]^4$	52	$\frac{1.82}{T_0}$	24

ตารางที่ 3.1 ฟังก์ชันถ่วงน้ำหนักข้อมูล

บทที่ 4
การวิเคราะห์เสียงพูด(Speech Analysis)

ในการเก็บหรือจดจำเสียงพูดเราต้องพยายามลดช่วงของสัญญาณเสียงพูดที่เกิดซ้ำกัน โดยใช้ตัวแทนของเสียงพูดที่สามารถแทนลักษณะสำคัญของเสียงพูดนั้นในเทอมของค่าพารามิเตอร์(parameter) เพื่อให้จัดการกับข้อมูลได้ง่าย

ต่อไปเราจะได้อธิบายถึงวิธีการที่ใช้ในการวิเคราะห์เสียงพูดทั้งในโดเมนเวลา (กระทำกับเสียงพูดที่เก็บไว้โดยตรง) และโดเมนความถี่(ผ่านการแปลงเป็นความถี่ก่อน) เพื่อหาตัวแทนของสัญญาณเสียงพูดในรูปของพารามิเตอร์ที่เหมาะสมที่สุดสามารถนำไปใช้งานได้และมีข่าวสารของข้อมูลครบถ้วนการวิเคราะห์ในโดเมนเวลาเราต้องการการคำนวณเพียงเล็กน้อยทำให้ถูกจำกัดให้วัดได้เฉพาะลักษณะง่ายๆเท่านั้นเช่น การวัดพลังงานและความเป็นคาบเวลาของสัญญาณ ในขณะที่การวิเคราะห์ในโดเมนความถี่จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพมากกว่าแต่การสุ่มสัญญาณเสียงจะต้องมีความเที่ยงตรงสูง

เทคนิคที่ใช้ในการวิเคราะห์เสียงพูดอาจทำได้ทั้งแบบดิจิตอลและ อนาลอกการประมวลผลสัญญาณอนาลอกเป็นการใช้วงจรทางด้านอิเลคทรอนิคส์ซึ่งมีข้อดีอยู่ตรงที่มีความเร็วสูง แต่จำเป็นต้องใช้วงจรเฉพาะอย่างทำให้ต้องต่อสายใหม่และปรับค่าใหม่ทุกครั้งที่นำไปประยุกต์ใช้กับงานอื่น ในขณะที่เทคนิคทางด้านดิจิตอลสร้างและเปลี่ยนแปลงได้ง่ายกว่าเพียงแต่ต้องการซอฟแวร์หรือโปรแกรมและฮาร์ดพิเศษ(ไมโครโปรเซสเซอร์และชิพ) ถึงแม้ว่าอาจจะมีปัญหาในเรื่องความเร็วไม่สามารถตอบสนองในเวลาจริงได้แต่ในปัจจุบันเทคโนโลยีทางด้าน VLSI ได้พัฒนาไปอย่างมากทำให้ลดข้อเสียเปรียบของเทคนิคดิจิตอลไปได้มาก

4.1.การวิเคราะห์เสียงพูดในช่วงสั้น ๆ(Short-time speech analysis)

สัญญาณเสียงพูดเป็นสัญญาณที่เปลี่ยนไปตามเวลาโดยเกิดในลักษณะแบบสุ่ม (random) แต่ก็ต้องขึ้นกับการควบคุมเสียงของผู้พูดด้วยเพราะเสียงที่เปล่งออกมาในช่วงระยะเวลาหนึ่งเท่านั้น จะขึ้นอยู่กับรูปร่างของท่อกำธรเสียง(vocal tract) และลักษณะการสั่นของเส้นเสียง(vocal cord) สัญญาณของเสียงพูดจึงเป็นสัญญาณที่เป็นคาบเวลาชั่วขณะ(quasi-periodic) หมายความว่าสัญญาณเสียงพูดมีคาบเวลาคงที่ในระยะเวลาสั้นๆและมีการเปลี่ยนแปลงในระหว่างระยะเวลาสั้นๆนั้น

ถ้าเราพูดช้ามากๆเสียงพูดที่ได้อาจจะเปลี่ยนแปลงอยู่ในช่วงมากกว่า 200 ms ก็ได้แต่การพูดโดยทั่วไปลักษณะการเปลี่ยนแปลงของเสียงพูดจะอยู่ช่วงค่าเฉลี่ยประมาณ 80 ms ในการวิเคราะห์เราจะใช้วินโดว์(window) วิเคราะห์สัญญาณเป็นช่วงๆหรือเรียกว่าการวิเคราะห์เฟรม(frame) ช่วงเวลาของวินโดว์มีค่าไม่แน่นอน สัญญาณเสียงพูดที่เร็วมากและไม่มีช่วงหยุดเลยเราอาจจะต้องใช้ช่วงเวลาของวินโดว์ต่ำถึง 5-10 ms

4.1.1. การใช้วินโดว์(windowing)

รูปแบบของการเฉลี่ยโดยปกติมีหลายวิธีหลายอย่างเพื่อให้ได้เส้นของพารามิเตอร์ที่เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้องทางเลือกทางหนึ่งที่เลือกใช้คือขนาดของวินโดว์ซึ่งมีปัจจัยที่เกี่ยวข้องคือ

1.วินโดว์ต้องมีขนาดเล็กเพียงพอกับลักษณะของคำพูด

2.วินโดว์จะต้องยาวพอที่จะใช้ในการคำนวณหาพารามิเตอร์

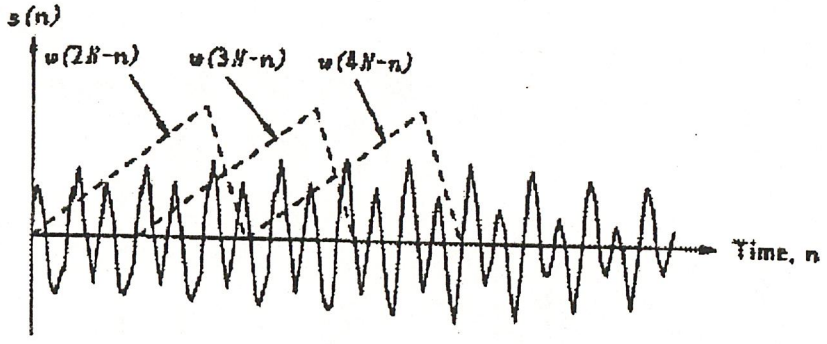
3.วินโดว์ที่ดีจะต้องไม่สั้นไปจนข้ามบางช่องของคำพูด การวิเคราะห์จะทำเป็นคาบๆไป ซ้ำๆกันตลอดที่ได้สัญญาณ

เงื่อนไขท้ายนี้จะขึ้นอยู่กับ frame rate (จำนวนครั้งต่อวินาที ที่กระทำต่อสัญญาณที่วิเคราะห์) มากกว่าขนาดของวินโดว์ โดยปกติ frame rate ประมาณ 2 เท่า ของความถี่เพื่อให้วินโดว์ทับกัน 50 % การทำวินโดว์เป็นการคูณสัญญาณเสียงโดยวินโดว์ที่มีช่วงเวลาจำกัด (finite-duration window) w(n)ซึ่งกลุ่มของสัญญาณเสียงที่สุ่มมาจะถูกให้น้ำหนักโดยรูปของวินโดว์

วินโดว์อาจมีคาบเวลาแบบไม่จำกัด แต่ในทางปฏิบัติจะใช้แบบมีจุดสิ้นสุดเพื่อให้ง่ายในการคำนวณโดยการใช้ฟังก์ชัน w(n) ตรวจสอบส่วนต่างๆของ S(n) โดยการเคลื่อนวินโดว์ตามรูปที่ 4.1 วินโดว์ที่มีรูปแบบง่ายที่สุดคือวินโดว์รูปสี่เหลี่ยม(rectangular window) r(n)

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

$$0 \quad \text{ในกรณีอื่นๆ}$$



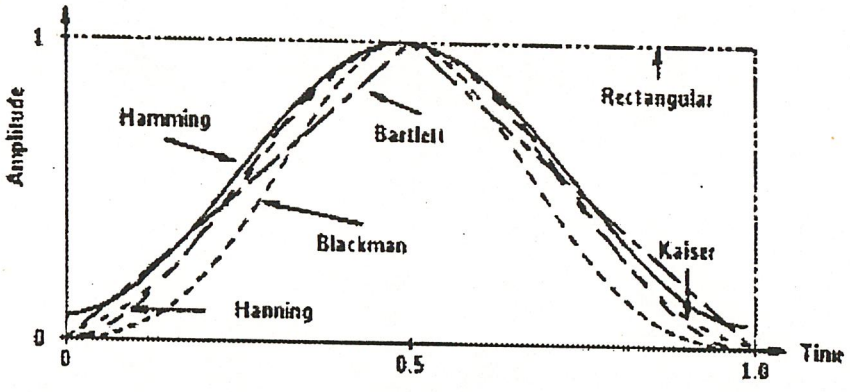
รูปที่ 4.1 แสดงสัญญาณเสียง s(n) กับวางทับสัญญาณของวินโดว์ 3 ตัว
โดยมีจุดเริ่มต้นของเวลาที่จุดสุ่ม 2N,3N และ 4N

การวิเคราะห์แบบนี้เป็นการจำกัดช่วงจุดสุ่มให้เหลือ N โดยแต่ละจุดสุ่มมีการถ่วงน้ำหนักเท่าๆกัน

วินโดว์ที่ใช้ในงานจริงนั้นต้องมีรูปร่างตามรูปที่ 6.2 สมมติว่าเสียงพูดประมาณค่าคงที่ในช่วง 10 ms เราต้องใช้วินโดว์ที่มีช่วงเวลา เท่ากับ 20 ms โดยให้จุดตรงกลาง 10 ms ถ่วงน้ำหนักมากกว่าจุดต้นและจุดปลายเหตุผลที่ต้องถ่วงน้ำหนักจุดกลางมากกว่าจุดปลายเพราะรูปร่างของวินโดว์มีผลต่อเอาท์พุทต่อพารามิเตอร์เสียงพูดเมื่อเราเลื่อนวินโดว์ในโดเมนเวลาเพื่อวิเคราะห์เฟรมของสัญญาณเสียงพูดผลของพารา
มิเตอร์อาจเปลี่ยนแปลงเป็นอย่างมากถ้าใช้ฟังก์ชัน r(n) ต่างกัน ตัวอย่างเช่นการพลังงานโดยการบวกผลกำลังสองของจุดสุ่มในวินโดว์สี่เหลี่ยมทำให้เกิดการกระเพื่อมรอบจุดความถี่ดังนั้นเราจะใช้แฮนนิ่งวินโดว์(hanning window) แทนวินโดว์สี่เหลี่ยมซึ่งมีรูปร่างเท่ากัน

$w(n) = h(n) = 0.54-0.46\cos(2\pi n/N-1)$ เมื่อ $0 <= n <= N-1$

$= 0$ ในกรณีอื่นๆ



รูปที่ 4.2 รูปแบบของวินโดว์ซึ่งมีช่วงเวลาเท่ากับหนึ่ง

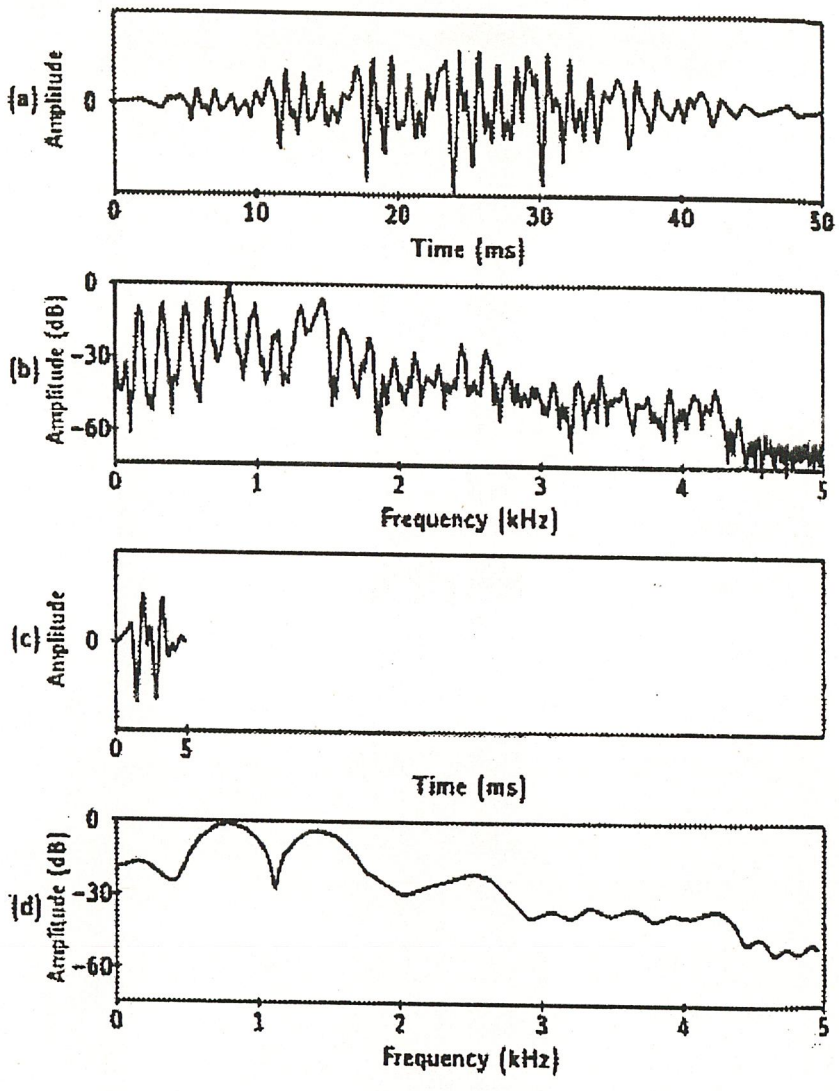
การทำให้ขอบของวินโดว์ลาดลง ในการวิเคราะห์เฟรมโดยการเลื่อนตลอดแนวความยาวของสัญญาณทั้งหมดไม่มีผลเสียต่อพารามิเตอร์ของเสียงพูด

4.2.พารามิเตอร์ในโดเมนเวลา(Time- Domain Parameter)

ในขบวนการของสัญญาณเสียงในฟังก์ชันเวลาจะมีความได้เปรียบในด้านความง่าย คำนวณเร็ว และง่ายในการแปลงความหมาย พารามิเตอร์ของเสียงสำหรับเข้ารหัสและการจำสามารถจะทำได้จากการวิเคราะห์ฟังก์ชันเวลาเช่น พลังงาน(แอมพลิจูด),เสียง

4.2.1.การวิเคราะห์ในโดเมนเวลา

การวิเคราะห์ในฟังก์ชันวลาที่เปลี่ยนจากสัญญาณเสียงพูดสูพารามิเตอร์ 1 ตัวหรือมากกว่าซึ่งโดยปกติจะช้ากว่าสัญญาณเริ่มแรกมาก การเก็บและการ manipulate ในกรณีของสัญญาณเสียงจะมีประสิทธิภาพมากกว่าการเก็บสัญญาณเสียงโดยตรงตัวอย่างเช่น คำพูดที่ถูกสุ่มมาที่ 6000 - 10000 ตัวอย่าง/วินาที เพื่อที่จะเก็บช่วงความถี่ให้ถึง 3-5 kHz และโดยปกติแล้วการเปล่งเสียง 100 ms จะต้องการถึง 1000 ตัวอย่าง เพื่อให้ได้ความถี่ถูกต้องในการแสดงผลออกมา การสุ่มสัญญาณพารามิเตอร์โดยทั่วไปจะสุ่มที่ 40-100 ตัวอย่าง/s (มีระบบที่ต้องสุ่มถึง 200 ตัวอย่าง/s) ดังนั้นเมื่อเปลี่ยนคลื่นเสียงไปสู่กลุ่มพารามิเตอร์ อัตราการสุ่มสามารถลดลง



รูปที่ 4.3 (a) แสดงสัญลักษณ์ที่ถูกคูณด้วยแฮมมิ่งวินโดว์

(b) สเปคตรัมของรูป (a)

(c) แสดงสัญญาณที่ถูกคูณด้วยแฮมมิ่งวินโดว์ 5 ms

(d) สเปคตรัมของรูป (c)

เทคนิคการประมวลผลแบบ short-time ส่วนมากจะสร้างพารามิเตอร์อยู่ในรูปแบบ

$$Q(n) = \sum_{m=-\infty}^{\infty} T[S(m)]w(n-m)$$

สัญญาณเสียง S(n) ได้รับการเปลี่ยนเป็นฟังก์ชัน T ซึ่งถูกให้น้ำหนักโดยวินโดว์ w(n) และถูกรวมกันเป็นพารามิเตอร์ Q(n)ที่อัตราสุ่มเริ่มต้นซึ่งจะเป็นตัวแทนแสดงสมบัติของคำพูดที่ถูกเฉลี่ย

ยภายใต้ช่วงของวินโดว์ Q(n) ได้มาจากการคอนโวลูชันของ T[S(n)] กับ w(n) เพื่อที่จะกระจายฟังก์ชันนั้น w(n) ใช้เป็น low pass filter,Q(n) จะถูกทำให้ลายเรียบลงไปในลักษณะของฟังก์ชัน T[S(n)] เดิม เมื่อ Q(n) เป็นเอาท์พุทของวงจรกรองผ่านความถี่ต่ำ (low pass filer) ในกรณีส่วนใหญ่แบนด์วิดของ Q(n) จึงเหมือนกับของวงจรกรองผ่านความถี่ต่ำ w(n) เพื่อการทำงานและการจัดเก็บที่มีประสิทธิภาพมากขึ้น Q(n)

4.3.ค่าพารามิเตอร์ในโดเมนความถี่(Frequency-Domain Parameter)

ค่าพารามิเตอร์สำคัญจำนวนมากถูกพบในโดเมนความถี่ เพราะท่อกำธรเสียงให้กำเนิดสัญญาณวิเคราะห์เป็นความถี่ได้ง่ายกว่าทำในโดเมนเวลาโดยตรงเสียงพูดของคนคนเดียวในแต่ละครั้ง เมื่อพิจารณาในโดเมนของเวลาจะเห็นผลของความแตกต่างของสัญญาณ แต่เมื่อพิจารณาในโดเมนความถี่สัญญาณที่ได้มีความคล้ายคลึงกันมาก

ระบบการได้ยินของมนุษย์จะตอบสนองต่อรูปร่างของสัญญาณเสียง(หรือขนาดที่กระจายอยู่ในโดเมนความถี่) มากกว่าที่เฟสของสัญญาณในโดเมนเวลาด้วยเหตุผลเหล่านี้ การวิเคราะห์ความถี่จึงกถูกใช้ในการดึงค่าพารามิเตอร์สำคัญจากสัญญาณเสียงพูด ต่อไปจะได้อธิบายถึงวิธีการต่างๆที่จะนำไปใช้ในการวิเคราะห์ความถี่

4.3.1. การวิเคราะห์ความถี่ฟิลเตอร์แบงค์(filter bank)

การวิเคราะห์ความถี่ด้วยฟิลเตอร์แบงค์เป็นเทคนิคที่นิยมกันมากเพราะสามารถตอบสนองได้ในเวลาจริง โครงสร้างง่ายและไม่แพงโดยใช้ฟิลเตอร์แบงค์หรือกลุ่มของแบนพาลฟิลเตอร์(bandpass filter) ซึ่งอาจจะเป็นอนาลอกฟิลเตอร์หรือดิจิตอลฟิลเตอร์ก็ได้ แบนพาสฟิลเตอร์แต่ละตัวจะแยกวิเคราะห์ความถี่ของสัญญาณเสียงเป็นช่วงๆ ในทางการค้า วิธีฟิลเตอร์มีความยืดหยุ่นมากกว่าการวิเคราะห์ด้วย DFT เพราะช่วงแบนวิดธ์(bandwidth) สามารถปรับช่วงตามขอบเขตการรับฟังเสียงของมนุษย์ได้มากกว่าการกำหนดตายตัวว่าต้องมีแบนวิดธ์กว้างแคบเท่านี้เท่านั้น

ในการประยุกต์ใช้งานจำนวนมากต้องการกลุ่มของพารามิเตอร์ความถี่ไม่มากในการอธิบายการกระจายของพลังงานของความถี่ โดยทั่วไปการใช้กลุ่มของแบนพาสฟิลเตอร์ 8-12 ตัวก็สามารถให้ค่าตัวแทนของความถี่ที่ซับซ้อนและมีประสิทธิภาพมากกว่าการวิเคราะห์ด้วย DFT ซึ่งให้รายละเอียดปลีกย่อยของแต่ละความถี่ได้มากกว่า

ในวิธีกาบบต่อเนื่องโดยใช้ฟังก์ชันวินโดว์ w ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้างความถี่ของฟิลเตอร์แต่ละตัวเท่าๆกันและมีแบนวิดธ์ที่เพิ่มขึ้นแบบลอกการิทึม(logarithm) แต่ไม่ควรเกิน 1 kHz โดยทั่วไปนิยมใช้ฟิลเตอร์ 1/3 ออคเตป(one-third-octave filter) ในระบบการจดจำเสียงพูดจริงๆนั้นอาจใช้ฟิลเตอร์แบงค์ถึง 2 ครั้ง ครั้งแรกเป็นการจำแนกเสียงอย่างคร่าวๆก่อน โดยใช้ฟิลเตอร์เพียงไม่กี่ตัวและวิเคราะห์หารายละเอียดปลีกย่อยอีกครั้งโดยใช้กลุ่มของฟิลเตอร์มากขึ้น

4.3.2. การวิเคราะห์ฟูเรียร์ช่วงสั้น(Short -Time Fourier Analysis)

การวิเคราะห์ฟูเรียร์ช่วงสั้นเป็นเทคนิคการหาความถี่ที่ใช้ในกันมานานแล้ว การวิเคราะห์ฟูเรียร์ให้เป็นตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชันความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งตลอดเวลาดังนั้นจึงจำเป็นต้องใช้การวิเคราะห์ในช่วงสั้นโดยใช้วินโดว์ การแปลงฟูเรียร์ช่วงสั้นมีสมการดังนี้

$$S_n(e^{jw}) = \sum_{m=-\infty}^{\infty} s(m)\,(e^{-jwm})w(n-m)$$

ในการคำนวณเราต้องใช้ DFT แทนการแปลงฟูเรียร์แบบต่อเนื่องโดยใช้ฟังก์ชันวินโดว์ w ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว(N คือช่วงเวลาหรือขนาดของวินโดว์ที่ใช้ในการแปลง DFT) ข่าวสารต่างๆใน $S_n(e^{jw})$ จะไม่สูญหายไปจากข้อมูลเดิม $S_n(e^{jw})$ ถ้าการแปลงนั้นสุ่มมาด้วยความถี่สูงเพียงพอ(คือ ช่วงระยะห่างระหว่าง N) และวินโดว์ w(n) ไม่มีจุดสุ่มที่เป็นศูนย์ตลอดช่วง N ตัวแปร N เป็นที่ต้องระวังมากเป็นพิเศษในการวิเคราะห์ความถี่ช่วงสั้น ถ้าค่าของ N ต่ำ(ใช้วินโดว์ช่วงสั้น) จะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำเฉพาะในช่วงสั้นๆเท่านั้น ในทางตรงกันข้ามถ้า N มีขนาดใหญ่จะให้ผลความละเอียดของเวลาที่แย่ แต่จะทำให้โดเมนความถี่มีความละเอียดสูงกว่า

บทที่ 5
การรู้จำ(recognition)

5.1.ประเภทของการรู้จำ

ในการประมวลผลสัญญาณเสียงพูดมีการประยุกต์ใช้งานสำคัญ 2 ด้านคือ

5.1.1.การรู้จำเสียงพูด(Speech recognition)

5.1.2.การรู้จำผู้พูด(Speaker recognition)

ทั้งสองหัวข้อนี้เป็นการรู้จำรูปแบบอย่างหนึ่ง การรู้จำรูปแบบได้ถูกนำไปประยุกต์ใช้งานหลายๆด้านเช่น การประมวลสัญญาณภาพแต่ในที่นี้จะอธิบายถึงการรู้จำรูปแบบที่เกี่ยวข้องกับการประมวลผลสัญญาณเสียงพูดเท่านั้น

5.2.การรู้จำรูปแบบ(Pattern recognition)

การรู้จำรูปแบบเป็นการเปรียบเทียบรูปแบบตรวจสอบ(test pattern) ซึ่งเป็นตัวแทนของสิ่งที่ไม่รู้เพื่อแยกแยะหรือรู้จำกับรูปแบบอ้างอิง (reference pattern) 1 ตัวหรือหลายตัวซึ่งได้รู้ลักษณะมาก่อน

ในแต่ละรูปแบบจะอยู่ในรูปของเวคเตอร์แต่ละสมาชิกของเวคเตอร์เป็นค่าที่วัดได้ของลักษณะเด่น(feature) ลักษณะเด่นเป็นลักษณะที่วัดได้ของลักษณะสัญญลักษณ์อินพุทและสามารถนำไปใช้ประโยชน์ในการรู้จำรูปแบบได้เช่น ในระบบการรู้จำเสียงพูดแบบแยกคำรูปแบบต่างๆจะเก็บอยู่ในรูปของฟังก์ชันเวลาเพราะค่าของแต่ละลักษณะที่วัดได้นั้นไม่ได้วัดที่จุดใดจุดหนึ่งของคำพูดแต่มาจากการวัดทั้งคำรูปแบบเหล่านี้เราเรียกว่า ต้นแบบ(template) ในระบบการรู้จำจะมีการทำงานคล้ายตัวอย่างดังต่อไปนี้

- สมมติว่า ระบบการรู้จำผู้พูดเป็นระบบที่ทำขึ้นสำหรับผู้พูด 50 คน เมื่อมีการพูดคำหรือวลี ระบบจะทำการประมวลผลสัญญาณเสียงพูดและดึงเอาลักษณะเด่นออกมา(โดยแยกคำและวัดค่า) เพื่อทำเป็นรูปแบบอ้างอิง ดังนั้นระบบนี้จึงมีไลบรารี่รูปแบบอ้างอิง(reference library) อยู่ 50 แบบเมื่อทำการเปรียบเทียบรูปแแบบตรวจสอบจะถูกเปรียบเทียบกับรูปแบบอ้างอิงในไลบรารี่ทั้งหมด เพื่อหารูปแบบอ้างอิงที่ใกล้เคียงกับรูปแบบตรวจสอบมากที่สุดเพื่อแยกแยะว่าเป็นผู้พูดคนใด

ระบบการรู้จำอัตโนมัติ นอกจากใช้ในการรู้จำคำพูดและผู้พูดแล้วยังอาจนำไปใช้ในงานด้านอื่นอีก โดยแยกสิ่งที่ไม่รู้ออกเป็นคลาส(class) เช่น ในระบบรู้จำตัวเลข คลาส คือคำว่า { ศูนย์,หนึ่ง,สอง,สาม,...,เก้า} ระบบจะมีไลบรารี่ของรูปแบบอ้างอิงในแต่ละคลาส ขบวนการรู้จำจะทำการเปรียบเทียบความคล้ายคลึงของสัญญาณไม่รู้กับคลาสแต่ละคลาสเพื่อหารูปแบบที่ใกล้เคียงที่สุด

การทำงานของการรู้จำโดยทั่วไปมีอยู่ 2 ขั้นตอน ดังนี้

1.การเรียนรู้ (หรืออาจจะเรียกว่า การฝึกก็ได้) เป็นการรวบรวมไลบรารี่ของรูปแบบอ้างอิง

2.การรู้จำ เป็นการใช้ไลบรารี่ที่สมบูรณ์แบบเปรียบเทียบกับอินพุทที่ไม่รู้

ในการเรียนรู้ ข้อมูลสำหรับแต่ละคลาสจะถูกป้อนเข้าระบบ ระบบจะทำการสร้างรูปแบบอ้างอิงหรือต้นแบบของแต่ละคลาส โดยการเฉลี่ยรูปแบบจำนวนมากแล้วนำมารวมกันเป็นไลบรารี่โดยแยกออกเป็นคลาสต่างๆในการรู้จำระบบจะทำการคำนวณรูปแบบของลักษณะเด่นของอินพุทที่ไม่รู้และทำการตรวจสอบกับคลาสต่างๆเพื่อหาคลาสที่มีรูปแบบอ้างอิงใกล้เคียงกับลักษณะเด่นที่ได้มากที่สุด

การออกแบบระบบการรู้จำมีปัญหาที่สำคัญอยู่ 2 ประการ คือ

1.การเลือกลักษณะเด่นและการคำนวณค่า

2.ทางเลือกของกฎการตัดสินใจ(decision rule)

ปัญหาแรกที่มีความสำคัญมากและยากมากกว่าเราจึงควรเริ่มต้นที่ปัญหาที่สองก่อนเพราะมีแนวความคิดหลายๆอย่างซึ่งจะช่วยในการอธิบายในหัวข้อการเลือกลักษณะเด่น

5.3.การรู้จำเสียงพูด(Speech recognition)

5.3.1.ประเภทของการจำเสียงพูด

การจำเสียงพูดโดยทั่วไปนิยมแบ่งออกเป็น 4 กลุ่มโดยเรียงลำดับตามความยากดังต่อไปนี้

5.3.1.1.การรู้จำแบบแยกคำ(Isolated-word recognition)เป็นการจำโดยต้องมีช่วงหยุดของแต่ละคำ

5.3.1.2.การจำเฉพาะคำ(Word spotting) ตรวจจับคำเฉพาะใดๆที่ปรากฏในประโยคนั้น

5.3.1.3.การจำแบบต่อเนื่อง(Connected speech recognition) รู้จำคำพูดได้ทั้งประโยคโดยไม่ต้องมีช่วงหยุดระหว่างคำ

5.3.1.4.การเข้าใจเสียงพูด(Speech understanding) เป็นแบบสมบูรณ์ของกลุ่มที่ 3 โดยสร้างแหล่งข้อมูลของการออกเสียงและหลักของภาษาจุดประสงค์หลักก็คือ การพูดที่ไม่ต้องถูกต้องมากนักแต่ให้เข้าใจความหมายของประโยคนั้นได้ในทางทฤษฎี ระบบเข้าใจเสียงพูดจะตัดทอนเอาเฉพาะความหมายของประโยคโดยไม่เข้มงวดเรื่องไวยากรณ์

5.3.2.การจำแนกแบบแยกคำ(Isolated-word recognition)

การจำแบบแยกคำเป็นจุดเริ่มต้นของการพัฒนาการจำเสียงพูดจนเป็นการจำเฉพาะคำและการจำแบบต่อเนื่องตามลำดับ ช่วงหยุดของแต่ละคำพูดทำให้การรู้จำคำง่ายขึ้น เพราะสามารถแยกแยะจุดสิ้นสุดของแต่ละคำ(หมายรวมถึงจุดเริ่มต้นและจุดสิ้นสุดของคำนั้น) ทำให้ผลที่เกิดจากความไม่ชัดเจนของการพูดลดน้อยลงไปได้มาก การจำแบบแยกคำขณะที่ออกเสียงจะต้องระมัดระวังมาก เพราะต้องการช่วงหยุดระหว่างการพูดแต่ละคำทำให้การพูดขาดความไพเราะของภาษาไป ไม่เหมือนกับวิธีการอื่นๆซึ่งสามารถพูดเป็นธรรมชาติได้มากกว่าและใช้ความระมัดระวังในการพูดน้อยกว่า

การจำแนกแบบแยกคำมีหลักการรู้จำโดยใช้ลักษณะของสัญญาณเสียงพูดดังต่อไปนี้

5.3.2.1.ขนาดหรือกำลังต่อเวลา

5.3.2.2.อัตราตัดผ่านศูนย์(zero-crossing rate)

5.3.2.3.สมดุลย์กลุ่มความถี่รวม(gross spectrum balance) เป็นการเปรียบเทียบพลังงานของกลุ่มความถี่สูงกับกลุ่มพลังงานความถี่ต่ำ

5.3.2.4.ความถี่สำคัญอาจมาในรูปของ

ก.ความถี่จากการแปลง DFT

ข.F1,F2,F3

ค.พารามิเตอร์ LPC หรือสัมประสิทธ์ PARCOR

ง.เอาท์พุทของฟิลเตอร์แบงค์(filter-bank) วิธีนี้เป็นที่นิยมกันที่สุดเพราะราคาถูกและความเร็วสูง

ขนาดนอกจากเป็นตัวบอกจุดสิ้นสุดของแต่ละคำและยังเป็นตัวชี้ความแตกต่างของพยัญชนะและสระอีกด้วย

5.3.3.การตรวจจับจุดสิ้นสุด(Endpoint detection)

ปัญหาพื้นฐานของระบบรู้จำเสียงพูด คือ เสียงที่เปล่งออกมาในขณะตรวจสอบนั้นอาจจะไม่เหมือนกันตอนที่เรียนรู้เสมอไป เสียงที่เปล่งออกมา 2 ครั้งจะมีระยะเวลาที่ต่างกันและระยะระหว่างการออกเสียงที่ไม่แน่นอน หมายความว่าคุณสมบัติที่ขึ้นกับเวลาจะทำให้การเปรียบเทียบไม่สามารถทำได้ เพราะคำอ้างอิงกับคำที่มาเปรียบเทียบเวลาจะต่างกัน ในกรณีเช่นนี้ถึงแม้ว่าจะเปรียบเทียบกับคำที่ไม่รู้กับรูปแบบที่ถูกต้องของตัวมันเองแล้วก็ตาม แต่ก็อาจจะให้ผลแตกต่างได้พอๆกับที่เปรียบเทียบกับที่เปรียบเทียบกับรูปแบบอ้างอิงของคำอื่นซึ่งจะเป็นปัญหา ถ้าคำที่ถูกเปรียบเทียบเป็นการเปรียบทั้งคำต้นแบบ แทนที่จะเปรียบเทียบบางส่วนของคำโดยการเปรียบเทียบส่วนต่อส่วน ความจริงแม้ในระหว่างเสียงสองเสียงที่ใช้ในการเรียนรู้ รายละเอียดของเวลายังต่างกันเลย เมื่อเก็บข้อมูลเสียงจากหลายๆคนจึงต้องทำการเฉลี่ยในระหว่างการเรียนรู้ก่อนนำไปเปรียบเทียบ ในขั้นตอนการรู้จำจะไม่มีความ

ในหลายๆกรณี ความแม่นยำของการปรับจะขึ้นอยู่กับความแม่นยำของการแสดงจุดสิ้นสุดของคำ ความผิดพลาดในการรู้จำจุดเริ่มต้นและจุดสิ้นสุดของคำจะทำให้เกิดความยุ่งยากขึ้นโดยความผิดพลาดของจุดสิ้นสุดจะเกิดขึ้น ในคำที่เริ่มต้นและลงท้ายด้วยพยางค์ที่ออกเสียงเบา ผู้พูดบางคนอาจจบประโยคด้วยเสียงสั้นพยางค์เสียงสั้นนี้จะทำให้เกิดความผิดพลาดในการเปรียบเทียบคำ

ในห้องทดลอง การรู้จำจุดสิ้นสุดอาจจะทำได้ง่าย เพราะข้อมูลเสียงพูดถูกรวบรวมภายใต้การควบคุมสภาวะ เพื่อหาจุดสิ้นสุด ยิ่งกว่านั้นผู้พูดที่บันทึกข้อมูลในการทดลอง จะทำด้วยความระมัดระวัง แต่นอกห้องทดลอง ไม่ได้เป็นอย่างนั้น เสียงพูดจะมีการรบกวนจากรอบข้างเพราะไม่มีการควบคุมสภาวะ และผู้พูดไม่มีแรงจูงใจที่จะช่วยในการรู้จำ

คุณสมบัติหลักของจุดสิ้นสุด คือพลังงานดังนั้น วิธีง่ายๆที่ใช้ในการตรวจจับจุดสิ้นสุดก็คือ การเปรียบเทียบพลังงานของสัญญาณกับค่าหรือระดับที่ตั้งไว้และจุดสิ้นสุดจากตำแหน่งของคำที่มีพลังงานต่ำกว่าระดับที่ตั้งไว้ ค่าระดับที่ตั้งไว้นี้ต้องมีความเข้มระดับหนึ่ง ถ้าความแรงของสัญญาณเสียงพูดอ่อน อาจจะต้องตั้งระดับต่ำๆ

บทที่ 6
นิวรอลเน็ตเวอร์ก(Neural Networks)

6.1 ลักษณะทั่วไปของนิวรอลเน็ตเวอร์ค

นิวรอลเน็ตเวอร์คเป็นโมเดลคอมพิวเตอร์แบบหนึ่งที่ได้รับการออกแบบและสร้างให้เลียนแบบเซลสมองหรือลักษณะโครงสร้างของนิวรอนของมนุษย์ซึ่งเป็น การประยุกต์ใช้ความรู้ และเทคโนโลยีทางด้านชีววิทยาและชีวฟิสิกส์ที่เกี่ยวข้องกับเซลสมองมนุษย์(Neurophysiology) โดยเทคโนโลยีทางด้านนี้มีชื่อเรียกหลายชื่อด้วยกันเช่น

- ระบบเซลสมองจำลอง(Artificial Neural systems-ANS)
- โมเดลเชื่อมโยง(Connectionist Models)
- คอนเน็คชั่นนิซึ่ม(Connectionism)
- ระบบประมวลผลขนานแบบกระจายอำนาจ(Parallel Distributed Processing-PDP)
- ระบบเครือข่ายนิวรอล(Neural Networks)

ลักษณะของนิวรอลอิเล็กทรอนิกส์จะถูกเชื่อมโยงต่อซึ่งกันและกันคล้ายคลึงกับการที่นิวรอนของมนุษย์เชื่อมต่อกันเป็นร่างแหด้วยลักษณะของการเชื่อมต่อกันจำนวนมากนี้จึงทำให้ระบบเครือข่ายนิวรอลถูกเรียกว่า"โมเดลเชื่อมโยง" (Connectionist Models)หรือ "ระบบคอนเน็กชั่นนิซึม"(Connectionism)

องค์ประกอบที่สำคัญของโครงสร้างการทำงานของเซลสมองมนุษย์ที่ทำให้มีการคิดค้นรูปแบบของระบบนิวรอลเน็ตเวอร์คสรุปได้ดังนี้คือ

6.1.1.ง่ายแต่จำนวนมหาศาล

โครงสร้างพื้นฐานของเซลสมองมนุษย์มีฟังก์ชันการทำงานแบบง่ายๆแต่อาศัยว่ามีเซลสมองจำนวนมหาศาล(สมองมนุษย์มี 10^{11} นิวรอนเซล) นิวรอลเน็ตเวอร์คก็ควรจะประกอบไปด้วยหน่วยประมวลผลจำนวนมากๆเช่นกัน โดยแต่ละเซลเป็นหน่วยประมวลผลขนาดเล็กซึ่งมีโครงสร้างหรือมีฟังก์ชันการทำงานอย่างง่ายๆ

6.1.2. เครือข่ายการเชื่อมโยง

เซลสมองมนุษย์จำนวนมากนั้นเชื่อมโยงต่อกันเป็นเครือข่าย การเชื่อมต่อกันถึงกันนั้นมีเงื่อนไขในรูปของน้ำหนักการต่อเชื่อม(Weighted Connections) และน้ำหนักในการเชื่อมโยงระหว่างหน่วยประมวลผลนี้จะเป็นการแสดงเงื่อนไขหรือบรรจุความรู้ในรูปแบบต่างๆเอาไว้

6.1.3. ช่วยกันทำและการกระจายอำนาจ

เซลสมองจำนวนมากมายช่วยกันทำงานในลักษณะขนาน(Massive Parallel Processing) ถึงแม้ว่าแต่ละเซลจะมีฟังก์ชันการทำงานแบบง่ายๆแต่ละเซลทำงานและรับผิดชอบงานในส่วนของตัวเองไปพร้อมๆกันการควบคุมการทำงานของระบบนิวรอลเน็ตเวอร์คจึงพยายามเลียนแบบการทำงานในลักษณะขนานและกระจายอำนาจนี้(Parallel Distributed Processing) แม้ว่าเซลนิวรอนบางเซลไม่สามารถทำงานได้ตามปกติ(เช่น เซลบางเซลเสียหายหรือตายไป) เน็ตเวอร์คก็ยังคงสามารถทำงานโดยส่วนรวมได้ถูกต้องคุณสมบัตินี้เรียกว่า " คุณสมบัติของการทนความผิดพลาด" (Fault Tolerance)

6.1.4. ความสามารถในการเรียนรู้แบบอัตโนมัติ

ระบบนิวรอลเน็ตเวอร์คีควรจะมีความสามารถในการเรียนรู้จากประสบการณ์เรียนรู้ลักษณะหรือกฎเกณฑ์ทั่ไปจากตัวอย่าง การเรียนรู้นี้เน้นที่วิธีการสร้างความรู้ภายในระบบ(Internal Representation) ในแบบอัตโนมัติ โดยขบวนการเรียนรู้จะเป็นการปรับค่าน้ำหนักและค่าพิกัดที่อยู่ภายในระบบให้สอดคล้องกับกลุ่มตัวอย่างของชุดข้อมูลอินพุท/เอาต์พุทที่ป้อนเข้าสู่ระบบในช่วงจังหวะของการเรียนรู้ กล่าวคือ น้ำหนักของจุดเชื่อมโยงจะต้องถูกปรับแต่ง/เปลี่ยนแปลงจนกระทั่งฟังก์ชันการทำงานของทั้งระบบเป็นไปตามลักษณะพิเศษของกลุ่มตัวอย่างที่ป้อนนั้น

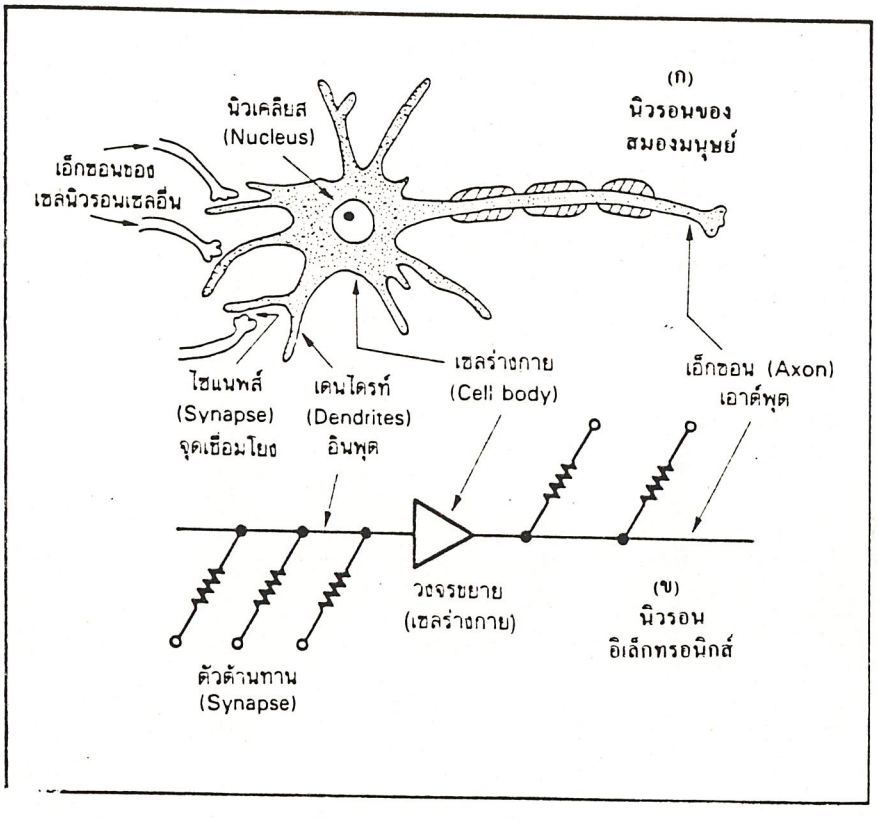
6.2 ชีวฟิสิกส์เบื้องต้นเกี่ยวกับนิวรอน

เราอาจจำลองแบบโครงสร้างพื้นฐานของนิวรอนหรือเซลสมองของมนุษย์ได้ในรูปของวงจรอิเล็คทรอนิกส์แบบง่ายๆหรือที่นิยมกันมากคือการจำลองแบบการทำงานของเครือข่ายนิวรอลในรูปของฟังก์ชันคณิตศาสตร์โดยจะกล่าวถึงลักษณะของเซลนิวรอนอย่างคร่าวๆดังแสดงในรูปที่ 6.1

ในรูปที่ 6.1(a) จะแสดงองค์ประกอบสำคัญๆของเซลสมองมนุษย์ เซลนิวรอรอนจะประกอบด้วยส่วนหลักๆ 4 ส่วนที่สำคัญในการส่งสัญญาณข้อมูลคือ ไซแนพส์(Synapse), เดนไดรต์(Dendrite), เซลร่างกาย(Cell body), เอ็กซอน(Axon) โดยเซลนิวรอนจะรับข้อมูลอินพุทจากนิวรอนเซลอื่นโดยผ่านทางจุดเชื่อมโยงระหว่างเซลที่เรียกว่า "ไซแนพส์" สัญญาณข้อมูลจากไซแนพส์จะถูกส่งผ่าน "เดนไดรต์" ซึ่งเป็นส่วนที่จะระโยงระยางรอบข้างเซลร่างกาย สัญญาณข้อมูลอินพุทจะได้รับการประมวลผลบางอย่างตามขบวนการที่เกิดขึ้นภายในเซลร่างกายแล้วสัญญาณเอาท์พุทจากเซลนิวรอนจะถูกส่งออกทางส่วนของเซลที่เรียกว่า " เอ็กซอน" สัญญาณนี้จะกระโดดข้ามผ่านไซแนพส์ (ไปด้วยเงื่อนไขบางอย่าง) ซึ่งเป็นส่วนอินพุทของเซลนิวรอนเซลอื่นต่อไป ส่วน

ในรูปที่ 6.1(b) เป็นวงจรอิเล็คทรอนิกส์ที่เลียนแบบการทำงานของเซลนิวรอนโดยวงจรขยายแบบอนาลอกรับสัญญาณอินพุทผ่านทางตัวความต้านทานแล้วส่งสัญญาณเอาต์พุทไปยังวงจรขยายตัวอื่นซึ่งผ่านทางตัวต้านทานเหมือนกัน

สมองมนุษย์ประกอบด้วยเซลนิวรอน(ดังรูป 6.1(a)) จำนวน 10^{11} นิวรอน (หนึ่งแสนล้านเซล) นิวรอนแต่ละเซลมีจุดเชื่อมโยงอินพุทและเอาต์พุท(ไซแนพส์) ประมาณ 10^4 การเชื่อมโยง ไซแนพส์ซึ่งเป็นจุดเชื่อมโยงของเอ็กซอน (อุปกรณ์เอาต์พุท) ของนิวรอนเซลหนึ่งและเดนไดรต์ (อุปกรณ์อินพุท) ของอีกเซลหนึ่งมีฟังก์ชันการทำงาน 2 ลักษณะคือ "ไซแนพส์ด้านบวก" (Excitatory Synapses) และ "ไซแนพส์ด้านลบ"(Inhibitory Synapses) ไซแนพส์ในด้านบวกเป็นไซแนพส์ชนิดที่ทำให้สัญญาณเอาต์พุทที่ส่งผ่านมามีความถี่สูงขึ้น ส่วนไซแนพส์ในด้านลบจะทำให้สัญญาณเอาต์พุทมีความถี่ต่ำลง นอกจากนี้ความถี่ของสัญญาณเอาต์พุทยังขึ้นอยู่กับความแรงหรือปริมาณของสัญญาณอินพุทและเงื่อนไขที่แสดงในรูปของน้ำหนักของจุดเชื่อมโยงไซแนพส์ด้วยคุณสมบัติในการจดจำ(Memory) และการประมวลผล (Processing) ของสมองมนุษย์จะขึ้นอยู่กับโครงสร้างของการเชื่อมโยงและน้ำหนักของจุดเชื่อมโยงไซแนพส์เหล่านี้



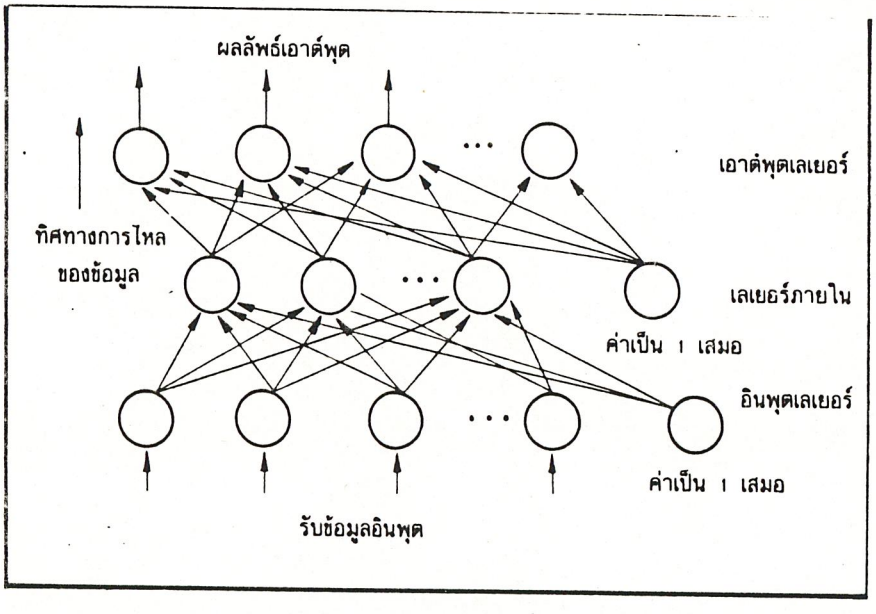
รูปที่ 6.1 (a) แสดงโครงสร้างเซลพื้นฐานของเซลนิวรอนของสมองมนุษย์

(b) แสดงวงจรอิเล็กทรอนิคส์แบบอะนาลอกเปรียบเทียบการทำงาน ของนิวรอนเซลสมอง

โมเดลเชิงคณิตศาสตร์ โดยเฉพาะอย่างยิ่งฟังก์ชันทางพีชคณิต มักจะถูกนำมาใช้อธิบายลักษณะโครงสร้างการทำงานของนิวรอนเนื่องจากมีความหมายที่เด่นชัดและประหยัดถ้อยคำที่อาจจะกำกวมเหมาะกับการพัฒนาระบบคอมพิวเตอร์ทางด้านซอฟต์แวร์และฮาร์ดแวร์

6.3 การประยุกต์ใช้งานนิวรอลเน็ตเวอร์ค

ระบบเครือข่ายนิวรอลถูกนำไปประยุกต์ใช้งานที่เป็นประโยชน์ในหลายด้าน เช่น การใช้เพื่อการรู้จำ/การรับรู้ (Pattern Recognition), การใช้งานเป็นหน่วยความจำแบบ Associative, การใช้งานเพื่อแก้ไขปัญหาการจัดวาง (Combinatorial Problem) เป็นต้น การประยุกต์ใช้งานที่ประสบผลสำเร็จมากที่สุดคือในงานของการรู้จำซึ่งอาจจะกล่าวได้ว่าข้อดีของการใช้ระบบนิวรอลเน็ตเวอร์คที่เห็นชัดเจนคือ ความสามารถในการแก้ปัญหาที่ไม่จำเป็นต้องมีขบวนการที่มีนิยามที่เด่นชัดเหมือนปัญหาเชิงคณิตศาสตร์ทั่วไป เช่น ปัญหาที่ต้องเกี่ยวข้องกับการแปลข้อมูลที่มีความซับซ้อนและไม่สามารถกำหนดให้แน่ชัดได้ว่าฟังก์ชันของการแปลเป็นอย่างไรหรือปัญหาที่ต้องมีการคาดเดาคำตอบในขณะที่ข้อมูลอินพุทอาจมีความผิดเพี้ยนหรือเต็มไปด้วยสิ่งรบกวน แต่ตรงกันข้าม ระบบเครือข่ายนิวรอนกลับแก้ไขปัญหาที่เพียงแต่มีการรวบรวมชุดตัวอย่างของปัญหา/คำตอบ และระบบจะพยายามปรับปรุงข้อมูลความรู้ภายในเพื่อให้สามารถให้คำตอบการแก้ปัญหาที่มีลักษณะคล้ายคลึงกับตัวอย่างที่เรียนรู้ไปแล้ว



รูปที่ 6.2 โครงสร้างไดอะแกรมของนิวรอลเน็ตเวอร์คแบบ " แบคพรอพาเกชั่นเน็ตเวอร์ค"

6.4 เน็ตเวอร์คแบบส่งผลถอยหลัง(Backpropagation Network)

นิวรอลเน็ตเวอร์คแบบนี้จะใช้งานได้ดีในการแก้ปัญหาที่เกี่ยวกับการรู้จำภาพที่มีความซับซ้อนและปัญหาที่เกี่ยวกับฟังก์ชันการแปลแบบซับซ้อน มีโครงสร้างไดอะแกรมดังแสดงในรูปที่ 6.2 เน็ตเวอร์คแบบนี้จะประกอบไปด้วยเซลนิวรอนที่เรียงกันอยู่อย่างน้อย 3 เลเยอร์ (layers:ชั้น) นั่นก็คือ อินพุทเลเยอร์, เลเยอร์ภายใน (Hidden layer) และเอาต์พุทเลเยอร์ ทิศทางการไหลของข้อมูลจะเป็นลักษณะเคลื่อนไปข้างหน้าจากอินพุทเลเยอรีผ่านเลเยอร์ชั้นในไปสู่เอาต์พุทเลเยอร์ เน็ตเวอร์คในลักษณะนี้จึงมีชื่อเรียกตามลักษณะโครงสร้างดังกล่าวว่า " เน็ตเวอร์คหลายชั้นเคลื่อนหน้า" (Multilayer feedforward network) โดยลักษณะการเรียนรู้ในเน็ตเวอร์คนี้จะต้องมีผู้ช่วยภายนอกในการป้อนตัวอย่างโดยลักษณะการเรียนรู้แบบนี้จะเรียกทางเทคนิคว่า " การเรียนรู้แบบมีผู้ช่วย" (Supervised Learning)

6.4.1. การทำงานของแบคพรอพาเกชั่นเน็ตเวอร์ค

ก่อนที่จะนำนิวรอลไปใช้งานเพื่อการรู้จำ (Pattern Recognition) เราจะต้องทำการฝึกสอนเน็ตเวอร์คก่อนว่าจะรู้จำเรื่องอะไร ขั้นตอนในการเรียนรู้มีดังนี้คือ เน็ตเวอร์คจะได้รับชุดตัวอย่างข้อมูลคู่อินพุท-เอาต์พุท (ได้รับการสอนว่า ถ้าอินพุทแบบนี้เอาต์พุทต้องแบบนี้เช่นเดียวกัน) สำหรับข้อมูลแต่ละคู่เน็ตเวอร์คจะดำเนินงานของการเรียนรู้เป็นวัฏจักรที่ประกอบด้วย 2 ขั้นตอน คือ "การส่งผลและการปรับค่า" (Propagation and Adaptation) ข้อมูลอินพุทที่ป้อนเข้าสู่นิวรอลในชั้นของอินพุทเลเยอร์จะถูกประมวลผลและส่งผลลัพธ์ต่อไปยังเลเยอร์ชั้นต่อไปจนถึงเอาต์พุทเลเยอร์ ค่าเอาต์พุทที่ได้ (จากการประมวลผลของเน็ตเวอร์ค) จะถูกนำไปเปรียบเทียบกับตัวอย่างเอาต์พุท (ที่กำหนดมา, เอาต์พุทที่ต้องการ) ค่าของความผิดพลาด (ระหว่างเอาต์พุททั้งสองแบบ) จะถูกคำนวณขึ้นมาสำหรับเซลนิวรอนแต่ละตัว

ค่าของความผิดพลาดจะถูกส่งถอยหลังกลับไปจากเลเยอร์เอาต์พุทไปยังโหนดต่างๆของเลเยอร์ภายในแต่ละโหนดนั้นจะรับค่าของความผิดพลาดนั้นเพียงบางส่วนขึ้นอยู่กับว่าโหนดหรือนิวรอนเซลนั้นว่าเป็นตัวส่งผลมากหรือน้อยไปสู่เอาต์พุทนั้น ขบวนการของการส่งค่าความผิดพลาดกลับไปนั้นจะทำซ้ำสำหรับเลเยอร์ถัดลงมาอีก จนกระทั่งทุกโหนดในเน็ตเวอร์คได้รับส่วนแบ่งของค่าความผิดพลาดนั้นหลังจากนั้นน้ำหนักเชื่อมต่อของแต่ละนิวรอนเซลจะถูกปรับค่าไปตามความมากหรือน้อยของสัญญาณค่าความผิดพลาดที่ได้รับเพื่อที่จะทำให้เน็ตเวอร์คปรับตนเองไปอยู่ในสถานะที่จะทำให้ตัวอย่างข้อมูลคู่อินพุท-เอาต์พุทได้รับการบันทึกโดยเข้ารหัสไว้ วัฏจักรของการส่งผลและการปรับค่าจะดำเนินต่อไปจนกระทั่งค่าของความผิดพลาดต่ำกว่าค่าที่กำหนดไว้ค่าหนึ่งแล้วจึงหยุด

จุดสำคัญของการเรียนรู้ในช่วงจังหวะที่เน็ตเวอร์ค ได้รับการฝึกฝนนั้นคือการที่นิวรอนเซลต่างๆในแต่ละเลเยอร์ทำการปรับค่าน้ำหนักของโหนดแตกต่างกันไปเพื่อที่จะทำการรู้จำลักษณะที่แตกต่างของข้อมูลอินพุท ดังนั้นหลังจากที่ได้รับการฝึกฝนแล้วเมื่อจะนำเน็ตเวอร์คไปใช้งานเพื่อรู้จำข้อมูลอินพุทชุดหนึ่งซึ่งแม้ว่าจะมีสัญญาณรบกวนหรือมีข้อมูลที่ไม่สมบูรณ์ หน่วยต่างๆของเลเยอร์ภายในจะให้ผลลัพธ์เอาต์พุทได้ถูกต้อง ถ้าอินพุทที่ป้อนเข้ามีลักษณะใกล้เคียงกับแพตเทอร์น (Pattern) ที่เคยเรียนรู้ไปแล้ว ในทางตรงกันข้าม ถ้าลักษณะข้อมูลไม่มีความใกล้เคียงกับแพตเทอร์นที่เคยผ่านการฝึกฝน หน่วยนิวรอนหน่วยนั้นก็จะไม่ส่งผลลัพธ์เลย

สรุปขั้นตอนของการฝึกสอน (Training) สำหรับนิวรอนเน็ตเวอร์คแบบส่งผลถอยหลังกลับ (Backpropagation Network) นี้โดยเขียนเป็นรูปแบบของโปรแกรมได้ดังนี้

- ขั้นตอนการฝึกฝน

```
begin
    while ค่าความผิดพลาดของเน็ตเวอร์คมีค่ามากกว่าค่าที่กำหนดไว้
    begin
        1. รับข้อมูลตัวอย่างอินพุท
        2. ทำการประมวลผล (propagate forward) โดยส่งผลลัพธ์
        ไปยังเอาต์พุทเลเยอร์
        3. ทำการคำนวณค่าความผิดพลาดระหว่างเอาต์พุทที่ได้จาก
        เน็ตเวอร์คกับข้อมูลตัวอย่างเอาต์พุทที่ต้องการ
        4. ส่งค่าความผิดพลาดกลับเข้าไปในเน็ตเวอร์ค (Error
        Backpropagation)
        6. ทำการปรับค่าน้ำหนักมากหรือน้อยตามค่าความผิดพลาด
    end
end.
```

- ขั้นตอนการประยุกต์ใช้งาน

begin

1. ทำการรับข้อมูลอินพุทของปัญหา

2. ทำการประมวลผล (Propagate forward) โดยส่งผลลัพธ์ไปยังเอาต์พุทเลเยอร์

3. ให้คำตอบที่เอาต์พุทเลเยอร์

end.

บทที่ 7
การทดลองและสรุปผลการทดลอง

7.1. ขั้นตอนการทำงาน แบ่งออกเป็น 2 ขั้นตอนคือ

7.1.1 การหาค่าพารามิเตอร์ของเสียงพูด

7.1.1.1 การบันทึกเสียงพูด

- บันทึกเสียงพูดโดยใช้ sound blaster 16 ASP ของ Creative Labs. ซึ่งไฟล์เสียงจะเก็บอยู่ในรูปไฟล์ .WAV โดยพูดผ่านทาง microphone ด้วยเสียงของเลขศูนย์ถึงเลขเก้า ไฟล์ที่ใช้บันทึกคือ ไฟล์ WREC.EXE ซึ่งมีรูปแบบดังนี้

WREC *Filename* /R:xx /A:xx /S:xx /M:xx /T:xx

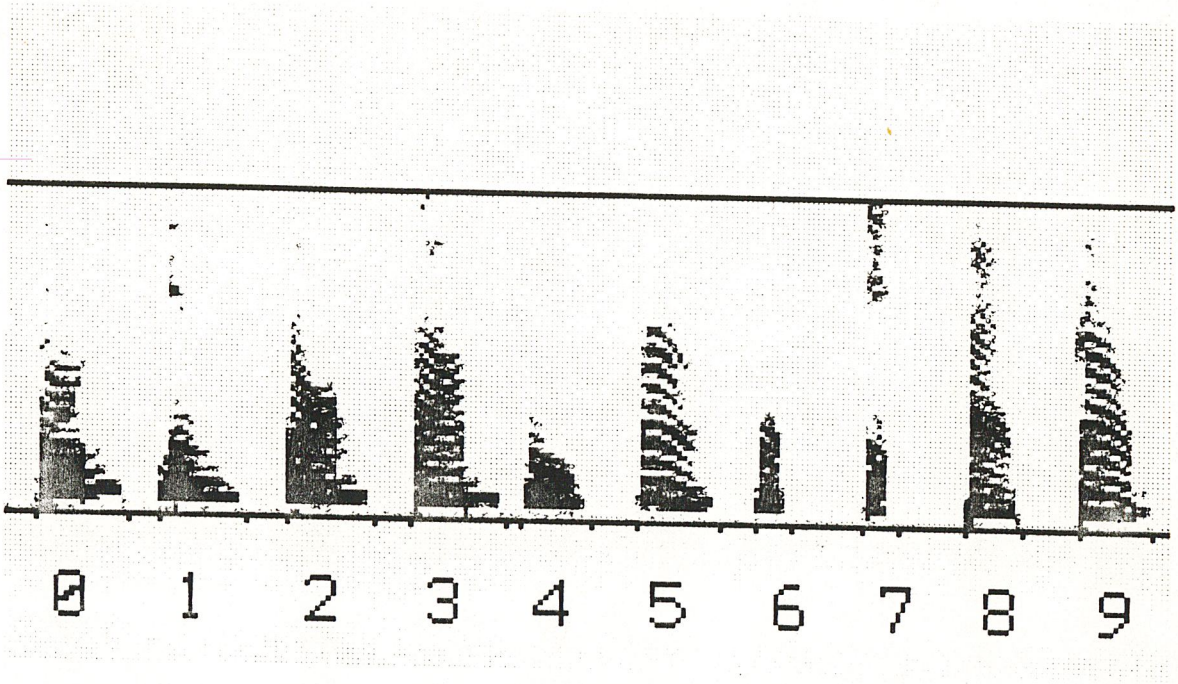
เช่น WREC 90.wav /R:8 /A:MIC /S:11025 /M:MONO /T:5

ความหมาย :

- *Filename* .WAV
- /R:xx คือ จำนวน bit ที่ใช้ในการ sample โดย xx = 8
- /A:xx คือ อุปกรณ์ที่ใช้ในการบันทึก xx = MIC (microphone)
- /S:xx คือ จำนวน sampling rate xx = 11025 Hz
- /M:xx คือ โหมดที่ใช้ในการทำงาน xx = MONO
- /T:xx คือ เวลาที่ใช้ในการบันทึก xx = 1 - 65535 second

7.1.1.2. แปลงสัญญาณเสียงจาก Time domain เป็น Frequency domain

- การแปลงสัญญาณเสียงพูดจาก time domain ---> frequency domain จะใช้ การแปลงฟาสท์ฟูเรียร์(Fast fourier Transform หรือ FFT)ที่มีลำดับสัญญาณทีละ 256 จุด เป็นอินพุทที่ได้มาจากไฟล์เสียง .WAV ที่ได้มาจากการบันทึกจากขั้นตอนที่แล้วซึ่งแสดงดังรูปที่ 7.1 เป็นการวาด image(ภาพ) ที่แสดงค่าแอมพลิจูดของสัญญาณเสียงที่บันทึกไว้ของเลข 0 - เลข 9 โดย image จะแสดงอยู่ในแนวแกน 3 แกนคือ แกนของแอมพลิจูด, แกนของความถี่, แกนของเวลา



รูปที่ 7.1 แสดงค่าแอมพลิจูดของเสียงพูดเลข 0 - เลข 9

7.1.1.3. หาขอบเขตของเสียงพูด

- การหาขอบเขตของคำพูดเป็นการหาช่วงของสัญญาณที่มีเสียงพูดโดยหาได้ได้จากค่าพลังงานของเสียงพูด(Magnitude) ซึ่งคำนวณได้จากสัญญาณในช่วง time domain จากสูตร

Magnitude = (ค่า real)^2 + (ค่าimagine)^2

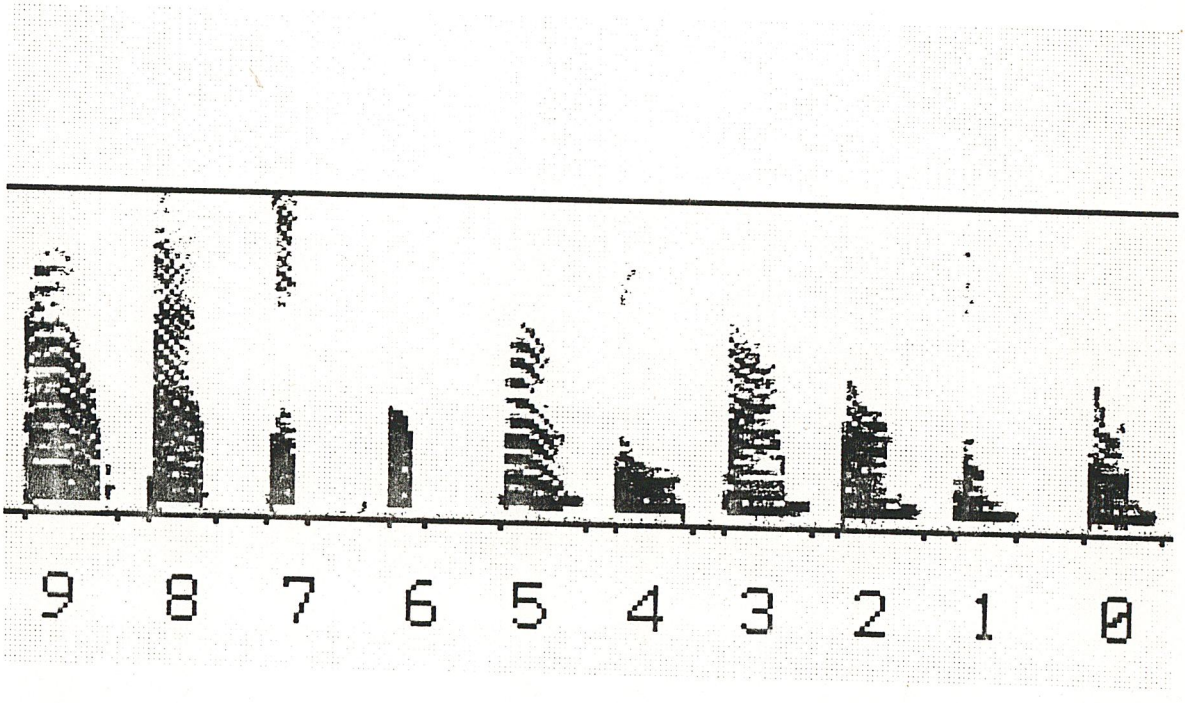
และกำหนดค่าพลังงานที่เป็นของจุดเริ่มต้นและจุดสิ้นสุดของคำ (Threshold)ในที่นี้ให้มีค่าเป็น 8.0 โดยพิจารณาว่าถ้าค่าพลังงานที่ได้มากกว่าหรือเท่ากับค่าที่กำหนดไว้จะเป็นจุดเริ่มต้นของคำและหลังจากนั้นค่าที่ต่ำกว่าจุดที่กำหนดค่าถัดไปจะเป็นจุดสิ้นสุดของคำแล้วทำการเก็บภาพ (image) ของช่วงคำนั้นหรือค่า magnitude ของช่วงนั้นไว้ในไฟล์ .PAT ซึ่งจะใช้เป็นพารามิเตอร์ของเสียงพูดที่จะนำไปใช้ในขบวนการอื่นต่อไป

- จุดเริ่มต้นคำมากกว่าหรือเท่ากับค่า Threshold (8.0)
- จุดสิ้นสุดคำจะเป็นจุดถัดมาที่มีค่าน้อยกว่าค่า Threshold (8.0)

ข้อมูลที่เก็บอยู่ในไฟล์ .PAT จะประกอบด้วย

- ข้อมูลลำดับแรกจะเป็นชนิดของตัวเลขที่มี image เก็บอยู่
- ข้อมูลลำดับที่สองจะเป็นขนาดของ image ของตัวเลขนั้น
- ข้อมูลอันดับสามขึ้นไปจะเป็นข้อมูลของ image ของเลขชนิดนั้น

ดังรูปที่ 7.2 จะเป็นรูปที่แสดง image ของ magnitude ของเสียงแต่ละคำพูดจากเลข 0 - เลข 9 ที่ได้ผ่านการหาขอบเขตของเสียงพูดแล้วซึ่งจะเป็นการตัดสัญญาณที่ไม่สำคัญออกไป เหลือไว้เพียงช่วงของเสียงที่มีความสำคัญ



รูปที่ 7.2 แสดงเสียงพูดที่ผ่านการหาขอบเขตของคำแล้ว (เลข 0 - เลข 9)

7.1.2. ประยุกต์ใช้งานกับ Neural Networks

7.1.2.1. ฝึกสอน Neural Network (Training)

- Networks ที่ใช้จะเป็น Backpropagation Network (BPN) ประกอบด้วย layers 3 ชั้นคือ input layers 1920 node (30x64), hidden layer 7 node, output layer 8 node ก่อนที่มีการ train นั้นจะต้องมีการกำหนด output ที่ต้องการก่อนโดยกำหนดให้ output มีขนาด 8 bit เป็นลักษณะของเลขฐานสองคือ 0 กับ 1 ซึ่งจะมีรูปแบบดังนี้

0 = 1 0 0 1 1 0 0 0

1 = 1 0 0 0 0 1 0 1

2 = 1 1 0 1 0 0 1 1

3 = 1 1 0 1 1 0 0 1

4 = 0 0 0 1 0 1 1 1

5 = 1 1 1 0 1 1 0 0

6 = 0 0 0 0 1 1 1 0

7 = 0 1 0 0 0 0 0 0

8 = 1 1 0 0 1 1 1 1

9 = 1 1 1 0 0 0 1 0

ในขั้นตอนการเรียนรู้แบ่งการทำงานออกเป็น 2 cycles คือ

1. Propagation forward cycle - นำเสียงพูดใน ไฟล์ .PAT และค่าของ weight ที่ถูกสุ่มขึ้นนำมาผ่านขบวนการคำนวณจาก input layer-->hidden layer-->output layer ตามลำดับจนได้ผลลัพธ์ที่ได้จากการคำนวณจริงออกมา (Actual output) ซึ่งมาจากสูตร

$$sum = \sum_{i=0}^{n} X_i \cdot W_i$$

$$output = 1 / 1 + e^{-sum}$$

โดยที่ X คือ ค่าอินพุทของไฟล์เสียง .PAT

W คือ ค่า weight ซึ่งเป็นค่าที่ถูกสุ่มขึ้นมาในตอนเริ่มของการ training

2. Error backpropagation cycle -: นำผลลัพธ์ที่ต้องการลบด้วยผลลัพธ์ที่ได้จากการคำนวณจะได้เป็นค่า error ออกมาซึ่งเป็นค่าที่นำไปปรับ weight ในขบวนการแรกที่เชื่อมต่อระหว่าง output layer กับ hidden layer และ hidden layer กับ input layer ซึ่งมีทิศทางจาก output –>input layer

error = Desire output (ผลลัพธ์ที่ต้องการ) - Actual output(ผลลัพธ์ที่ได้จริง)

error < Training Threshold (1.0) แสดงว่า networks เรียนรู้ตัวอย่างได้หมดสิ้น

ขบวนการ 2 cycles จะวนรอบทำงานจนกว่า error ที่ได้จะต่ำกว่าค่าที่ตั้งไว้ซึ่งเป็นการแสดงว่า network ได้เรียนรู้ในตัวอย่างที่เราป้อนเข้าไปเสร็จสิ้นแล้วโดยจะมีการเก็บค่า weight ที่ได้จากการเปลี่ยนแปลงครั้งสุดท้าย (ไฟล์ .WGH) ซึ่งก็คือค่า weight ที่ทำให้ network สามารถเรียนรู้ตัวอย่างได้สมบูรณ์และ weight ที่ได้จะนำไปใช้ในขบวนการอื่นต่อไป

7.1.2.2 ประยุกต์ใช้งานกับเสียงทดสอบ

- เป็นการทดสอบเสียงพูดใหม่ที่ยังไม่เคยเรียนรู้ (Train) มาก่อนด้วยค่า weight จากขบวนการที่แล้วโดยขบวนการทำงานจะเหมือนกับการเรียนรู้ตัวอย่าง (Train) ยกเว้นในส่วนของขบวนการ error backprogation cycle ซึ่งจะไม่มีการคำนวณ error จะเป็นการนำค่า image ของ magnitude ของเสียงใหม่ที่พูดเข้ามาคำนวณกับค่า weight (ไฟล์ .WGH) ที่เก็บเอาไว้ที่ทำให้ network สามารถเรียนรู้โดยสมบูรณ์จะได้ผลลัพธ์ของการรู้จำเสียงพูดนั้นออกมา ในการพิจารณาว่า network รู้จำเสียงของเลขต่างๆได้หรือไม่นั้นพิจารณาจากสูตร

Delta (ผลต่างในแต่ละ bit) = Desire output - Actual output

-0.5 <= Delta <= +0.5 –> ผลการรู้จำ (test_value =1) เป็น 1 ถ้านอกเหนือจากนี้ให้เป็น 0 (test_value = 0) แล้วรวมผลกันทั้ง 8 bit ถ้า test_value = 8 แสดงว่ารู้จำเลขนั้นได้ ถ้าน้อยกว่าแสดงว่ารู้จำไม่ได้โดยการรายงานผลของการรู้จำจะแสดงอยู่ในรูปของเปอร์เซนต์ (%) คือ

% การรู้จำ = (เสียงที่สามารถจำได้ / จำนวนเสียงทั้งหมด) * 100

สรุปขั้นตอนของการประยุกต์ใช้งานเสียงพูดกับนิวรอลเน็ตเวอร์คแบบส่งผลถอยหลังกลับ (Backpropagation Network) นี้โดยเขียนเป็นรูปแบบของโปรแกรมได้ดังนี้

- ขั้นตอนการฝึกฝน

```
begin
    while ค่าความผิดพลาดของเน็ตเวอร์คมีค่ามากกว่าค่าที่กำหนดไว้
        begin
        1. รับข้อมูลตัวอย่างอินพุท (ไฟล์ .PAT )
        2. ทำการประมวลผล (propagate forward) โดยส่งผลลัพธ์
        ไปยังเอาต์พุทเลเยอร์
        3. ทำการคำนวณค่าความผิดพลาดระหว่างเอาต์พุทที่ได้จาก
        เน็ตเวอร์คกับข้อมูลตัวอย่างเอาต์พุทที่ต้องการ (Desire output - Actual
        output)
        4. ส่งค่าความผิดพลาดกลับเข้าไปในเน็ตเวอร์ค (Error
        Backpropagation)
        5. ทำการปรับค่าน้ำหนักมากหรือน้อยตามค่าความผิดพลาด
    end
end.
```

- ขั้นตอนการประยุกต์ใช้งาน

```
begin
    1. ทำการรับข้อมูลอินพุทของปัญหา (ไฟล์ .PAT)
    2. ทำการประมวลผล (Propagate forward) โดยส่งผลลัพธ์ไปยังเอาต์พุท
    เลเยอร์
    3. ให้คำตอบที่เอาต์พุทเลเยอร์ (%)
end.
```

7.2 สรุปผลการทดลองและปัญหา

สำหรับการทดลองครั้งนี้ที่ให้ คอมพิวเตอร์รู้จำเสียงพูดโดยใช้ neural network เข้ามา

ช่วยเป็นการทดลองให้รู้จำเสียงพูดของเลขไทยตั้งแต่ 0 - 9 โดยผลของการรู้จำเมื่อคิดออกมาเป็นเปอร์เซนต์โดยเฉลี่ยแล้วจะอยู่ที่ประมาณ 50 % ซึ่งค่อนข้างจะต่ำอยู่พอสมควรทั้งนี้ก็เนื่องมาจากสาเหตุและข้อจำกัดหลายประการด้วยกันคือ

7.2.1 จำนวนของตัวอย่างที่ใช้ในการ training ยิ่งตัวอย่างที่ใช้ในการ train network มากเท่าใดการรู้จำในเสียงของบุคคลและเสียงนั้นๆย่อมดีขึ้นตามลำดับแต่ในการทดลองนี้ผู้ทดลองใช้ตัวอย่าง 10 - 30 ตัวอย่างของเสียงเลข 0 - 9 ซึ่งสาเหตุที่ใช้ตัวอย่างน้อยเพราะว่าในขั้นตอนการเรียนรู้ของ Backpropagation Networks นั้นจะช้ามากโดยเฉพาะอย่างยิ่งเมื่อมีจำนวนอินพุทและเอาต์พุทจำนวนมากนับเป็นข้อสียประการสำคัญของ network แบบนี้

7.2.2 ค่า error ที่แสดงว่า network สามารถเรียนรู้ตัวอย่างที่ให้มาได้สมบูรณ์โดยค่านี้ต้องกำหนดให้มีค่าเหมาะสมเพียงพอที่ network จะเรียนรู้ได้ไม่ควรสูงเกินไปและต่ำเกินไปจน network ไม่สามารถเข้าถึงค่าที่กำหนดไว้ได้ซึ่งจะต้องหาค่านี้ให้ได้ซึ่งในการทดลองนี้ค่าที่ใช้ทดลองอาจเป็นค่าที่ไม่เหมาะสมในการเรียนรู้

7.2.3 เสียงพูดที่ใช้ในการทดลองต้องไม่พูดติดกันซึ่งระบบที่พัฒนาขึ้นยังไม่มีความสามารถดีพอที่จะตรวจจับเสียงพูดที่พูดต่อเนื่องกันได้ดังนั้นเสียงพูดทดสอบที่ติดกัน network จะไม่สามารถรู้จำว่าเป็นเสียงอะไร

7.2.4 ลักษณะของเสียงตัวอย่างที่ใช้ในการเรียนรู้อาจจะไม่ครอบคลุมลักษณะของเสียงที่ใช้ในการทดลองทั้งหมดทำให้ network ไม่สามารถรู้จำได้เมื่อเสียงผู้ทดลองที่ใช้ในการทดสอบ ณ เวลาและสถานที่ต่างกัน

7.2.5 ค่าที่ใช้ในการกำหนดจุดเริ่มต้นและจุดสิ้นสุดของคำในการหาขอบเขตของเสียงพูดซึ่งจะต้องหาค่าและใช้ค่าที่เหมาะสมเป็นเกณฑ์ซึ่งการทดลองนี้ค่าที่ใช้คือ 8.0 ซึ่งค่านี้อาจเป็นค่าที่ทำให้การเก็บค่าพารามิเตอร์บางเสียงพูดของตัวเลขบางตัวไม่ได้ครบในการแยกแยะลักษณะเฉพาะของตัวเลขแต่ละตัว

7.2.6 ค่าของตัวเลขที่ใช้สุ่มค่า weight ขึ้นมาในตอนแรกของการเรียนรู้ซึ่งในการทดลองนี้ใช้ตัวเลขตั้งแต่ 1 - 32767 จำนวนซึ่งมีจำนวนมากมากในการที่จะเลือกได้ค่าที่เหมาะสมที่ทำให้ network สามารถเรียนรู้ตัวอย่างที่ให้เข้าไปได้สำเร็จซึ่งค่านี้จะเปลี่ยนไปตลอดเวลาไม่แน่นอนตามจำนวนตัวเลขหรือเสียงพูดที่พูดเข้ามาใหม่ เป็นต้น

ซึ่งสาเหตุและข้อจำกัดเหล่านี้อาจจะเป็นส่วนหนึ่งที่มีผลทำให้ระบบที่พัฒนาออกมามีประสิทธิภาพไม่ดีเท่าที่ควรนอกจากนี้แล้วในการทดลองครั้งนี้ยังพบปัญหาอีกหลายประการเช่น

- ปัญหาในเรื่องหน่วยความจำของคอมพิวเตอร์ในการทำ FFT เนื่องจากว่าอินพุทที่ใช้เป็นไฟล์เสียงที่มีขนาดของไฟล์ใหญ่ในการโหลดขึ้นมาทำ FFT นั้นไม่สามารถแปลงได้ในครั้งเดียวเนื่องจากหน่วยความจำไม่พอซึ่งถ้าทำได้จะทำให้การแปลง FFT ทำได้รวดเร็วจึงเลี่ยงใช้วิธีการข้อมูลขึ้นมาแปลงทีละชุดซึ่งแก้ปัญหานี้ได้แต่ก็ต้องอ่านข้อมูลหลายรอบ

- ปัญหาเรื่องความเร็วที่ช้าของเครื่อง PC โดยเฉพาะในขั้นตอนการเรียนรู้ของ network ต้องใช้เวลานานในการรอผลลัพธ์และยิ่งในการเรียนรู้ที่มีอินพุทและเอาต์พุทจำนวนมาก เป็นต้น

กิตติกรรมประกาศ

งานวิจัยชิ้นนี้เสร็จสมบูรณ์ลงได้ด้วยความช่วยเหลือจากบุคคลหลายๆบุคคลด้วยกัน ขอขอบคุณอาจารย์ที่ปรึกษาที่ให้คำแนะนำ ปรึกษาและพร้อมทั้งให้ความช่วยเหลือทางด้านอุปกรณ์ต่างๆที่ใช้ในงานวิจัย ขอขอบคุณภาควิชาวิศวกรรมคอมพิวเตอร์ที่เอื้ออำนวยในการใช้อุปกรณ์ต่างๆของภาคฯอาทิเช่น คอมพิวเตอร์ พรินท์เตอร์ ไมโครโฟน เป็นต้น ขอบคุณเพื่อนๆทั้งในภาคฯและนอกภาคฯที่ให้ความเกื้อหนุนและบั่นทอนกำลังใจกันเสมอมา ขอบคุณห้อง project ที่เป็นแหล่งรวบรวมและรองรับทุกสภาวะอารมณ์ทั้งไฟดับและไฟไม่ดับ

และสุดท้ายขออาราธนา คุณพระศรีรัตนตรัยและสิ่งศักดิ์สิทธิ์ในสากลโลกนี้รวมทั้งกราบขอบพระคุณผู้บังเกิดเกล้าและบูรพาจารย์ทุกท่านที่ทำให้ข้าพเจ้าได้มายืนที่จุดนี้.....วิศวกร

บรรณานุกรม

1. ศ.ดร.วัลลภ สุระกำพลธร,"การประมวลผลสัญญาณเชิงตัวเลข(Digital Signal Processing) ", สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,กรุงเทพฯ,2533

2.Borland International,Inc.,"Turbo C++,Library Reference",Borland International., USA., 1990.

3. D.O. Shaughnessy.," peech Communication ", Addison Wesley Publishing Company, pp.204-241,1987.

4. James A. Freeman,David M. Skapura, " Neural Networks Algorithms, Applications, and Programming Techniques",Addison Wesley,1991.

5.Lawrence R. Rabiner and Ronald W.Schafer, " Digital Processing of Speech Signals", Prentice-Hall International,Inc., Englewood Cliffs,NJ.,1987

6.O.E.Brigham.,"The Fast Fourier Transform and its applications", Prentice-Hall international. Inc., 1988

7. Paul M. Embree and Bruce Kimble,"C Language Algorithms for Digital Signal Processing", Prentice-Hall International, Inc., Englewood Cliffs, N.J.,1991