



การรู้จำเสียงพูด

SPEECH RECOGNITION

โดย

นาย ยงยุทธ เทพเวียน

นาย ทรวง คุปติวิทยากุล



ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขา วิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2537

๐๕5036

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2537

ปริญญาานิพนธ์ ปีการศึกษา 2537

ภาควิชา อิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

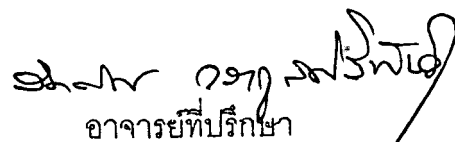
เรื่อง การรู้จำเสียงพูด (SPEECH RECOGNITION)

ผู้จัดทำ

นาย ทรวง คุปติวิทยากุล 35103182

นาย ยงยุทธ์ เทพเวียน 35103198




อาจารย์ที่ปรึกษา

(ผศ.ดร.รัตติกอ วรากุลศิริพันธุ์)

ADVISOR

ASSISTANT PROFESSOR DR.RUTTIKORN VARAKULSIRIPUNTH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อ

Abstract

บทนำ

บทที่ 1	การได้ยิน	1
บทที่ 2	อวัยวะที่ใช้ในการเปล่งเสียง	4
บทที่ 3	การจดจำ	7
บทที่ 4	การวิเคราะห์เสียงพูด	12
บทที่ 5	การแปลงฟูรีเยอร์	19
บทที่ 6	การแปลงฟาสต์ฟูรีเยอร์ และการประยุกต์ใช้งาน	23
บทที่ 7	แนวทางและการออกแบบฮาร์ดแวร์	49

หนังสืออ้างอิง

กิตติกรรมประกาศ



การจดจำเสียงพูด

ผู้จัดทำ

1. นาย ทรวร คุปติวิทยากุล 35103182
2. นาย ยงยุทธ เทพเรียน 35103198

อาจารย์ที่ปรึกษา

ผศ.ดร. รัตติกร วรากุลศิริพันธุ์

ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ จะกล่าวถึง ส่วนของฮาร์ดแวร์ที่ใช้กับระบบจดจำเสียงพูด ทฤษฎีการสุ่ม การเก็บข้อมูลเสียงพูดลงบนเครื่องไมโครคอมพิวเตอร์ เพื่อเก็บลงบนหน่วย ความจำ และทฤษฎีของการจดจำเสียงพูดโดยย่อ เสียงพูดจะถูกบันทึกผ่านเข้ามาทาง ไมโครโฟน และแปลงไปเป็นสัญญาณดิจิทัลโดยใช้ส่วนฮาร์ดแวร์ จากนั้นนำข้อมูลที่ แปลงได้ นำไปคำนวณและประมวลผล โดยส่วนซอฟต์แวร์ ซึ่งการคำนวณจะใช้กระบวนการ วิเคราะห์โดยใช้เทคนิคการแปลงฟูริเยร์ (FFT) เพื่อเปลี่ยนสัญญาณเสียง อินพุต ในโดเมนเวลา ให้เป็นสเปกตรัมในโดเมนความถี่ และหาคุณลักษณะของเสียงพูด โดย การตรวจจับความถี่จุดศูนย์กลางที่มีขนาดสูงสุดในโดเมนความถี่ การคำนวณหาค่าพลังงานของ เสียง แล้วนำมาเปรียบเทียบกับข้อมูลค้นแบบเพื่อแสดงผลลัพธ์ออกมาทางจอคอมพิวเตอร์

SPEECH RECOGNITION

Mr.Suang Cupativitayakul 35103182

Mr.Yongyut Thaprean 35103198

Advisor

Asst.Prof. DR.RuttikomVarakulsiripunth

Academic Year 1994

Abstract

This Thesis describes about hardware part of Speech recognition , the sampling theory , the storage of data which in store in memory and the theory of the speech recognition system. Speech signal will be detected from microphone and converted to digital signal by hardware part. Then data will be taken by software part to calculate and process with mathematic algorithm using " Fast Fourier Transform " technique to change the input speech signal in time domain to spectrum in frequency domain and find the feature of speech signal detecting the maximum peak of all frequency sampled point , " The Domain Analysis " . And then matching pattern algorithm will be used to find an output of recognition.

บทนำ (Introduction)

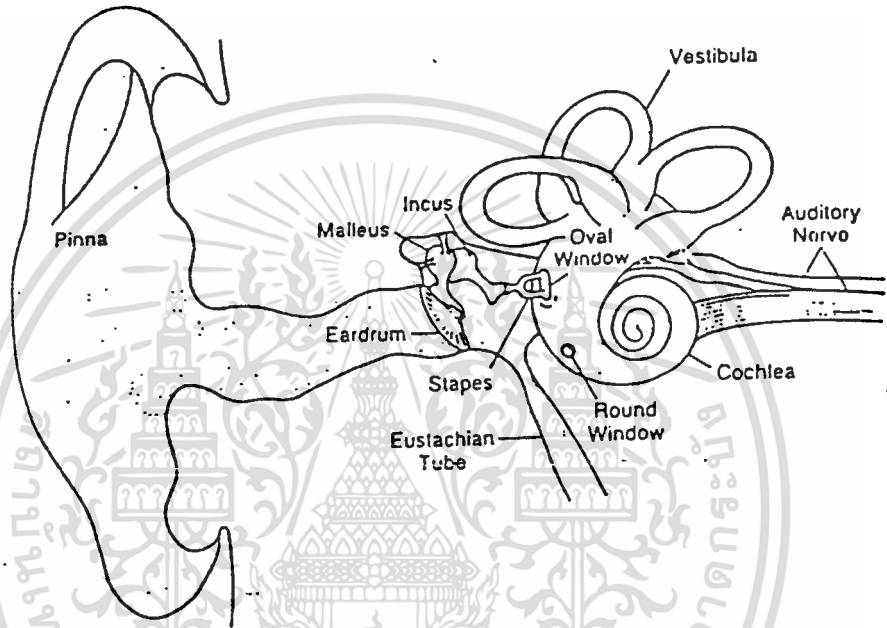
วิธีการใช้คอมพิวเตอร์นั้นได้มีพัฒนาการขึ้นมาอย่างต่อเนื่อง จากการติดต่อที่ผู้ใช้จะต้องพิมพ์คำสั่งลงไปที่เรียกว่าเป็นการติดต่อแบบ Text Based เปลี่ยนมาเป็นการใช้สัญลักษณ์ที่เป็นภาพกราฟิก ที่เรียกว่าเป็นการติดต่อที่เป็น Graphics User Interface การพัฒนาสิ่งเหล่านี้ก็เพื่อที่จะแก้ปัญหา หรือลดความยุ่งยากในการใช้เครื่องคอมพิวเตอร์ นอกจากความพยายามในการแก้ปัญหาในแง่การออกแบบซอฟต์แวร์ และการออกแบบฮาร์ดแวร์ เช่นการนำอุปกรณ์อินพุทที่เป็น Pointing device เช่น เมาส์ จดระบบสัมผัสหรือการพัฒนาคอมพิวเตอร์ระบบปากกา เพื่อทำให้สามารถใช้เครื่องคอมพิวเตอร์ได้อย่างสะดวกมีประสิทธิภาพ ในปัจจุบันการพัฒนาเทคโนโลยีด้านมัลติมีเดียเป็นไปอย่างต่อเนื่อง ทั้งด้านฮาร์ดแวร์และซอฟต์แวร์ สามารถบันทึกเสียง ภาพ วิดีโอเพื่อใช้งานร่วมกันในคอมพิวเตอร์ได้ สิ่งหนึ่งที่นำเสนอติดตามคือการพัฒนาวิธีการควบคุมหรือสั่งงานคอมพิวเตอร์ด้วยเสียง

ด้วยความคิดค้นที่จะควบคุมการทำงานของเครื่องคอมพิวเตอร์ด้วยเสียงพูด ทำให้มีการวิจัยและพัฒนาเทคโนโลยีเกี่ยวกับเสียงมาเป็นระยะเวลาหลายปี แต่ส่วนใหญ่จะเน้นการใช้งานกับเครื่องใหญ่ เพื่อควบคุมระบบโทรคมนาคม หรือเป็นการพัฒนาเพื่อการรักษาความปลอดภัยทางการทหาร สำหรับเครื่องไมโครคอมพิวเตอร์เมื่อมีการผลิต Sound Card ขึ้นมาใช้เพื่อเป็นอุปกรณ์บันทึกเสียงเข้ากับระบบแอมัลติมีเดีย นักวิจัยทั้งหลายก็เริ่มมองเห็นความฝันที่เป็นจริงนั้นมาในการที่จะประยุกต์เอาข้อมูลเสียง มาควบคุมการทำงานของเครื่องคอมพิวเตอร์ นับว่าเป็นงานวิจัยที่ได้รับความสนใจมาก นอกเหนือจากงานวิจัยที่จะทำให้เครื่องคอมพิวเตอร์อ่านหนังสือที่เรียกว่า Text to Speech คือการแปลงข้อมูลที่เป็นความมาให้ออกมาเป็นเสียงพูด ดังที่จะเห็นได้จากผลิตภัณฑ์ที่แถมมากับ Sound Card บางตัวเช่นโปรแกรม Dr.Sbiatso ที่แถมมากับการ์ด Sound Brasterของบริษัท Creative Technology เป็นต้น โปรแกรมประเภทนี้เป็นการวิเคราะห์เสียงพูดแล้วแปลงเป็นคำสั่งให้เกิดการประมวลผล หรือทำงานโดยมีการสอนให้คอมพิวเตอร์จดจำเสียงพูดได้ จึงเรียกว่าเป็นโปรแกรมจดจำเสียงพูด (Speech Recognition)

บทที่ 1

การได้ยิน (Hearing)

ระบบการฟังหมายถึงสิ่งที่ได้รับมาถูกเปลี่ยนไปเป็นสัญญาณกระตุ้นกระตุ้นไปเป็นระบบประสาท เริ่มจากโครงสร้างของหูซึ่งแบ่งออกเป็น 3 ส่วน ได้แก่ หูชั้นนอก หูชั้นกลาง และหูชั้นใน



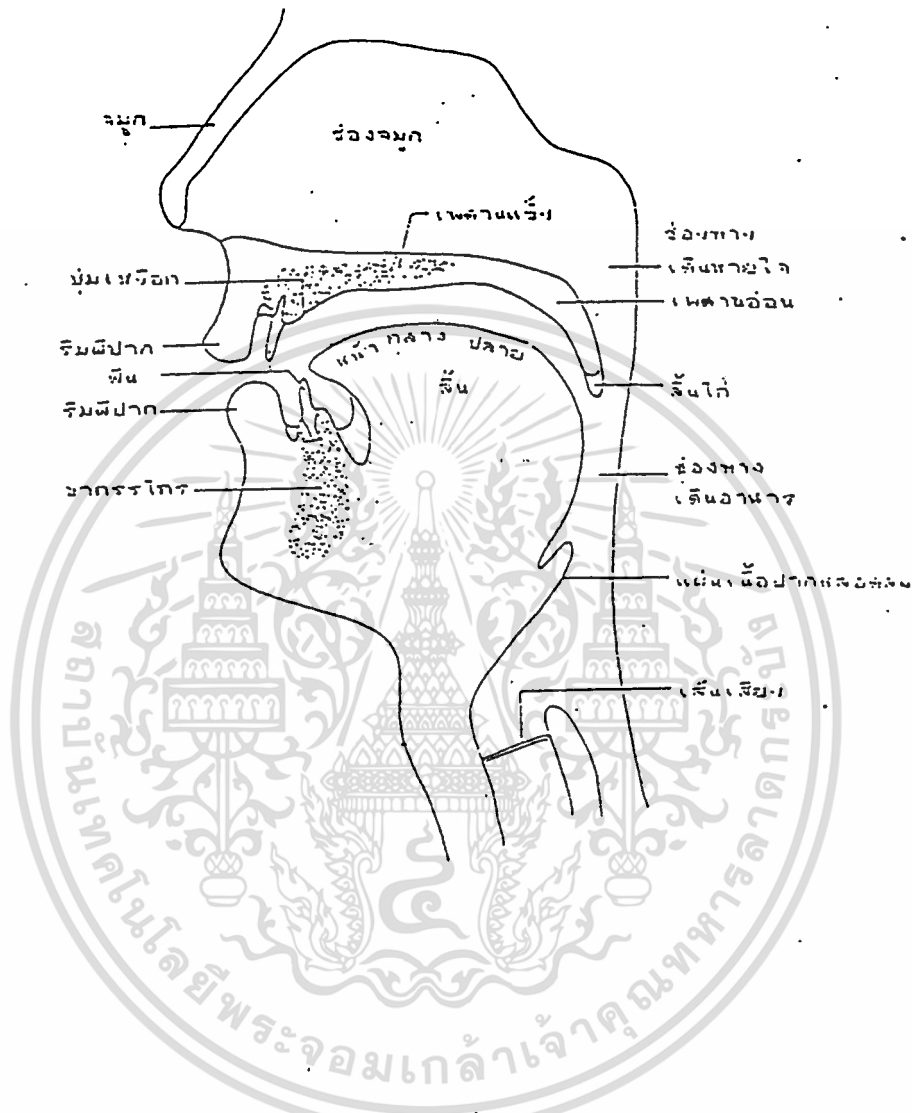
รูปที่ 1.1 แสดงส่วนประกอบของหู

อวัยวะที่ใช้ในการออกเสียง (Organs of Speech)

อวัยวะที่ใช้ในการออกเสียงพูดมีอยู่หลายส่วน แต่ละส่วนสามารถทำให้เสียงพูดแตกต่างกันออกไปได้ อวัยวะเหล่านี้มีปาก และส่วนต่างๆในปาก ช่องคอ กล่องเสียง ช่องปาก และช่องจมูกดังรายละเอียดในรูปที่ 1.2 อวัยวะที่ใช้ในการออกเสียงมี 2 จำพวกคือ

1. อวัยวะที่ใช้ในการกระทำอาหาร (Articulator) หมายถึง อวัยวะส่วนที่เคลื่อนไหวเพื่อปลักลมไปยังส่วนต่างๆ อวัยวะตัวปลักลมอาหารที่สำคัญ คือ ลิ้น ซึ่งเป็นส่วนที่เคลื่อนไหวได้มากที่สุด

2. อวัยวะที่เป็นตำแหน่งที่เกิดเสียงต่างๆ (Point of Articulator) หมายถึง ตำแหน่งหรือ ฐานกรณ์ที่เกิดเสียงต่างๆ เช่น เพดาน ส่วนต่างๆ ริมฝีปาก ฟัน เป็นต้น



รูปที่ 1.2 แสดงอวัยวะที่ใช้ในการออกเสียงพูด

การเกิดของเสียง (Speech Production)

ขั้นตอนการเกิดของเสียงแบ่งได้เป็น 3 ขั้นตอนใหญ่คือ

1. ขั้นเริ่มต้น (Initiation) ขั้นตอนนี้เริ่มจากลมเริ่มถูกขับออกจากปอด เพื่อเข้าสู่ขั้นตอนที่สองต่อไป

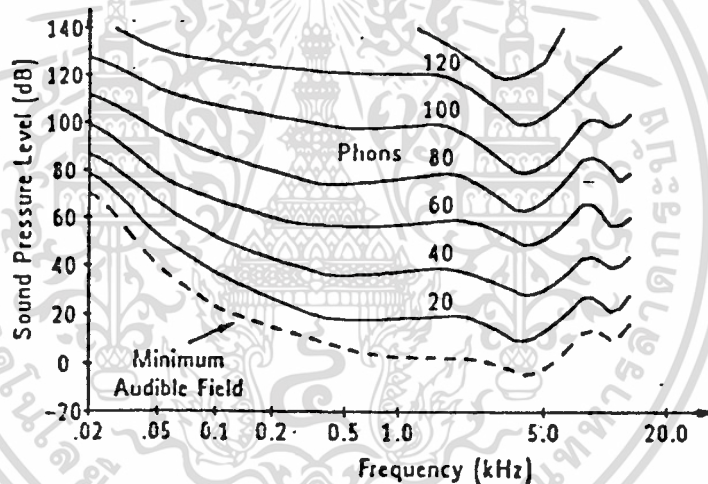
2. ขั้นตอนดัดแปลงลมที่เส้นเสียง (Phonation) เป็นขั้นตอนที่ลมจากปอดจะผ่านมายังหลอดลม และกล่องเสียง ซึ่งที่กล่องเสียงนี้ เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดปิด ทำให้เกิดเสียงได้ 2 ชนิดคือ เสียงก้อง (Voiced Sounds) และเสียงไม่ก้อง (Unvoiced - Sounds) อวัยวะที่ใช้ในตอนนี้ คือ ส่วนที่ต่อจากปอดขึ้นมาที่กล่องเสียง

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ขั้นตอนเปลี่ยนแปลงลักษณะเสียง (Articulation) ขั้นตอนนี้ลมที่ผ่านออกมาจากกล่องเสียงจะถูกแปลงให้เกิดเสียงในลักษณะต่างๆวิธีที่ใช้ในขั้นตอนนี้ คือ ส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก

ความเข้าใจ (Perception)

หน้าที่ของการรับเสียงและความสามารถ โดยปรกติจะอยู่ในช่วง 16 Hz ถึง 16 KHz ในเด็กอาจถึง 20 KHz ยังมีอยู่ การรับฟังจะลดสมรรถภาพลงตามอายุ อาจได้ถึง 10 KHz การได้ยินความถี่ในช่วงย่านต่ำสัญญาณที่ได้อินจะเป็นแบบขบวนสัญญาณ (Pulse train) ที่ความถี่ด้านสูงความถี่จะค่อยจางลงจนไม่ได้ยิน ช่วงของความเข้มจะอยู่ในช่วงที่ 1-130 dB ที่ความเข้มต่ำเสียงจะเบาจนไม่ได้ยิน ส่วนที่ความเข้มสูงอาจทำให้แก้วหูจบปวดได้



แสดงเส้นแสดงระดับความเท่ากันของระดับความดัง

ความดัง การรับรู้ของความดัง เป็นทั้งหน้าที่ของความถี่และระดับของเสียง โดยการเปรียบเทียบโทน (Tone) ของเสียงที่ต่างกันของความถี่และแอมพลิจูด

การเข้าใจเสียงพูด ปัญหาพื้นฐานที่กำหนดโดยเสียงพูดคือ สัญญาณเสียงที่เข้ามาที่หูจะแปลงโดยสมองเป็นเสียงพูดได้อย่างไร จากการทดลองหลายๆครั้ง แล้วทำการทดลองก็ได้รับคำตอบบางส่วน อย่างไรก็ตามคำตอบที่สมบูรณ์ก็ยังไม่ปรากฏอย่างเด่นชัด เราสามารถสรุปความสอดคล้องของบางส่วนของข้อมูลที่หาข้อมูลมาได้ ดังนี้

ปรากฏว่า สมองจะสร้างความแตกต่างระหว่างเสียงพูดกับสัญญาณที่ไม่ใช่เสียงพูด การทดลองพบว่าเมื่อผู้ฟังได้ยินเสียงที่สังเคราะห์ขึ้น ไม่ใช่เสียงพูด แต่ทำคล้ายเสียงพูด จะไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พบขอบเขตที่มีการเปลี่ยนแปลงกระทันหันข้างใดข้างหนึ่ง จะถูกเข้าใจว่าเป็นเสียงพูด ส่วนอีกข้างหนึ่งจะไม่ใช่เสียงพูด

สมองสามารถแยกเสียงที่ไม่ใช่เสียงพูดและไม่ใช่เสียงพูดได้ เชื่อว่าสมองส่วนที่ประมวลผลเสียงพูดจะอยู่ที่สมองซีกซ้าย ที่เป็นเช่นนี้ เพราะหูข้างขวาจะแสดงสมรรถนะที่ดีกว่าหูทางด้านซ้ายในการรับฟัง มีความเชื่อในทางที่ว่า ระบบประสาทของมนุษย์ ใช้ในการถอดรหัสเสียงที่เข้ามา สมองจะทำการแยกจำพวกของเสียงพูด นอกจากจะแบ่งเป็นสัญญาณเสียงพูดและไม่ใช่เสียงพูดแล้วจะแยกความถี่มูลฐานของฮาร์โมนิคและแยกออกมาเป็นเสียงพูดหลายชนิดหน่วยประมวลผลกลางจะแสดงการจดจำโดยใช้ผลอันนี้ในการรับเข้ามา

การวิจัยชี้ให้เห็นว่า การเข้าใจเสียงพูดขึ้นอยู่กับคุณลักษณะความชัดเจนในการออกเสียงและการวิเคราะห์เสียงพูด



บทที่ 2

อวัยวะที่ใช้ในการเปล่งเสียง (Articulation)

อวัยวะที่ใช้ในการเปล่งเสียง แบ่งได้เป็น 3 พวกใหญ่ๆ คือ

1. อวัยวะที่ใช้ในการสร้างลม คือ ส่วนที่ทำให้เกิดการเคลื่อนไหวของลม
2. อวัยวะที่เคลื่อนที่ได้ (Active Articulation) หมายถึง อวัยวะที่ติดกับกระดูกของคางส่วนล่าง ได้แก่ ริมฝีปาก และลิ้น
3. อวัยวะที่เคลื่อนที่ไม่ได้ (Passive Articulation) หมายถึง อวัยวะที่ติดกับกระดูกคางส่วนบน ได้แก่ ริมฝีปากบน ฟันบน ปุ่มเหงือกเพดานแข็ง เพดานอ่อน ลิ้นและอวัยวะที่เป็นช่องว่าง ได้แก่ ช่องคอ ปาก ช่องจมูก

ถ้าจะลำดับการเกิดเสียงแล้วจะแบ่งออกเป็น 3 ขั้นตอนใหญ่ๆ

1. จุดเริ่มต้น (Initiation) อวัยวะที่ใช้ในขั้นตอนนี้คือ ปอด เป็นขั้นตอนที่ใช้ ท่อลมเริ่มขับออกจากปอด

2. การดัดแปลงลมที่เส้นเสียง (Phonation) อวัยวะที่ใช้ในขั้นตอนนี้คือ ส่วนที่ต่อจากปอดขึ้นมา จนถึงกล่องเสียง เป็นขั้นตอนที่ลมออกจากปอด จะผ่านมายังหลอดลม และกล่องเสียง ซึ่ง ณ ที่กล่องเสียงนี้ เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิดปิด ทำให้เกิดเสียงได้ 2 ชนิดคือถ้าเส้นเสียงเปิดตลอดเวลาที่ลมผ่าน ลมจะผ่านออกมาได้โดยสะดวก ซึ่งจะทำให้เกิดเสียงชนิดไม่ก้อง แต่ถ้าเส้นเสียงปิดกั้นลมไว้ลมที่ผ่านออกมาจะเพิ่มแรงดันมากขึ้นจนเส้นเสียงเปิดและปิดสลับกันไป ทำให้เกิดเสียงชนิดก้องและ เรียกความถี่ในการปิดเปิดของเส้นเสียงว่า ความถี่พื้นฐาน (Fundamental frequency)

ขั้นตอนที่ 3 การเปลี่ยนแปลงลักษณะเสียง (Articulation) อวัยวะที่ใช้ในขั้นตอนนี้คือ ส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก และช่องว่างใหญ่อีก 2 ส่วน คือ ช่องปากและช่องจมูก ขั้นตอนนี้ลมที่ผ่านออกมาจากกล่องเสียงจะถูกแปลงให้เกิดเสียงลักษณะต่างๆ ทั้งนี้เพราะว่าช่องว่างภายในปาก จะมีการเปลี่ยนแปลงขนาดตลอดเวลาที่เปลี่ยนแปลงเสียง และช่องจมูกก็จะถูกควบคุมโดยเพดานอ่อน (Velum) ดังนั้นถ้าช่องว่างต่างๆเหล่านี้เปลี่ยนแปลงขนาดไป ก็จะทำให้เกิดความถี่กำทอน (Resonant Frequency) ของเสียงเปลี่ยนแปลงไปด้วย ซึ่งช่องว่างเหล่านี้เปรียบเสมือนกับท่อที่มีขนาดต่างๆกัน

หน่วยการจำเสียงพูด (Speech recognition Unit)

มีการจำแนกวิธีการตามลักษณะของการพูดไว้ 4 แบบคือ

1) ผู้พูดคนเดียว หรือผู้พูดคนเดียวกัน (Speaker dependent recognition - system) คือการพูดเป็นหน่วยเสียง หน่วยพยางค์ หน่วยคำหรือวลี สิ่งที่ต้องทำคือ เมื่อฟังเสียงจากผู้พูดแล้ว จะบันทึกรูปแบบของเสียงตามวิธีของเครื่องเอง โดยอาศัยขั้นตอนนี้

เมื่อผู้ใช้ต้องการสอนให้เครื่องรู้จักคำโดยพูดผ่านไมโครโฟน สัญญาณเสียงซึ่งอยู่ในรูปของสัญญาณไฟฟ้า (analog) จะถูกเปลี่ยนให้เป็นสัญญาณดิจิทัลก่อน จากนั้นก็จะนำข้อมูลดิจิทัลไปคำนวณ ซึ่งจะบอกถึงลักษณะเฉพาะของคำนั้นๆ และเก็บข้อมูลต้นแบบนี้ไว้ในหน่วยความจำ หรือในแผ่นแม่เหล็กสำหรับการสอนเครื่องให้รู้จักคำนั้นจะมีความเชื่อถือได้มากน้อยแค่ไหนขึ้นอยู่กับ algorithm ที่ใช้ในการวิเคราะห์

2) ผู้พูดเป็นใครก็ได้ (Speaker-independent recognition system) และวิธีการพูดยังเป็นแบบคำแยกจากกัน (discrete utterance) วิธีนี้ในแบบนี้จะสามารถจำคำได้น้อยกว่าในแบบวิธีที่หนึ่งมาก ต้นแบบจะถูกโปรแกรมไว้ล่วงหน้าแล้วใช้ algorithm จัดเข้ากลุ่มเพื่อที่จะฟังเสียงคนใดก็ได้

3) สอนให้เครื่องรู้จักคำในลักษณะของหน่วยเสียงหรืออย่างมากหน่วยพยางค์ (Unconnected speech) เป็นเทคนิคที่เริ่มใช้มาก่อนใช้กับระบบที่ไม่แพง ต้นแบบจะถูกเก็บไว้เป็นหน่วยเสียงหรือหน่วยพยางค์เท่านั้น

4) ผู้พูดสามารถพูดแบบต่อเนื่องเหมือนการพูดปกติ (Connected speech) เครื่องจะต้องสามารถรู้จักและจำแนกคำที่มีการออกเสียงเชื่อมต่อกัน การพูดแบบนี้สำหรับมนุษย์เป็นเรื่องง่าย เพราะไม่ต้องตระเตรียมเป็นพิเศษ แต่สำหรับเครื่องเป็นสิ่งที่ยากมาก เพราะการพูดแบบนี้ไม่สามารถหาเส้นแบ่งเขตระหว่างคำหรือระหว่างพยางค์ได้อย่างแม่นยำ ดังนั้นการรู้จักคำโดยเอาเปรียบเทียบกับต้นแบบจึงไม่ใช่ของง่าย วิธีการนี้ใช้ได้กับคอมพิวเตอร์ระดับมินิขึ้นไปเท่านั้น

เสียงพูดเป็นเสียงธรรมชาติอย่างหนึ่งของมนุษย์ที่ใช้ในการสื่อสารซึ่งกันและกันทำให้มีการถ่ายทอดความรู้และข่าวสาร และเนื่องจากมนุษย์พยายามพัฒนาคอมพิวเตอร์ให้ใช้กับชีวิตประจำวันได้มากยิ่งขึ้นดังนั้นมนุษย์สามารถติดต่อกับคอมพิวเตอร์โดยการพูดได้จะเป็นประโยชน์อย่างมาก

ในปัจจุบันอาจกล่าวได้ว่าเป็นยุค " โลกาภิวัตน์ (Globalization) " การติดต่อระหว่างมนุษย์และคอมพิวเตอร์โดยใช้เสียงพูดเป็นเทคโนโลยีที่ได้รับความสนใจและพัฒนาในหลายประเทศ

การสร้างระบบจดจำเสียงพูดให้คอมพิวเตอร์เป็นหนึ่งในหลายๆสิ่งที่มีมนุษย์สร้างขึ้นสำหรับมนุษย์นั้นได้เปรียบที่ที่สามารถแยกสัญญาณที่ความถี่ต่างๆกันออกได้ดีทำให้ไม่เกิดการสับสนในการฟังแต่กระบวนการวิเคราะห์เสียงพูดนั้นจะต้องแยกความถี่ในช่วงที่เราสนใจออกมา เพราะการเปลี่ยนแปลงสัญญาณอาจเกิดสัญญาณรบกวนความถี่สูง (High Frequency Noise) หรือสัญญาณนิบเปิล (Nibble) ซึ่งเป็นสัญญาณความถี่ต่ำๆได้

ขั้นตอนต่างๆของการวิเคราะห์ ตลอดจนวิธีการค้นหาคำรู้จักโดยนำไปเปรียบเทียบกับต้นแบบมี 5 วิธีการใหญ่ๆดังนี้

เอกสารนี้เป็นเอกสารที่ Digital Waveform Coding การศึกษาเท่านั้น ไม่นุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 2 Time Domain analysis Methodes
- 3 Shot-Time Spectrum Analysis
- 4 Homomophic Predictive Analysis
- 5 Linear Predictive Analysis



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การจดจำ (recognition)

3.1 ประเภทของการจดจำ

ในการประมวลผลสัญญาณเสียงพูด มีการประยุกต์ใช้ในงานสำคัญ 2 ด้านคือ

1. การจดจำเสียงพูด (Speech recognition)
2. การจดจำผู้พูด (Speaker recognition)

ทั้งสองหัวข้อนี้เป็นการจดจำรูปแบบอย่างหนึ่ง การจดจำรูปแบบได้ถูกนำไปใช้ในงานหลายๆด้าน เช่น การประมวลผลสัญญาณภาพ แต่ในที่นี้จะอธิบายถึงการจดจำรูปแบบ ที่เกี่ยวข้องกับการประมวลผลสัญญาณเสียงพูดเท่านั้น

3.2 การจดจำรูปแบบ (Pattern recognition)

การจดจำรูปแบบเป็นการเปรียบเทียบรูปแบบตรวจสอบ (Test pattern) ซึ่งเป็นตัวแทนของสิ่งที่ไม่รู้เพื่อแยกแยะหรือจดจำ กับรูปแบบอ้างอิง (Reference pattern) หนึ่งตัวหรือหลายตัว ซึ่งได้รู้ลักษณะมาก่อนแล้ว

ในแต่ละรูปแบบจะอยู่ในรูปแบบของเวกเตอร์ แต่ละสมาชิกของเวกเตอร์เป็นสมาชิกที่วัดได้ของลักษณะเด่น (Feature) ลักษณะเด่นเป็นลักษณะที่วัดได้ของสัญลักษณ์ของสัญญาณอินพุตและสามารถนำไปใช้ประโยชน์ในการจดจำรูปแบบได้ เช่น ในระบบการจดจำเสียงพูดแบบแยกคำ รูปแบบต่างๆจะเก็บในรูปแบบของฟังก์ชันเวลา เพราะค่าแต่ละลักษณะนั้นไม่ได้วัดที่จุดใดจุดหนึ่งของคำพูด แต่มาจากการวัดทั้งคำ รูปแบบเหล่านี้เราเรียกว่า ต้นแบบ (Template)

ในระบบการจดจำจะมีการทำงานคล้ายตัวอย่างดังต่อไปนี้

- สมมุติว่า ระบบการจดจำผู้พูดเป็นระบบที่สร้างขึ้นสำหรับผู้พูด 50 คน เมื่อมีการพูดคำหรือวลีระบบจะทำการประมวลคำหรือสัญญาณเสียงพูดและดึงเอาลักษณะเด่นออกมา (โดยแยกคำและวัดค่า) เพื่อทำเป็นรูปแบบอ้างอิง ดังนั้นระบบนี้จึงมีไลบรารีรูปแบบอ้างอิงอยู่ถึง 50 รูปแบบ เมื่อทำการเปรียบเทียบรูปแบบตรวจสอบจะถูกเปรียบเทียบกับรูปแบบอ้างอิงในไลบรารีทั้งหมด เพื่อหารูปแบบอ้างอิงที่ใกล้เคียงกับรูปแบบมากที่สุด เพื่อแยกแยะว่าเป็นผู้พูดคนใด

ระบบการจดจำอัตโนมัติ นอกจากใช้ในการจดจำคำพูดและผู้พูดแล้ว ยังอาจนำไปใช้ในงานด้านอื่นอีก โดยแยกสิ่งที่ไม่รู้ออกเป็นคลาส (Class) เช่น ในระบบจดจำตัวเลข คลาสคือคำว่า { ศูนย์, หนึ่ง, สอง, สาม, . . . เก้า } ระบบจะมีไลบรารีในรูปแบบอ้างอิงในแต่ละคลาส

เอกสารนี้เป็นขบวนการจดจำจะทำการเปรียบเทียบเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทียบความคล้ายคลึงของสัญญาณไม่รู้จักกับคลาสแต่ละคลาส เพื่อหารูปแบบที่ใกล้เคียงที่สุด การทำงานของการจดจำโดยทั่วไปมีอยู่สองขั้นตอน ดังนี้

1. การเรียนรู้ (หรืออาจจะเรียกว่าการฝึกก็ได้) เป็นการรวบรวมไลบรารีของรูปแบบอ้างอิง
2. การจดจำ เป็นการใช้ไลบรารีที่สมบูรณ์แบบเปรียบเทียบกับอินพุท ที่ไม่รู้

ในการเรียนรู้ ข้อมูลของแต่ละคลาสจะถูกป้อนเข้าสู่ระบบ ระบบจะทำการสร้างรูปแบบอ้างอิงหรือต้นแบบของแต่ละคลาส โดยการเฉลี่ยรูปแบบจำนวนมากแล้วนำมารวมกันในไลบรารีโดยแยกเป็นคลาสต่างๆ ในการจดจำ ระบบจะทำการคำนวณรูปแบบของลักษณะเด่นของอินพุทที่ไม่รู้ และทำการตรวจสอบกับคลาสต่างๆ เพื่อหาคลาสที่มีรูปแบบอ้างอิงใกล้เคียงกับลักษณะเด่นที่ได้มากที่สุด

การออกแบบระบบการจดจำ มีปัญหาที่สำคัญอยู่ 2 ประการคือ

1. การเลือกลักษณะเด่นและการคำนวณค่า
2. ทางเลือกของกฎการตัดสินใจ (Decision rule)

ปัญหาแรกที่มีความสำคัญมากและยากมากกว่า เราจึงควรเริ่มต้นที่ปัญหาที่สองก่อน เพราะมีแนวความคิดหลายๆอย่าง ซึ่งจะช่วยอธิบายในหัวข้อลักษณะเด่น

3.3 การจดจำเสียงพูด (Speech recognition)

3.3.1 ประเภทของการจดจำเสียงพูด

การจดจำเสียงพูดโดยทั่วไปแบ่งออกเป็น 4 กลุ่ม โดยทั่วไปเรียงลำดับจากความยากดังต่อไปนี้

1. การจำแบบแยกคำ (Isolated-word recognition) เป็นการจำโดยต้องมีช่วงหยุดของแต่ละคำไป
2. การจำเฉพาะคำ (Word spotting) ตรวจคำเฉพาะใดๆที่ปรากฏในประโยคนั้น
3. การจำแบบต่อเนื่อง (Connected speech recognition) จดจำคำพูดได้ทั้งประโยค โดยไม่ต้องมีช่วงหยุดระหว่างคำ
4. การเข้าใจเสียงพูด (Speech understanding) เป็นแบบสมบูรณ์ของกลุ่มที่

3 โดยสร้างแหล่งข้อมูลของการออกเสียงและหลักของภาษานั้น จุดประสงค์หลักก็คือ การพูดที่ไม่ต้องมากนักแต่ให้เข้าใจความหมายของประโยคนั้นได้ ในทางทฤษฎี ระบบเข้าใจเสียงพูด จะตัดทอนเอาเฉพาะความหมายของประโยค โดยไม่เข้มงวดเรื่องไวยากรณ์



3.3.2 การจำแบบแยกคำ (Isolated-word recognition)

การจำแบบแยกคำ เป็นจุดเริ่มต้นของการพัฒนาจดจำเสียงพูด จนเป็นการจำเฉพาะคำและการจำแบบต่อเนื่องตามลำดับ ช่วงหยุดของแต่ละคำพูดทำให้การจดจำทำได้ง่ายขึ้น เพราะสามารถแยกแยะจุดสิ้นสุดของคำ (หมายรวมถึงจุดเริ่มต้นและจุดสิ้นสุดของคำนั้น) ทำให้ผลที่เกิดจากความไม่ชัดเจนของการพูดลดน้อยลงไปมาก การจำแบบแยกคำขณะที่แยกเสียงจะต้องระมัดระวังอย่างมาก เพราะต้องการช่วงหยุดของแต่ละคำ ทำให้การพูดขาดความไพเราะของภาษาไป ไม่เหมือนวิธีการอื่นๆ ซึ่งสามารถพูดเป็นธรรมชาติได้มากกว่าและใช้ความระมัดระวังในการพูดมากกว่า

การจำแบบแยกคำ มีหลักการการจดจำโดยใช้ลักษณะของสัญญาณเสียงพูด ดังต่อไปนี้

1. ขนาดหรือกำลังต่อเวลา
 2. อัตราตัดผ่านศูนย์ (zero-crossing rate)
 3. สมดุลย์กลุ่มความถี่รวม (Gross spectrum balance) เป็นการเปรียบเทียบพลังงานของกลุ่มความถี่สูงกับกลุ่มพลังงานความถี่ต่ำ
 4. ความถี่สำคัญ อาจมาในรูปของ
 - ก. ความถี่จากการแปลง DFT
 - ข. F1, F2, F3
 - ค. พารามิเตอร์ LPC หรือสัมประสิทธิ์ PARCOR
 - ง. เอาร์พุทของฟิลเตอร์แบงก์ (Filter - bank) วิธีนี้เป็นวิธีที่นิยมกันมากที่สุดเพราะราคาที่ถูกและความเร็วสูง
- ขนาดนอกจากเป็นตัวบอกจุดสิ้นสุดของแต่ละคำ และยังเป็นตัวชี้ความแตกต่างของพยัญชนะและสระอีกด้วย

3.3.3 การตรวจจับจุดสิ้นสุด (Endpoint detection)

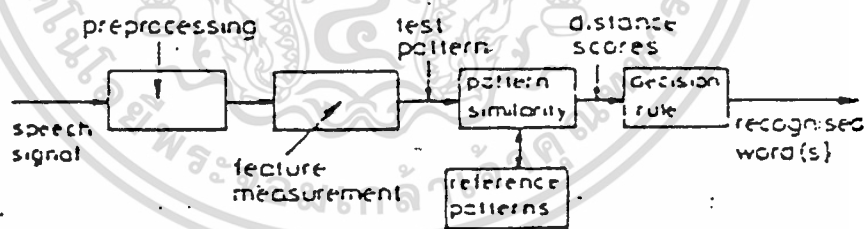
ปัญหาพื้นฐานของระบบจดจำเสียงพูด คือ เสียงที่เปล่งออกมาในขณะที่ตรวจสอบนั้น อาจจะไม่เหมือนกันตอนที่เรารู้เสมอไป เสียงที่เปล่งออกมา 2 ครั้ง จะมีระยะเวลาที่ต่างกัน และระยะที่ออกเสียงกันไม่แน่นอน หมายความว่าคุณสมบัติที่ขึ้นกับระยะเวลาที่ต่างกันเวลาจะทำให้การเปรียบเทียบไม่สามารถทำได้เพราะค่าที่อ้างอิงกับเวลาที่นำมาเปรียบเทียบต่างกัน ในกรณีเช่นนี้ ถึงแม้ว่าจะเปรียบเทียบกับค่าที่ไม่รู้จักกับรูปแบบที่ถูกต้องของตัวเองแล้วก็ตาม แต่ก็อาจจะให้ผลที่แตกต่างกันได้พอๆกับที่เปรียบเทียบกับรูปแบบอ้างอิงของคำอื่น ซึ่งจะ เป็นปัญหา ถ้าค่าที่ถูกเปรียบเทียบเป็นการเปรียบเทียบทั้งคำต้นแบบแทนที่จะเปรียบเทียบ บางส่วนของการเปรียบเทียบส่วนต่อส่วน ความจริงแม้ในระหว่งเสียงสองเสียงที่ใช้ในการ

เรียนรู้ รายละเอียดของเวลาที่ต่างกันเลย เมื่อเก็บข้อมูลของหลายๆคน จึงต้องทำการเฉลี่ยในระหว่างการเรียนรู้ก่อนนำไปเปรียบเทียบ ในหลายๆวิธี ความแม่นยำของการปรับจะขึ้นอยู่กับความแม่นยำของการแสดงจุดสิ้นสุดของคำ ความผิดพลาดในการจดจำจุดเริ่มต้นและจุดสิ้นสุดของคำจะทำให้เกิดความยุ่งยากขึ้น โดยความผิดพลาดของจุดสิ้นสุดจะเกิดขึ้นในคำที่เริ่มต้นและลงท้ายด้วยพยางค์ที่ออกเสียงเบา ผู้พูดบางคนอาจจบประโยคด้วยเสียงสั้น พยางค์ที่เสียงสั้นนี้จะทำให้เกิดความผิดพลาดในการเปรียบเทียบค่า

ในห้องทดลอง การจดจำจุดสิ้นสุดจะทำได้ง่าย เพราะข้อมูลเสียงพูดถูกรวบรวมภายในสภาวะเพื่อหาจุดสิ้นสุด ยิ่งกว่านั้น ผู้พูดที่บันทึกในห้องทดลอง จะทำด้วยความระมัดระวัง แต่ในห้องทดลองไม่ได้เป็นอย่างนั้น เสียงพูดจะมีการรบกวนจากรอบข้างเพราะไม่มีการควบคุมสภาวะ และผู้พูดไม่มีแรงจูงใจในการจดจำ

คุณสมบัติหลักของจุดสิ้นสุด คือพลังงาน ดังนั้นวิธีการง่ายๆที่จะใช้ตรวจจับจุดสิ้นสุดก็คือ การเปรียบเทียบพลังงานของสัญญาณกับค่าหรือระดับที่ตั้งไว้ โดยตรวจสอบจุดเริ่มต้นของคำจากตำแหน่งของคำที่มีพลังงานเริ่มมากกว่าระดับที่ตั้งไว้ ค่าระดับที่ตั้งไว้ต้องมีค่าความเข้มระดับหนึ่ง ถ้าความแรงของสัญญาณเสียงพูดอ่อน อาจจะต้องตั้งระดับที่ต่ำๆ

3.4 ระบบการจดจำเสียงพูดแบบแยกคำและไม่ขึ้นกับผู้พูด ด้วยเทคนิค DTW



รูปที่ 3.1 แสดงแบบจำลองของการจดจำรูปแบบในการจดจำเสียงพูด

รูปที่ 3.1 ได้แสดงแบบจำลองของการจดจำรูปแบบของสัญญาณเสียงพูด ซึ่งมีลักษณะการทำงานที่สำคัญ 2 อย่าง คือ การฝึก และการตรวจสอบ การฝึกเป็นการสอนศัพท์ให้กับระบบจดจำเสียงพูด โดยใช้ค่าเฉลี่ยของแต่ละคำ เป็นตัวแทนหรือต้นแบบอ้างอิง (Reference template) ของคำนั้น การเฉลี่ยค่าๆเดียวนั้นหลายครั้ง หลังจากการปรับแนวเวลา (Time alignment) เป็นวิธีการอย่างหนึ่ง เพื่อให้ได้ต้นแบบอ้างอิงตามที่ต้องการ

ในการตรวจสอบหรือการจดจำ เป็นการเปรียบเทียบค่ากับต้นแบบอ้างอิงที่เก็บไว้แล้ว เมื่อแปลงเสียงออกมาคำพูดนั้นจะโดนบันทึกโดยตัวกรองความถี่ต่ำผ่าน(Bandpass -filter) ลดทอนความถี่ต่ำและจำกัดช่วงความถี่ เพื่อป้องกันความผิดเพี้ยนของสัญญาณจากการสุ่มเชิงเลข (Digitize) หลังจากนั้น ค่าที่แปลงออกมาจะถูกแยกสัญญาณที่ไม่มีเสียงพูด (Silence) ออกไป และมีการตรวจสอบจุดสิ้นสุดของแต่ละคำ เพื่อสะดวกในการประมวลผลต่อไป แทนที่จะเก็บคำที่พูดออกมาทั้งหมด พารามิเตอร์เสียงพูด(Speech parameter) ที่เป็นลักษณะเด่นของคำนั้นก็จะแยกออกมา ในขั้นตอนนี้จะต้องลดขนาดของพารามิเตอร์ที่แยกได้ เพื่อทำการเปรียบเทียบกับรูปแบบอ้างอิง (Reference pattern) หรือรูปแบบที่ใกล้เคียงกับรูปแบบของคำที่ตรวจสอบมากที่สุด วิธี NN (nearest neighbour) และ KNN (K nearest neighbour) เป็นกฎการตัดสินใจ (Decision rule) ที่นิยมกันมากที่สุด

โดยทั่วไป ผู้พูดไม่สามารถพูดซ้ำคำเดิมให้เหมือนเดิมทุกอย่างได้ ผลที่ได้คือช่วงของเสียงพูดที่เกิดขึ้นในเวลาใดๆ T_r อาจเกิดขึ้นที่เวลาต่างกับเวลา T_t ของแบบตรวจสอบ (Test pattern) ทำให้เกิดปัญหาในการเปรียบเทียบรูปแบบ วิธีการหนึ่งที่จะเอาชนะรูปแบบนี้ได้ ก็คือการเปลี่ยนอัตราส่วนแกนเวลาของรูปแบบตรวจสอบแบบไม่เชิงเส้น เพื่อให้ตรงกับรูปแบบอ้างอิงให้มากที่สุด วิธีนี้เราเรียกว่า DTW (Dynamic time warping) ซึ่งเป็นวิธีที่นิยมกันมากที่สุด

บทที่ 4

การวิเคราะห์เสียงพูด (Speech Analysis)

ในการเก็บหรือจดจำเสียงพูด เราต้องพยายามลดช่วงของสัญญาณเสียงพูดที่เกิดซ้ำกัน โดยใช้ตัวแทนของเสียงพูดที่สามารถแทนลักษณะสำคัญของเสียงพูดนั้นในเทอมของค่าพารามิเตอร์ (Parametre) เพื่อให้สามารถจัดการกับฐานข้อมูลได้ง่าย

ต่อไปเราจะได้อธิบายถึงวิธีการที่เราจะใช้ในการวิเคราะห์เสียงพูด ทั้งในโดเมนเวลา (กระทำกับเสียงพูดที่เก็บไว้โดยตรง) และโดเมนความถี่ (ผ่านการแปลงเป็นความถี่ก่อน) เพื่อหาตัวแทนของเสียงพูดในรูปของพารามิเตอร์ที่เหมาะสมที่สุด สามารถนำไปใช้งานได้ และมีข่าวสารของข้อมูลครบถ้วน การวิเคราะห์ในโดเมนเวลา เราต้องการการคำนวณเพียงเล็กน้อย ทำให้ถูกจำกัดให้วัดได้เฉพาะลักษณะที่วัดได้ง่ายเท่านั้น เช่นการวัดพลังงานและคาบเวลาของสัญญาณ ในขณะที่การวิเคราะห์ในโดเมนความถี่จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพมากกว่า แต่การสุ่มสัญญาณเสียงพูดจะต้องมีความเที่ยงตรงสูง

เทคนิคที่ใช้ในการวิเคราะห์เสียงพูด อาจทำได้ทั้งแบบดิจิตอลและอนาลอก การประมวลผลอนาลอก เป็นการใช่วงจรทางด้านอิเล็กทรอนิกส์ ซึ่งมีข้อดีอยู่ที่ความเร็วสูง แต่จำเป็นต้องใช่วงจรเฉพาะอย่างทำให้ต้องต่อสายใหม่และปรับค่าใหม่ทุกครั้งที่น่าไปประยุกต์ใช้กับงานอื่น ในขณะที่เทคนิคทางดิจิตอลสร้างและเปลี่ยนแปลงได้ง่ายกว่า เพียงแต่ต้องการซอฟต์แวร์หรือโปรแกรม และฮาร์ดแวร์พิเศษ (ไมโครโปรเซสเซอร์และชิพ) ถึงแม้ว่า อาจจะมีปัญหาเรื่องความเร็ว ไม่สามารถตอบสนองในเวลาจริงได้ แต่ในปัจจุบันเทคโนโลยีทางด้าน VLSI ได้พัฒนาไปอย่างมาก ทำให้ลดข้อเสียของเทคนิคทางดิจิตอลไปได้มาก สำหรับงานในด้านการทดลอง ขอแนะนำให้ใช้ทางด้านดิจิตอลเพราะสามารถแก้ไขได้ง่ายกว่า

4.1 การวิเคราะห์เสียงพูดในช่วงสั้น (Short time speech analysys)

สัญญาณเสียงพูดเป็นสัญญาณที่เปลี่ยนแปลงไปตามเวลาโดยเกิดในลักษณะแบบสุ่ม (Random) แต่ก็ขึ้นกับการควบคุมเสียงของผู้พูดด้วย เพราะเสียงที่เปล่งออกมาในระยะเวลาหนึ่งนั้น จะขึ้นกับรูปร่างของท่อทางเสียง (Vocal tract) และลักษณะการสั่นของเส้นเสียง (Vocal-cord) สัญญาณที่เป็นเสียงพูดจึงเป็นคาบเวลาชั่วขณะ (quasi -periodic) หมายความว่า สัญญาณเสียงพูดมีคาบเวลาคงที่ในเวลาอันสั้น และมีกาเปลี่ยนแปลงในระหว่างเวลานั้น

ถ้าเราพูดซ้ำมาหลายๆ เสียงที่พูดอาจมีการเปลี่ยนแปลงอยู่ในช่วง 200 ms ก็ได้ แต่การพูดโดยทั่วไป ลักษณะการเปลี่ยนแปลงของเสียงพูดจะอยู่ในช่วงค่าเฉลี่ยประมาณ 80 ms ในการวิเคราะห์เราจะใช้วินโด (Window) วิเคราะห์สัญญาณเป็นช่วงๆหรือเรียกว่าการวิเคราะห์เฟรม (Frame) ช่วง

-เวลาของวินด์โรว์มีค่าไม่แน่นอน ในบางครั้งอาจจะต้องใช้ถึง 100 ms แต่ในบางครั้งเมื่อนำมาวิเคราะห์ สัญญาณเสียงที่เร็วมากและไม่มีช่วงหยุดเลย เราอาจจะต้องใช้เวลาของวินด์โรว์ค่าถึง 5-10 ms

4.1.1 การใช้วินด์โรว์ (Windowing)

รูปแบบของการเฉลี่ย โดยปกติมีวิธีหลายอย่างเพื่อให้ได้เส้นของพารามิเตอร์ที่เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้องทางเลือกทางหนึ่งคือการใช้ขนาดของวินด์โรว์ซึ่งมีปัจจัยเกี่ยวข้องคือ

- (1) วินด์โรว์จะต้องเล็กพอ ๆ กับคุณลักษณะของคำพูด
- (2) วินด์โรว์จะต้องยาวพอที่จะใช้ในการคำนวณพารามิเตอร์
- (3) วินด์โรว์ที่ดีจะต้องไม่สั้นไปจนข้ามบางช่องของคำพูด การวิเคราะห์จะ

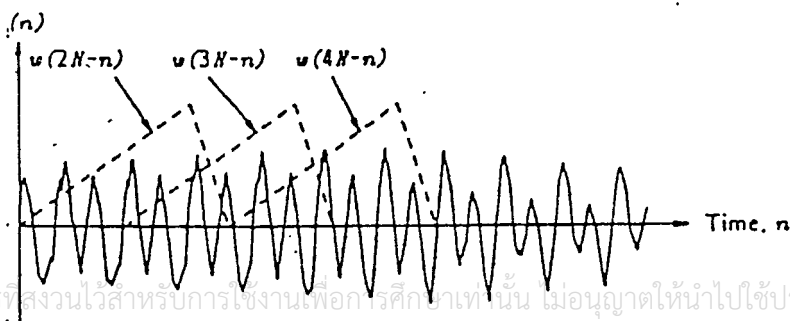
ทำเป็นคาบๆ ไปซ้ำกันตลอดที่ได้สัญญาณ

เงื่อนไขท้ายนี้จะขึ้นอยู่กับ frame rate (จำนวนครั้งต่อวินาที ที่กระทำต่อสัญญาณที่วิเคราะห์) มากกว่าขนาดของวินด์โรว์ โดยปกติ frame rate ประมาณสองเท่าของความถี่เพื่อให้สัญญาณทับกัน 50% การทำบนวินด์โรว์เป็นการคูณสัญญาณเสียงโดยวินด์โรว์ที่มีช่วงเวลาจำกัด (finite duration - window) $w(n)$ ซึ่งกลุ่มของสัญญาณที่สุ่มมาจะถูกให้นำหน้าโดยวินด์โรว์

วินด์โรว์อาจมีคาบเวลาแบบไม่จำกัด แต่ในทางปฏิบัติจะใช้แบบมีจุดสิ้นสุด เพื่อให้ง่ายในการคำนวณโดยการใช้ฟังก์ชัน $w(n)$ ตรวจสอบส่วนต่างๆ ของ $S(n)$ โดยการเคลื่อนวินด์โรว์ตามรูป 4.1 วินด์โรว์ที่มีรูปแบบง่ายที่สุดคือแบบสี่เหลี่ยม $r(n)$

$$w(n) = r(n) = 1 \quad 0 \leq n \leq N-1 \quad (4.1)$$

0 อื่นๆ



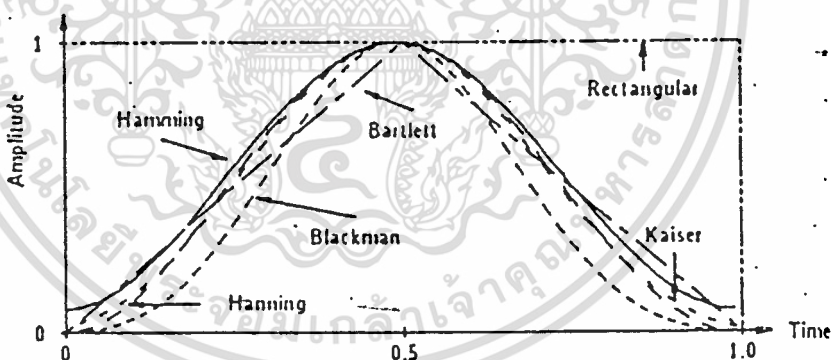
รูปที่ 4.1 แสดงสัญญาณเสียง $s(n)$ กับวางทับสัญญาณของวินโดว์ 3 ตัว โดยมีจุดเริ่มต้นของเวลาที่จุดศูนย์กลาง $2N, 3N,$ และ $4N$

การวิเคราะห์แบบนี้เป็นการกำจัดช่วงจุดศูนย์กลางให้เหลือ N โดยแต่ละจุดศูนย์กลางมีการถ่วงน้ำหนักที่เท่ากัน

วินโดว์ที่ใช้ในความจริงนั้นต้องเป็นไปตามรูปที่ 4.2 สมมุติว่าเสียงพูดมีค่าคงที่ในช่วง 10 ms เราต้องใช้วินโดว์ในช่วงเวลาเท่ากับ 20 ms โดยให้จุดตรงกลางเท่ากับ 10 ms ถ่วงน้ำหนักมากกว่าจุดต้นและจุดปลาย เพราะรูปร่างของวินโดว์มีผลต่อเออาร์พีทต่อพารามิเตอร์เสียงพูด ผลของพารามิเตอร์อาจเปลี่ยนแปลงอย่างมากถ้าใช้ฟังก์ชัน $r(n)$ ต่างกัน ตัวอย่างเช่นการพลังงานโดยการบวกผลกำลัง 2 ของจุดศูนย์กลางในวินโดว์สี่เหลี่ยม ทำให้เกิดความกระเพื่อมรอบจุดความถี่ ดังนั้นเราจะใช้แฮนนิ่งวินโดว์ (Hanning window) แทนวินโดว์สี่เหลี่ยมซึ่งมีรูปร่างเท่ากัน

$$w(n) = h(n) = 0.54 - 0.46 \cos(2\pi n/N - 1) \text{ เมื่อ } 0 \leq n \leq N-1 \quad (4.2)$$

0 อื่นๆ



รูปที่ 4.2 รูปแบบของวินโดว์ ซึ่งมีช่วงเวลาเท่ากับหนึ่ง

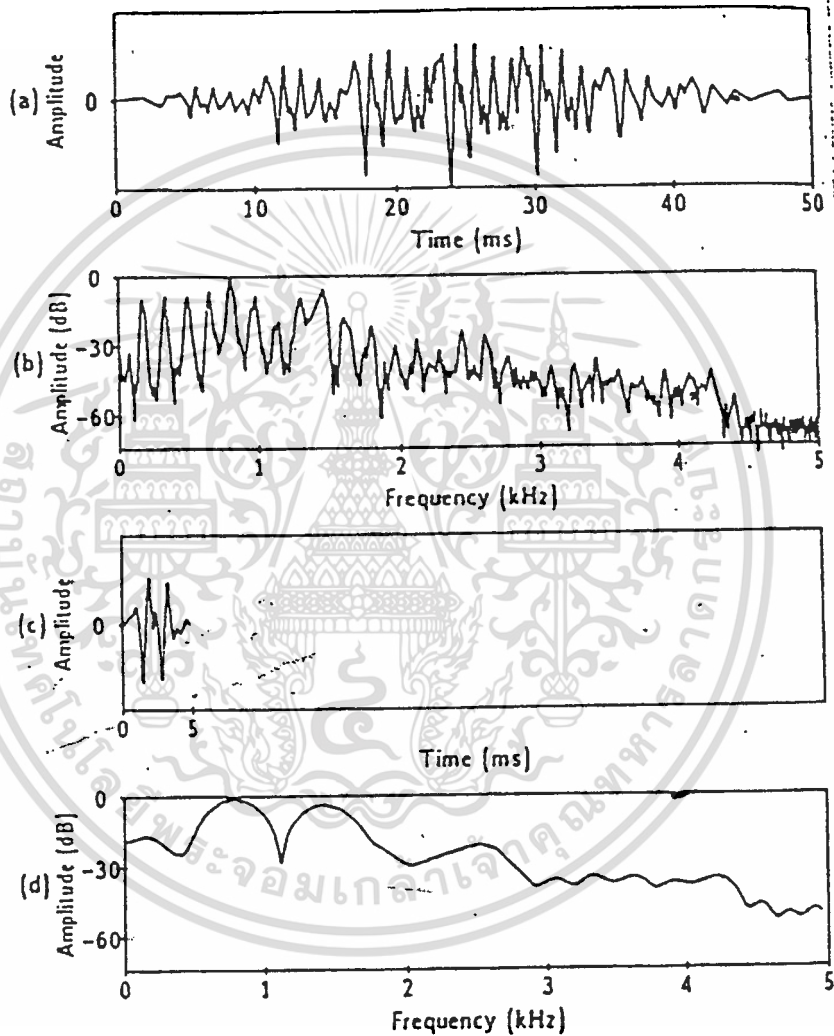
การทำให้ขอบของวินโดว์ลาดลง โดยการวิเคราะห์เฟรมในการเลื่อนตลอดแนวความยาวของสัญญาณทั้งหมด ไม่มีผลเสียต่อพารามิเตอร์ของเสียงพูด

4.2 พารามิเตอร์ในโดเมนเวลา (Time Domain Parameter)

ในขอบข่ายของสัญญาณเสียงในฟังก์ชันเวลาจะมีความได้เปรียบด้านความง่ายคำนวณเร็ว และง่ายในการแปลความหมาย พารามิเตอร์ของเสียงในการเข้ารหัสและการจำจะสามารถทำได้จากการวิเคราะห์ฟังก์ชันของเวลา เช่น พลังงาน (หรือแอมพลิจูด), เสียง

4.2.1 การวิเคราะห์ทีโดเมนเวลา

การวิเคราะห์ของฟังก์ชันเวลาที่เปลี่ยนแปลงจากสัญญาณเสียงสู่พารามิเตอร์ 1 ตัวหรือมากกว่าซึ่งโดยปกติจะช้ากว่าสัญญาณเริ่มแรกมาก การเก็บและการ manipulate ใน



รูปที่ 4.3 (a) แสดงของสัญญาณที่ถูกคูณด้วยแฮมมิงวินด์โ 50 ms
 (b) สเปคตรัมของรูป (a)
 (c) แสดงของสัญญาณที่ถูกคูณด้วยแฮมมิงวินด์โ
 (d) สเปคตรัมของรูป (c)

กรณีของสัญญาณเสียงจะมีประสิทธิภาพสูงกว่าการเก็บสัญญาณเสียงโดยตรงตัวอย่าง เช่น คำพูดที่ถูกสุ่มมาที่ 6000-10000 ตัวอย่าง/วินาที เพื่อที่จะเก็บช่วงความถี่ให้ถึง 3-5 KHz และโดยปกติแล้วการแปลงเสียง 100 ms จะต้องการถึง 1000 ตัวอย่าง เพื่อให้ได้ความถี่ถูกต้องในการแสดงผลออกมามากรวมสัญญาณพารามิเตอร์ส่วนใหญ่จะสุ่มที่ 40-100 ตัวอย่าง/s (มีบางระบบที่ต้องการถึง 200 ตัวอย่าง/s) ดังนั้นเมื่อเปลี่ยนกลุ่มของคลื่นเสียงไปยังกลุ่มพารามิเตอร์ อัตราการสุ่มสามารถลดลง

เทคนิคการประมวลผลแบบ Short time ส่วนมาก จะสร้างพารามิเตอร์ในรูปแบบ

$$Q(n) = \sum_{m=-\infty}^{m=\infty} T[S(m)]w(n-m) \quad (4.3)$$

สัญญาณเสียง $S(n)$ ได้รับการเปลี่ยนเป็นฟังก์ชัน T ซึ่งถูกให้น้ำหนักโดยวินด์โดว์ $w(n)$ และถูกรวมกัน (sum) เป็นพารามิเตอร์ $Q(n)$ ที่อัตราการสุ่มเริ่มต้นซึ่งจะเป็นตัวแทนแสดงคุณสมบัติของคำพูดที่ถูกเฉลี่ยภายใต้ช่วงของวินด์โดว์ $Q(n)$ ได้มาจากการคอนโวลูชันของ $T[S(n)]$ กับ $w(n)$ เพื่อที่จะกระจาย ฟังก์ชันนั้น $w(n)$ ให้เป็น Low pass filter , $Q(n)$ จะถูกทำให้สววยเรียบลงไปในลักษณะของฟังก์ชัน $T[S(n)]$ เดิม เมื่อ $Q(n)$ เป็นเข้าที่พู่ทของวงจรรองผ่านความถี่ต่ำ Low pass Filter (วินด์โดว์) ในกรณีส่วนใหญ่แบนด์วิดของ $Q(n)$ จึงเหมือนวงจรรองผ่านความถี่ต่ำ $w(n)$ เพื่อการทำงานและการจัดเก็บที่มีประสิทธิภาพมากขึ้น $Q(n)$

4.3 ค่าพารามิเตอร์ในโดเมนความถี่

ค่าพารามิเตอร์ที่สำคัญจำนวนมากถูกพบในโดเมนความถี่ เพราะต่อถ้ารเสียงให้การเนิดสัญญาณที่สามารถวิเคราะห์เป็นความถี่ได้ง่ายกว่าในโดเมนเวลาโดยตรง เสียงพูดของคนคนหนึ่งในแต่ละครั้ง เมื่อพิจารณาในโดเมนเวลาจะเห็นผลของความแตกต่างของสัญญาณ แต่เมื่อพิจารณาในโดเมนความถี่ สัญญาณที่ได้มีความคล้ายคลึงกันมาก

ระบบการได้ยินของมนุษย์จะตอบสนองต่อรูปร่างของสัญญาณเสียง (หรือขนาดที่กระจายอยู่ในโดเมนความถี่) มากกว่าที่เฟสของสัญญาณในโดเมนเวลาด้วยเหตุผลเหล่านี้การวิเคราะห์ความถี่จึงถูกใช้ในการดึงค่าพารามิเตอร์ที่สำคัญจากสัญญาณเสียงพูด

ต่อไปจะได้อธิบายถึงวิธีการต่างๆ ที่จะนำไปใช้ในการวิเคราะห์ความถี่

4.3.1 การวิเคราะห์ด้วยฟิลเตอร์แบงก์ (Filter bank)

การวิเคราะห์ความถี่ด้วยฟิลเตอร์แบงก์ เป็นเทคนิคที่นิยมกันมาก เพราะสามารถตอบสนองได้ในเวลาจริง โครงสร้างง่ายและไม่แพงโดยใช้ฟิลเตอร์แบงก์หรือกลุ่มของแบนด์พาสฟิลเตอร์ (Bandpass Filter) ซึ่งอาจเป็นอนาล็อกฟิลเตอร์หรือดิจิทัลฟิลเตอร์ก็ได้ แบนด์พาสฟิลเตอร์แต่ละตัวจะแยกวิเคราะห์ความถี่ของสัญญาณเสียงในช่วงๆ ในทางด้านการค้า วิธีฟิลเตอร์มีความยืดหยุ่นมากกว่าการวิเคราะห์ด้วย DFT เพราะช่วงแบนด์วิดท์ (Bandwidth) สามารถรับขอบเขตของการรับฟังเสียงของมนุษย์ได้มากกว่าการกำหนดตายตัวว่าต้องมีแบนด์วิดท์กว้างแคบเท่านี้เท่านั้น

ในการประยุกต์ใช้งานจำนวนมาก ต้องการกลุ่มของพารามิเตอร์ความถี่ไม่มากในการอธิบายการกระจายของพลังงานความถี่ โดยทั่วไปการใช้กลุ่มของแบนด์พาสฟิลเตอร์ 8-12 ตัวก็สามารถให้ค่าตัวแทนของความถี่ที่ซับซ้อนและมีประสิทธิภาพมากกว่าการวิเคราะห์ด้วย DFT ซึ่งให้รายละเอียดปลีกย่อยของแต่ละความถี่ได้มากกว่า

ในวิธีการทั่วไป ฟิลเตอร์จะมีการระขะตามบาร์คสเกล (bark scale) คือ ให้มีระยะห่างของฟิลเตอร์แต่ละตัวเท่าๆกัน และมีแบนด์วิดท์ที่เพิ่มขึ้นแบบลอการิทึม (logarithm) แต่ไม่ควรเกิน 1 KHz โดยทั่วไปนิยมใช้ฟิลเตอร์ 1/3 ออกเตฟ (one-third-octave filter) ในระบบการจดจำเสียงพูดจริงๆนั้นอาจจะใช้ฟิลเตอร์แบงก์ถึง 2 ครั้ง ครั้งแรกเป็นการจำแนกเสียงอย่างคร่าวๆก่อน โดยใช้ฟิลเตอร์เพียงไม่กี่ตัว และวิเคราะห์หารายละเอียดปลีกย่อยอีกครั้ง โดยใช้กลุ่มของฟิลเตอร์มากขึ้น

4.3.2 การวิเคราะห์ฟูเรียร์เบื้องต้น (Short time fourier analysys)

การวิเคราะห์ฟูเรียร์ช่วงสั้นเป็นเทคนิคการหาความถี่ที่ใช้กันมานานแล้ว การวิเคราะห์ฟู-เรียร์ให้ตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชันความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งตลอดเวลา ดังนั้นจึงจำเป็นต้องทำการวิเคราะห์ในช่วงสั้นๆ โดยใช้วินด์โวล์

การแปลงฟูเรียร์ช่วงสั้นมีสมการว่า

$$S_n(e^{j\omega}) = \sum_{m=-\infty}^{\infty} s(m) [\exp(-j\omega m) w(n-m)] \quad (6.4)$$

ในการคำนวณเรสต้องใช้ DFT แทนการแปลงฟูเรียร์แบบต่อเนื่อง โดยใช้ฟังก์ชันวินด์โวล์ w ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน N ตัว (N คือช่วงเวลาหรือขนาดของวินด์โวล์ที่ใช้ในการแปลง DFT) ข่าวสารต่างๆใน $S_n(e^{j\omega})$ จะไม่สูญหายไปจากข้อมูลเดิม $s_n(e^{j\omega})$ ถ้าการแปลงนั้นสัมพันธ์กับความถี่สูงเพียงพอ (คือ ช่วงระยะห่างระหว่าง N) และวินด์โวล์ $w(\tau)$ ไม่มีจุดศูนย์ที่เป็นศูนย์ตลอดช่วง N ตัวแปร N เป็นตัวแปรที่จะต้องระวังมากเป็น

ละเอียดในโดเมนความถี่จะหายามาก เพราะจะให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำ เฉพาะในช่วงสั้นๆเท่านั้น ในทางตรงกันข้ามถ้า N มีขนาดใหญ่ จะให้ผลของความละเอียด เวลาที่แย่ แต่จะทำให้โดเมนความถี่ที่มีความละเอียดสูงกว่า



บทที่ 5

การแปลงฟูรีเยร์ (fourier transform)

การแปลงฟูรีเยร์เป็นเครื่องมือวิเคราะห์ในงานวิทยาศาสตร์สาขาต่างๆ ในปัจจุบัน การประยุกต์ ใช้งานที่นิยมใช้วิเคราะห์ระบบเชิงเส้นที่ไม่มีการนำไปแปรผันตามเวลา (linear-time invariant system) แต่การแปลงฟูรีเยร์ที่จะกล่าวถึงต่อไปนี้เป็นหลักหรือคุณสมบัติที่สำคัญ ซึ่งสามารถที่จะใช้งานที่วิเคราะห์โดยทั่วไป

5.1 ฟูรีเยร์อินทิกรัล (Fourier integral)

ฟูรีเยร์อินทิกรัล มีคำจำกัดความ เป็นสมการว่า

$$H(f) = \int h(t) e^{j2\pi ft} dt$$

$h(t)$ เป็นเทอมของฟังก์ชันของตัวแปรเวลา และ $H(f)$ เป็นเทอมของฟังก์ชันตัวแปรความถี่ การแปลงฟูรีเยร์ของฟังก์ชันของเวลา จะแทนด้วยตัวอักษรตัวใหญ่

ในรูปทั่วไปการแปลงฟูรีเยร์ มีรูปเป็นจำนวนเชิงซ้อน (complex number)

$$H(f) = R(f) + jI(f) = |H(f)| e^{j\theta(f)} \quad (5.2)$$

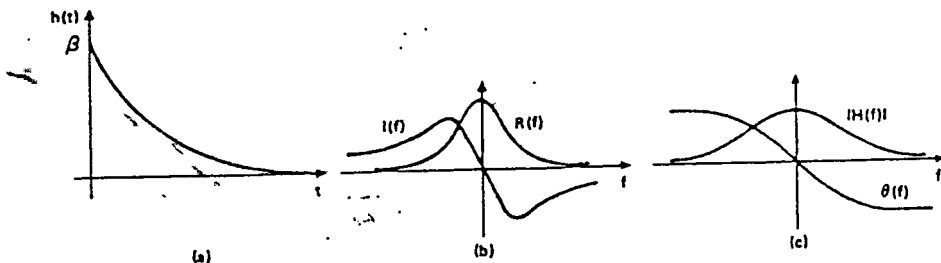
เมื่อ $R(f)$ เป็นส่วนจริงของการแปลงฟูรีเยร์

$I(f)$ เป็นส่วนจินตภาพของการแปลงฟูรีเยร์

$|H(f)|$ เป็น แอมพลิจูดหรือฟูรีเยร์สเปกตรัมของ $h(f)$ ซึ่งได้จาก

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

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



รูปที่ 5.1 (a) ตัวอย่างของฟังก์ชันในโดเมนเวลา

(b) ส่วนจริงและส่วนจินตภาพของการแปลงฟูเรียร์

(c) แอมพลิจูดและเฟสของการแปลงฟูเรียร์

5.2 การแปลงอินเวอร์สฟูเรียร์ (Invers Furier Transform)

การแปลงอินเวอร์สฟูเรียร์ มีค่าจำกัดความดังนี้

$$h(t) = \int H(f) e^{j2\pi ft} df \quad (5.3)$$

5.3 การแปลงฟูเรียร์แบบไม่ต่อเนื่อง (discreate fourier transform) หรือ DFT

DFT เป็นลำดับของความถี่ไม่ต่อเนื่องที่ได้จากการสุ่มในหนึ่งคาบเวลาของการแปลงฟูเรียร์

เมื่อกำหนดให้การสุ่มเท่ากับ N สุ่มบนคาบเวลา $0 < w < 2\pi$

ดังนั้น

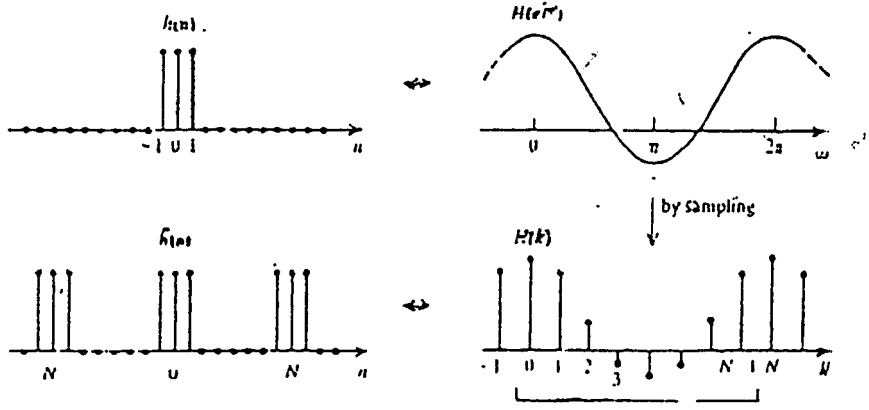
$$wk = 2\pi k/N \quad \text{เมื่อ } 0 < k < N-1 \quad (5.4)$$

ถ้ากำหนดให้ $\{h(n)\}$ เป็นลำดับของ discreat time และลำดับของ Fourier - Transform $H(e^{jw})$ ให้เท่ากับ $\{H(k)\}$ โดยที่

$$H(k) = H(e^{jw}) \quad \text{เมื่อ } w = w_k = 2\pi k/N \text{ และ } 0 < k < N-1 \quad (5.5)$$

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

เราจะเห็นว่า $H(e^{jw})$ มีคาบเวลาเท่ากับ w หรือ 2π ซึ่งได้จากการกระจายอนุกรมฟูเรียร์และคำนวณค่าสัมประสิทธิ์จากลำดับ $\{h(n)\}$ แสดงว่าเมื่อสัญญาณเวลาต่อเนื่องถูกสุ่มว่าคาบเวลาจากการสุ่ม T_s สเปคตรัมจากการแปลงผลลัพธ์ของฟูเรียร์จะมีฟังก์ชันคาบเวลาของความถี่เท่ากับ $2\pi/T_s$ จากรูปที่ 1 เมื่อ $H(e^{j\omega})$ ถูกสุ่มด้วยคาบเวลา $\omega_s = 2\pi/N$ เมื่อแปลงกลับเป็นฟังก์ชันเวลาไม่ต่อเนื่อง $\{h(n)\}$ จะมีคาบเวลา $2\pi/\omega_s = N$



รูปที่ 5.2 แสดงผลลัพธ์ในลำดับของเวลา จาก sampled spectrum ในกรณีนี้ $N = 8$ และอยู่ในช่วง $0 < \omega < 2\pi$

ดังนั้นลำดับ discrete time ที่อยู่ในฟังก์ชันคาบ สามารถหาในเทอมของ $\{h(n)\}$ ได้

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

ลำดับ $\{h(n)\}$ เรียกว่า periodic extension ของ $\{h(n)\}$ จำนวนจุดสุ่มในหนึ่งคาบของ spectrum N จะมีค่าเท่ากับคาบของ $\{h(n)\}$

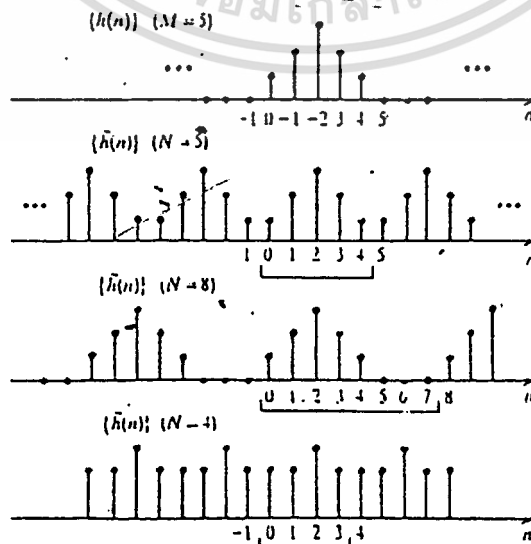
เราสามารถเลือกค่าเท่าใดก็ได้ ซึ่งเป็นการสุ่มของ $H(e^{j\omega})$ ภายใต้ $0 < \omega < 2\pi$ เมื่อ N เป็นคาบของ $\{h(n)\}$ เราต้องไม่เลือกค่าที่ไม่น้อยเกินไป ในลักษณะที่ $\{h(n)\}$ เป็นฟังก์ชันคาบเวลาแบบไม่จำกัด ทำให้เกิดสัญญาณที่ถูกสุ่มทับกัน สัญญาณที่ได้จะเกิดผิดพลาดไป สามารถแสดงจากตัวอย่างข้างล่าง

กำหนด

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

$$= 0 \text{ เมื่อ } M \leq n < N-1$$

จะได้จุดที่เกิดจากการสุ่ม เมื่อ N มีค่าต่างๆ ดังรูป



รูปที่ 5.3 แสดงความสัมพันธ์ของช่วงเวลาของลำดับ M และจำนวนจุดในการสุ่ม

จากรูป จะเห็นว่าถ้าค่า $M > N$ รูปสัญญาณจะทับกัน (overlab) หรือเกิด time-aliasing ซึ่งสามารถป้องกันได้โดยเลือกค่า $N \geq M$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การแปลงฟูรีเยร์ (FFT) และการประยุกต์ใช้งาน

FFT (fast fourier transform) เป็นอัลกอริทึม (algorithm) ที่ช่วยในการคำนวณ DFT มีประสิทธิภาพมากขึ้น ลดจำนวนการกระทำทางคณิตศาสตร์และขนาดของหน่วยความจำที่ต้องใช้ในการคำนวณ DFT อัลกอริทึมของ FFT จึงเป็นเทคนิคที่ถูกเลือกใช้ในการประยุกต์ใช้งานในด้านต่างๆ

6.1 ลักษณะของปัญหา

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

$$C_x(K) = 1/N \sum X(m) W^{km}, \quad k = 0, 1, \dots, N-1 \quad (6.1-1)$$

เมื่อ $W = e^{-i2\pi/N}$, $i = \sqrt{-1}$ เราเรียกสมการ(1) นี้ว่า DFT ของ $\{X(m)\}$ อัลกอริทึมที่เราจะได้กล่าวถึงต่อไปเรียกว่าการแปลงฟูรีเยร์ (FFT) เป็นอัลกอริทึมที่พัฒนาโดย Cooley และ Tukey บางทีอาจจะเรียกว่าอัลกอริทึมของ Cooley และ Tukey ก็ได้ ตั้งแต่นี้ไป เราจะตั้งข้อสมมุติว่า

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

จากที่เราได้เข้าใจกันแล้ว ถ้าไม่ต้องการให้ข้อมูลสูญหาย ต้องมีขนาด N ที่เหมาะสม ตามทฤษฎีการสุ่ม (Sampling theorem) นั่นคือ $N \geq 2BL$ เมื่อ B เป็นแบนวิดธ์ของ $x(t)$ ในหน่วย Hz และ L คือช่วงเวลาของมัน สำหรับในกรณีของ N ไม่เท่ากับ 2^n เราจะได้พูดถึงกันต่อไป

เราจะพิจารณาในกรณีที่ $\{X(m)\}$ เป็นค่าจำนวนจริง และ $N = 8$ จากคุณสมบัติคอนจูเกต (complex conjugate property) เรามี

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

ดังนั้นเมื่อเทียบกับสมการที่ (5.1-1) ทำให้การคำนวณเหลือเพียง

$$C_X(k) = 1/8 \sum X(m) W^{km}, \quad k = 0, 1, \dots, 4 \quad (6.2-1)$$

เมื่อ $W = e^{-i2\pi/4}$

จากสมการ (1) จะได้ว่า

$$8 C_X(k) = \sum X(m) \cos(mk\pi/4) - i \sum X(m) \sin(mk\pi/4) = A(k) - B(k)$$

เมื่อ

$$A(k) = \sum X(m) \cos(mk\pi/4),$$

และ

$$B(k) = \sum X(m) \sin(mk\pi/4), \quad k = 0, 1, \dots, 4 \quad (6.2-2)$$

ใช้สัญลักษณ์เมทริกซ์ สมการที่ (2) จะเป็น

$$A = CX$$

$$B = SX$$

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

$$S =$$

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

C =

$$\begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
 1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} & -1 & -1/\sqrt{2} & 0 & 1/2 \\
 1 & 0 & -1 & 0 & 1 & 0 & -1 & 0 \\
 1 & -1/\sqrt{2} & 0 & 1/\sqrt{2} & -1 & 1/\sqrt{2} & 0 & -1/\sqrt{2} \\
 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1
 \end{array}$$

เมื่อพิจารณาคุณสมบัติ(3) จะเห็นว่าสมาชิกของเมทริกซ์ C และ S มีค่าซ้ำกัน ซึ่งเป็นเหตุผล 2 ประการที่พัฒนาอัลกอริทึมฟาสต์ฟูเรียร์ ดังนี้

1. ความเป็นซายน์(sinusoidal) จัดกลุ่มด้วยลักษณะเด่น ซึ่งจะเกิดขึ้นได้เมื่อเลือกจุดสุ่มที่ได้ถูกต้อง

2. ถ้า N เป็นจำนวนของจุดสุ่มของ $x(t)$ ที่มีรูปแบบไม่แน่นอนแล้ว การเกิดซ้ำกันของเมทริกซ์ C และ S จะเพิ่มขึ้นตามจำนวนตัวประกอบของ $N = 2 \cdot n = 2 \times 2 \times \dots \times 2$ ดังนั้น เมื่อจำนวนของข้อมูลมาก การใช้อัลกอริทึมฟาสต์ฟูเรียร์จะช่วยประหยัดเวลาและหน่วยความจำเป็นจำนวนมาก

6.3 หนทางการพัฒนาอัลกอริทึม

หนทางที่จะพัฒนาอัลกอริทึมตามที่ต้องการคือการหาความสัมพันธ์กับการซ้ำกันในรูปแบบทั่วไปของเมทริกซ์ C และ S กับกำลังของ W หรือหลักการที่ N ของหนึ่ง (principle N-th root of unity) เพื่อหาสัญลักษณ์ของรูปทั่วไป

สัญลักษณ์ แสดงค่าในฐานสิบแต่ละตัวของ $m, 0 \leq m \leq N-1$ ให้อยู่ในรูปของเลขฐานสองหรือไบนารี (Binary) ดังต่อไปนี้

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

เมื่อ

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

ในทำนองเดียวกัน ค่าในฐานสิบแต่ละตัวของ 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

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

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

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

(6.3-1)

เปลี่ยนรูปแบบของสมการ (1) ตามความสัมพันธ์ที่ได้ดังนี้

$$\sum X(m)W^{km} = \sum \sum \dots \sum X(m_{n-1}, m_{n-2}, \dots, m_1, m_0)W^{k(m_{n-1}2^{n-1} + \dots + m_02^0)} \quad (6.32)$$

6.4 การพัฒนาอัลกอริทึม

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

$$C_X(k) = (1/8) \sum X(m)W^{km}, k = 0, 1, \dots, 7 \quad (6.4-1)$$

เมื่อ $W = e^{-j2\pi/4}$ เราแสดงค่าของ m ในรูปแบบของไบนารี ดังนี้

$$m = m_22^2 + m_12^1 + m_02^0 \quad (6.4-2)$$

จากสมการ (6.3-2) สมการที่ (1) และ (2) จะได้ว่า

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

(6.4-3)

จากสมการที่ (3) ผลบวกในสุดของ m_2 แทนด้วยสัญลักษณ์ M_2 จะได้ว่า

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

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

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

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น หากมีข้อผิดพลาดประการใด ขออภัยและสงวนสิทธิ์ในเอกสารทุกครั้งที่มีการนำไปใช้

$$M_2 = \sum X(m_2, m_1, m_0) (-1)^{k_0 m_2} \quad (6.4-4)$$

จากสมการ (4) ผลบวกของ m_2 แสดงค่าที่ได้ในรูปฟังก์ชันของ k_0, m_1, m_0

$$M_2 = \sum X(m_2, m_1, m_0) (-1)^{k_0 m_2} = X_1(k_0, m_1, m_0) \quad (6.4-5)$$

แทนสมการ (5) ลงในสมการ (3)

$$8C_X(k) = \sum \sum X_1(k_0, m_1, m_0) W^{2k m_1} W^{k m_0} \quad (6.4-6)$$

ทำซ้ำอีกครั้งในส่วนในสุดของสมการ (6) และแทนด้วยสัญลักษณ์ M_1

$$\begin{aligned} M_1 &= \sum X_1(k_0, m_1, m_0) W^{2k m_1} \\ &= \sum X_1(k_0, m_1, m_0) (-i)^{(4k_2 + 2k_1 + k_0) m_2} \end{aligned} \quad (6.4-7)$$

ในสมการ (7) จำนวน $(i)^{4k_2 m_1} = 1$ จะได้รูปแบบที่ง่ายขึ้น ดังนี้

$$M_1 = \sum X_1(k_0, m_1, m_0) (-i)^{(2k_1 + k_0) m_1} \quad (6.4-8)$$

ผลบวกบน M_1 จะได้ผลลัพธ์ M_1 ที่เป็นฟังก์ชันของ k_0, k_1, m_0

$$M_1 = X_2(k_0, m_1, m_0) \quad (6.4-9)$$

แทนสมการ (9) ลงในสมการ (6) จะได้

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

ซึ่งจะได้

$$M_0 = \sum X_2(k_0, k_1, m_0) [(1-i)/\sqrt{2}]^{(4k_2 + 2k_1 + k_0) m_0} \quad (6.4-10)$$

การทำผลบวกบน m_0 จะได้ M_0 ที่เป็นฟังก์ชันของ k_0, k_1, k_2 ดังนี้

$$M_0 = X_3(k_0, k_1, k_2) \quad (6.4-11)$$

$$8C_X(k) = 8C_X(k_2, k_1, k_0) = X_3(k_0, k_1, k_2) \quad (6.4-12)$$

ขั้นแรกในการคำนวณหาค่าตอบพิจารณา

1. $N = 8$ ทำให้ผล $\log 2N = 3$
2. coefficient $(-1), (-i), [(1-i)/\sqrt{2}]$ มีความสัมพันธ์กับรากที่

$$A_2 = e^{-i2\pi/2^r}, r = 1, 2, \dots, \log_2 N \quad (6.4-13)$$

เมื่อ A_2 เป็นรากที่ 2^r ของ unity โดยมีคุณสมบัติดังนี้

$$1. A_2 = W^{N/2}, W = e^{-i2\pi/n} \quad (6.4-14a)$$

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

$$; r = 1, 2, \dots, \log_2 N, 1 = 0, 1, \dots, 2^{r-1} \quad (6.4-14b)$$

เมื่อ $= 2^{r-1}$

$$3. (A_N)^{N/2} = -1 \quad (6.4-14c)$$

จากสมการข้างต้นสามารถเขียนได้ในเทอม A_2 จะได้

$$X_1(k_0, m_1, m_2) = \sum X(m_2, m_1, m_0) A_2^{k_0 m_2} \quad (6.4-15)$$

ในสมการ(15) k_0 มีค่า 0 หรือ 1 ทำให้ได้สมการ 4 สมการ ในแต่ละค่า k_0

$$\text{กรณี 1 } k_0 = 0 \quad (6.4-16a)$$

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

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

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

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

$$\text{กรณี 2 } k_0 = 1 \quad (6.4-16b)$$

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

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

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

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

จากสมการที่ 9 ในเทอม A_4

$$X_2(k_0, k_1, m_0) = X_1(k_0, 0, m_0) + X_1(k_0, 1, m_0) A_4^{2k+k} \quad (6.4-17a)$$

$$\text{กรณี 1 } k_1, k_0 = 0, 0$$

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

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

$$\text{กรณีที่ 2 } k_1, k_0 = 0, 1 \quad (6.4-17b)$$

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

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

กรณีที่ 3 $(k_1, k_0) = (1, 0)$ (6.4-17c)

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

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

กรณีที่ 4 : $(k_1, k_0) = (1, 1)$ (6.4-17d)

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

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

สมการ (11) เขียนในเทอม A_8 จะได้

$$X_2(k_0, k_1, k_2) = X_2(k_0, k_1, 0) + X_2(k_0, k_1, 1) A_8^{(4k_2 + 2k_1 + k_0)} \quad (6.4-$$

18)

กรณี 1 : $(k_2, k_1, k_0) = (0, 0, 0)$ (6.4-19a)

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

กรณี 2 : $(k_2, k_1, k_0) = (0, 0, 1)$ (6.4-19b)

$$X_3(1,0,0) = X_2(1,0,0) + A_8 X_2(1,0,1) \implies X_3(4) = X_2(4) + A_8 X_2(5)$$

กรณี 3 : $(k_2, k_1, k_0) = (0, 1, 0)$ (6.4-19c)

$$X_3(0,1,0) = X_2(0,1,0) + A_8 X_2(0,1,1) \implies X_3(2) = X_2(2) + A_8 X_2(3)$$

กรณี 4 : $(k_2, k_1, k_0) = (0, 1, 1)$ (6.4-19d)

$$X_3(1,1,0) = X_2(1,1,0) + A_8 X_2(1,1,1) \implies X_3(6) = X_2(6) + A_8 X_2(7)$$

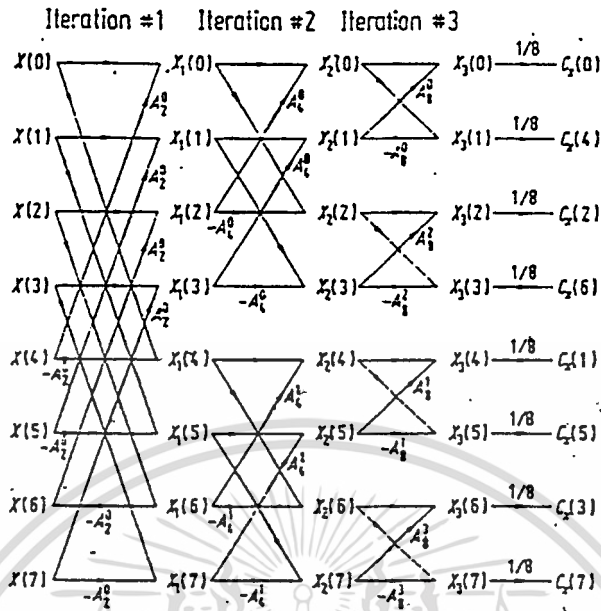
กรณี 5 : $(k_2, k_1, k_0) = (1, 0, 0)$

$$X_3(0,0,1) + X_2(0,0,0) + A_8 X_2(0,0,0) \implies X_3(1) = X_2(0) - X_2(1)$$

ลำดับการทำงานทางคณิตศาสตร์ตั้งแต่สมการ (19a) ถึงสมการ (19e) ได้แสดงไว้ในรูปที่ 6.1 และการทำซ้ำ (iteration) ครั้งที่ 3 ก็คือ $r = 3$ ในสมการที่ (13)

เพื่อให้ได้ค่า $C_x(k)$ ตามที่ต้องการใช้สมการ

$$8C_x(k_2, k_1, k_0) = X_3(k_0, k_1, k_2) \quad (6.4-20)$$



รูปที่ 6.1 แสดงแผนภาพการไหล (Signal flow-graph) ของสัญญาณ FFT เมื่อ $N = 8$

จะเห็นว่าการสลับตำแหน่งของสัมประสิทธิ์ไบนารี k_0, k_1 และ k_2 ใน $X_3()$

เมื่อเทียบกับตำแหน่งจริง $C_x(k)$

เราเรียกลักษณะเช่นนี้ว่า ปรากฏการณ์การกลับบิต (bit reversal) $X_3()$ มีความสัมพันธ์กับ $C_x(k)$ ตามสมการข้างล่างนี้

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

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

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

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

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

(6.4-21)

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

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

จะได้

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

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

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

เอกสารนี้เป็นเอกสาร $C_x(5), C_x(6), C_x(7)$ เราสามารถหาได้จากสมการ (12) และ (18) ประโยชน์ด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$8C_X(5) = X_3(5) = X_2(4) \cdot A_8 X_2(5)$$

$$8C_X(6) = X_3(3) = X_2(2) \cdot A_8 X_2(3)$$

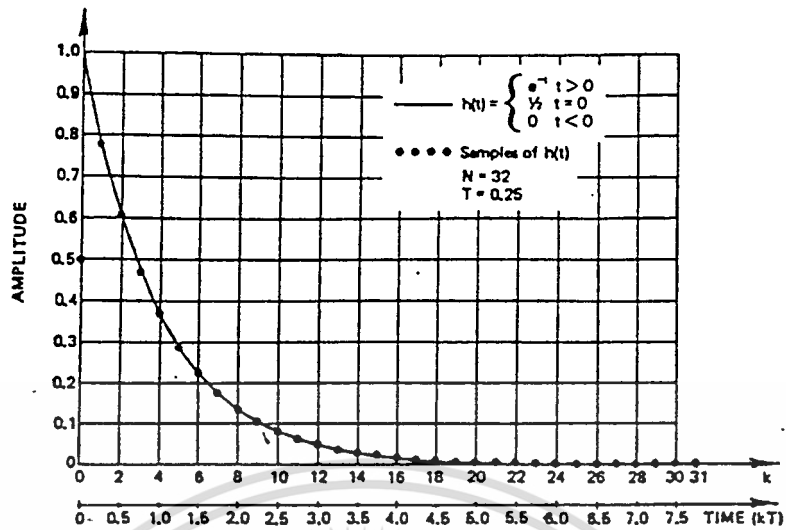
$$8C_X(5) = X_3(7) = X_2(6) \cdot A_8 X_2(7)$$

(6.4-22)

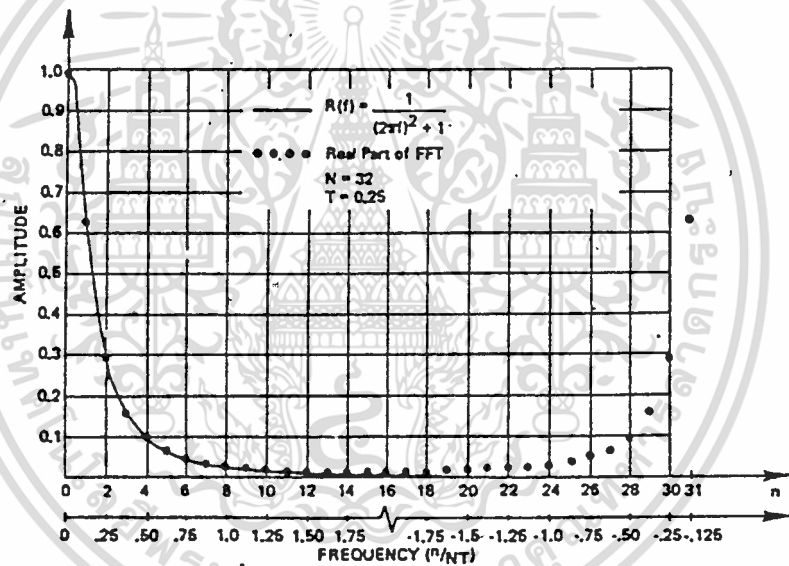
จากขั้นตอนข้างต้นเป็นขั้นตอนและแนวทางในการเขียน โปรแกรม FFT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 6.2 แสดงตัวอย่างการแปลงฟูเรียร์ โดยผ่าน FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

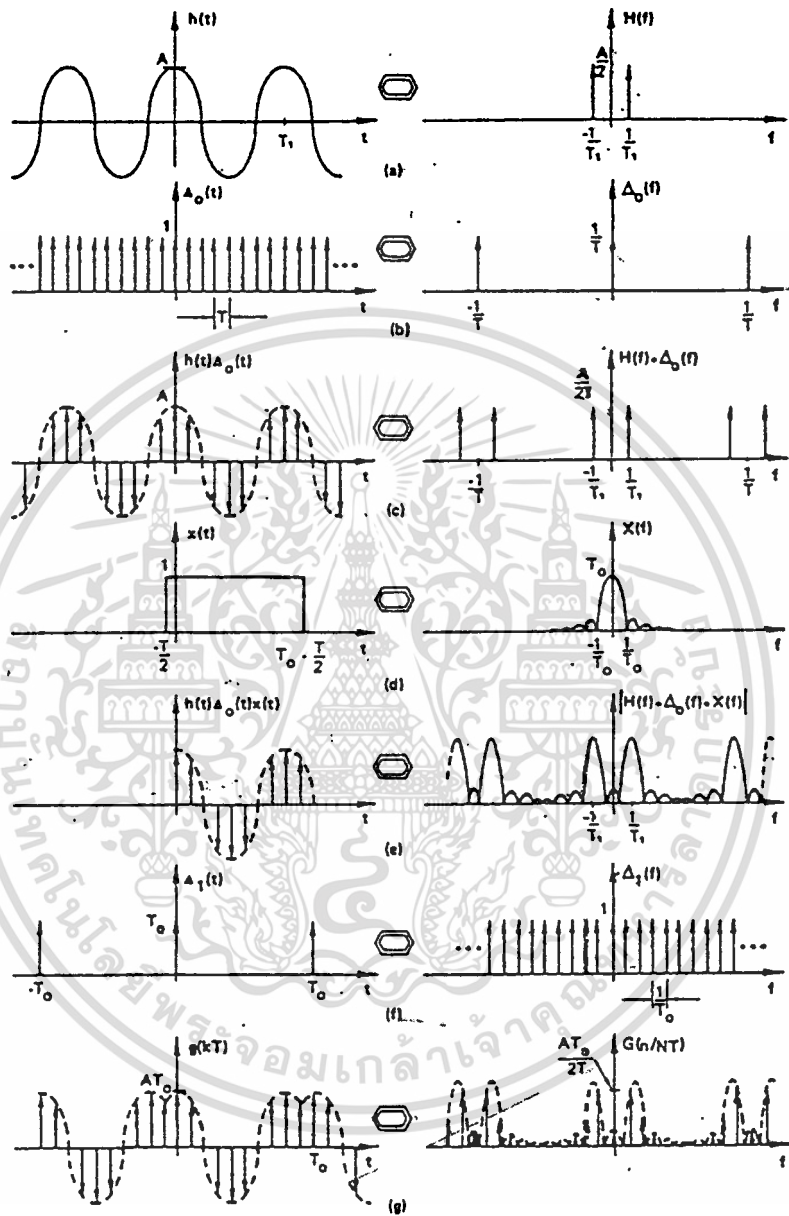
6.5 การประยุกต์ใช้งาน FFT

6.5.1 ความละเอียด(resolution)ของ FFT

ผลลัพธ์ของ FFT ในรูปที่ 5.2(b) และ (c) แต่ละความถี่มีระยะห่าง $f_0 = 1/NT$ ดังนั้นจุดสุ่มในโดเมนจึงมีค่าตั้ง $0/NT, 1/NT, 2/NT, \dots, (N/2)/NT$ สำหรับด้านความถี่บวก ระยะห่างของความถี่ $f_0 = 1/NT$ เป็นเทอมที่แสดงความละเอียดของ FFT แต่ละสมาชิกของความถี่ที่ได้เรียกว่าสมาชิกความละเอียด (Resolution Element) หรือเซลล์ความละเอียด (Resolution -Cell) เราอาจจะคิดว่าความละเอียดเป็นเทอมที่ทำให้เราสามารถแยกความแตกต่างของแต่ละความถี่ โดยประมาณค่าความถี่ที่ได้จากการแปลงเป็นความถี่ตั้งแต่ $0/NT, 1/NT, 2/NT, \dots, (N/2)/NT$ ซึ่งเราอาจจะลดระยะความห่างของแต่ละความถี่ได้ โดยการเพิ่มจำนวนข้อมูล N ซึ่งก็คือการเพิ่มระยะเวลาการตัดทอน (Truncation) ของฟังก์ชันที่ต้องการแปลง ถ้าค่าของ N เพิ่มขึ้นเป็นสองเท่า ระยะห่างของความถี่ก็จะลดลงสองเท่าเช่นกัน

จากรูปที่ 5.3 ระยะห่างของความถี่ (ความละเอียด) ของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ถูกกำหนดจากความกว้างของสี่เหลี่ยมที่ไปคูณหรือตัดทอนฟังก์ชันก่อนทำการแปลง การตัดทอนในโดเมนเวลาก็คือ การทำคอนโวลูชันกับฟังก์ชัน $[\sin(f)/f]$ กับการแปลงของฟูเรียร์ฟังก์ชันเริ่มต้น การทำคอนโวลูชันกับฟังก์ชัน $[\sin(f)/f]$ ทำให้ผลการแปลงคลุ่มเครือและเลือนลาง ถ้าฟังก์ชันที่ตัดทอนในโดเมนเวลา มีความกว้างมากขึ้น จะทำให้ผลของฟังก์ชันแคบลง และความคลุ่มเครือของความถี่ลดลงด้วย ความคลุ่มเครือของความถี่ยิ่งลดน้อยลงเท่าใด ก็ยิ่งทำให้ประสิทธิภาพของการใช้ FFT ในการแก้ปัญหาในแต่ละงาน เป็นไปได้มากยิ่งขึ้น

ความเข้าใจผิด โดยทั่วไปของผู้ที่ใช้ FFT คือ การเพิ่มข้อมูลจำนวน N โดยการต่อเติมศูนย์และตัดทอนฟังก์ชันและตีความหมายของฟังก์ชันความถี่ที่ได้เป็นฟังก์ชันที่มีความละเอียดสูง ตอนนี้เรามาเปรียบเทียบรูปที่ 6.2 กับรูปที่ 6.4 ในรูปที่ 6.4a แสดงผลจากการแปลงดิสครีตฟูเรียร์ เช่นเดียวกับรูปที่ 6.2 (ข) เราต้องการตรวจสอบผลของการเพิ่มศูนย์ให้กับฟังก์ชันเวลาของรูปที่ 5.4(a) สมมุติว่าจำนวนของศูนย์ที่เพิ่มเข้าไปนั้น



รูปที่ 6.3 แสดง DFT ของสัญญาณคาบเวลา โดยมีช่วงการตัดทอนน้อยกว่า 1 คาบ
เวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

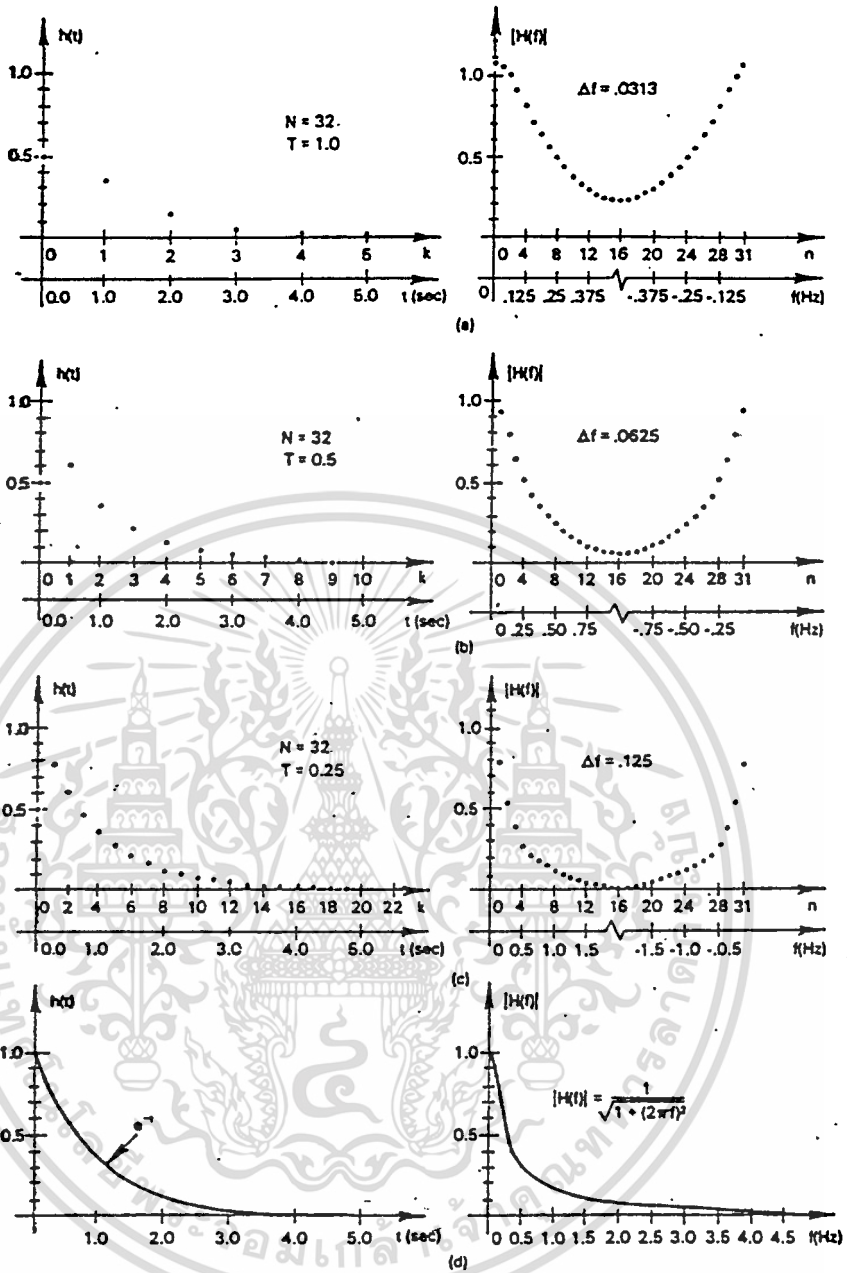
ลูชันของฟังก์ชันความถี่ของรูปที่ 6.4(a) และรูปที่ 6.4 (b) แต่ความละเอียดของความถี่ได้ถูกกำหนดไว้ในรูปที่ 6.4(a) เรียบร้อยแล้ว การทำคอนโวลูชันเป็นการเพิ่มจุดสุ่มในโดเมนความถี่โดยการเฉลี่ยค่า (interpolation) ด้วยฟังก์ชัน $[\sin(f)/f]$ กับผลการแปลงฟูเรียร์ของฟังก์ชันเริ่มต้น ดังนั้น ถึงแม้ว่าความห่างของระยะจาก FFT จะมีระยะใกล้ขึ้นโดยการต่อเติมศูนย์ นอกเสียจากว่าฟังก์ชันที่จะต่อเติมศูนย์มีค่าครอบคลุมช่วงนั้นอยู่แล้ว

จากหัวข้อนี้ เราอาจสรุปได้ว่า ความละเอียดนั้นจะต้องพิจารณาจากช่วงเวลาในโดเมนเวลาของสัญญาณและในการประยุกต์ใช้ FFT ช่วงเวลาของสัญญาณถูกตั้งใหม่เป็นช่วงระยะห่างของการตัดทอนข้อมูล

6.5.2 ผลความผิดเพี้ยนจาก FFT (FFT aliasing)

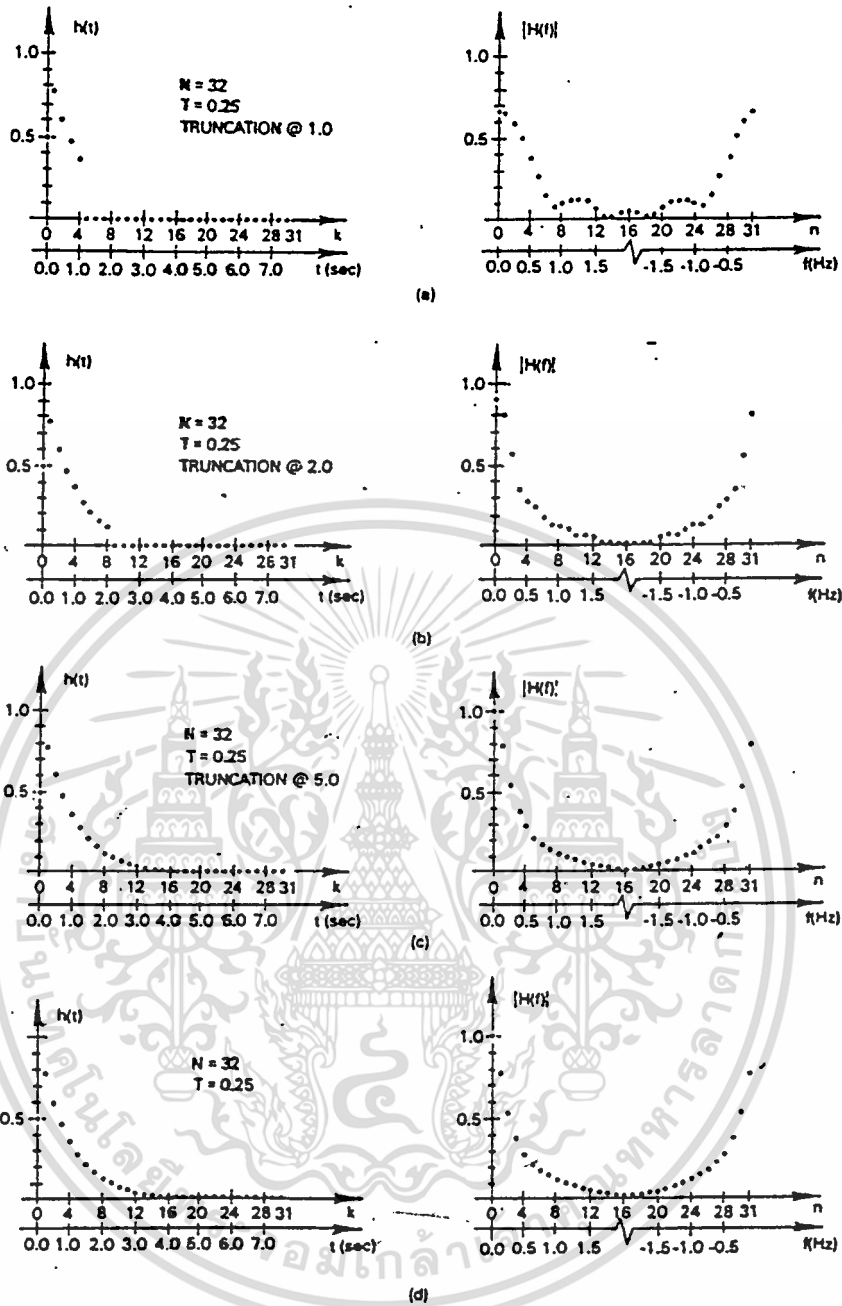
ปัญหาหนึ่งที่เราพบในการแปลงฟูเรียร์จาก FFT คือความผิดเพี้ยน (aliasing) จากที่เคยกล่าวมาแล้วในการสุ่มข้อมูล ความผิดเพี้ยนจะเกิดเมื่อจุดสุ่มของฟังก์ชันเวลา มีระยะห่างของการเก็บมากเกินไปผลของความถี่ที่ผิดเพี้ยนนี้เรียกว่า การพับ (fold) หรือการซ้อน (overlap) ในตัวของมันเอง ซึ่งได้แสดงตัวอย่างในรูปที่ 6.5

ในรูปที่ 6.5 เรามีจุดสุ่มของฟังก์ชัน $h(t) = e^{-t}, t > 0$ โดยช่วงเวลาการสุ่ม $T = 1.0, 0.5$ และ 0.25 ตามลำดับ จำนวนข้อมูลตั้งให้เท่ากับ 32 สำหรับแต่ละกรณี ขนาดของการแปลงฟูเรียร์ที่ได้ กำหนดโดยการใช้ FFT ซึ่งได้แสดงไว้ในรูปที่ 6.5(d) จะเห็นว่าความผิดเพี้ยนมีค่าลดลง เมื่อมีคาบเวลาเท่ากับ 0.5 และเมื่อคาบเวลาเท่ากับ 0.25 ผลที่ได้จะมีความคล้ายคลึงกับผลที่ได้จากการคำนวณแบบต่อเนื่องมากที่สุด



รูปที่ 6.5 แสดงตัวอย่างของความผิดเพี้ยนในโดเมนความถี่จากอัตราการสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6.6 แสดงการตัดทอนในโดเมนเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 6.5 ได้แสดงหลักการลดความผิดเพี้ยน โดยการลดช่วงการสุ่ม ในกรณีนี้จะเห็นว่าไม่มีผลของการตัดทอน เนื่องจาก T มีค่าต่ำ จึงทำให้ NT มีค่ามากกว่าความกว้างของช่วงไม่เป็นศูนย์ (nonzero interval) ของฟังก์ชันมาก

6.5.3 การตัดทอนของ FFT ในโดเมนเวลา (FFT Time - Domain Truncation)

ปัญหาความผิดพลาดอื่นที่มักพบจากการประยุกต์ใช้ FFT ในการแปลงฟูเรียร์ คือความผิดพลาดเนื่องจากการแปลงข้อมูลในโดเมนเวลา (time domain truncation) ความผิดพลาดนี้เกิดจากการเลือกกลุ่มข้อมูลจุดสุ่มเพื่อหาลักษณะบางส่วนของรูปคลื่นเบื้องต้น รูปที่ 6.6(a) และ รูปที่ 6.6(b) แสดงตัวอย่างของจุดนี้ เมื่อเราตัดทอนฟังก์ชัน $h(t)$ ที่ $NT = 1, 2, 5 S$. ตามลำดับ ขนาดของการแปลงฟูเรียร์ที่ผ่าน FFT ได้แสดงในรูป

การตัดทอนที่ 1 วินาที ทำให้ได้ผลจาก FFT มีการกระเพื่อม (rippling) เมื่อใช้ช่วงเวลาของการตัดทอนเพิ่มขึ้น เท่ากับ 2 วินาที ผลของ FFT ที่ได้มีการกระเพื่อม น้อยลง และเมื่อเพิ่มช่วงเวลาการตัดทอนให้เป็น 5 วินาทีผลของ FFT ที่ได้ไม่ปรากฏผลของการเกิดการกระเพื่อม เหมือนกับรูปที่ถูกตัดในรูปที่ 6.6 (c)

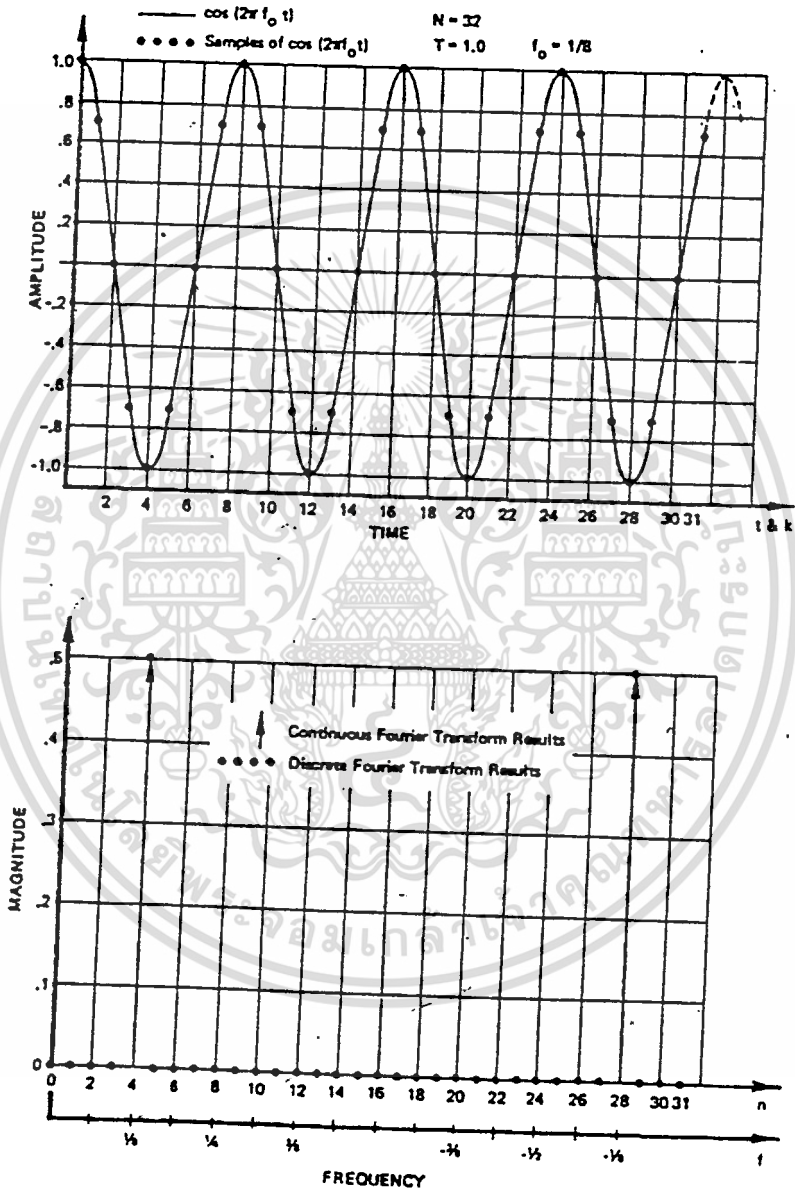
ในรูปที่ 6.6 แสดงผลของการทดลอง โดยการพิจารณาช่วงการตัดทอนที่ดีที่สุด โดยการเพิ่มช่วงเวลาการตัดทอน ซึ่งเราก็ได้เห็นผลของการลดลงของการกระเพื่อม

6.5.4 FFT ของฟังก์ชันคาบเวลา

การคำนวณหา FFT ของฟังก์ชันคาบเวลา เราจะต้องเลือกช่วงเวลาการสุ่ม หรือการตัดทอน จากที่กล่าวมาแล้ว การเลือก T ต้องเลือกให้เหมาะสมเพื่อไม่ให้เกิดการผิดเพี้ยน การตัดทอนของฟังก์ชันเวลาคาบเวลาเป็นปัญหาใหม่ที่ไม่ได้อธิบายในหัวข้ออื่นมาก่อน มาก่อน

ถ้าเราเลือกจำนวนจุดสุ่ม N ของการแปลงฟูเรียร์ เท่ากับ หนึ่งคาบเวลาพอดีหรือเท่ากับจำนวนเท่าของคาบเวลา ผลของความถี่ที่ได้จะตรงกับสัญญาณเริ่มต้น

เพื่ออธิบายจุดนี้ เราทำการคำนวณ FFT ของฟังก์ชันโดซายน์ในรูปที่ 6.7 (a) โดยมีช่วงการสุ่ม $T = 1.0S$ และจำนวนจุดสุ่มเท่ากับ 32 และแสดงผลลัพธ์ที่ได้ในรูปที่ 6.7(b) โดยแสดงเป็นขนาดของ FFT ของจุดเหล่านี้จากรูปจะเห็นว่าจุดอื่นเป็นศูนย์หมด นอกจากจุดที่เป็นความถี่จริง



รูปที่ 6.7 แสดงผลจาก FFT ของฟังก์ชันคาบเวลา โดยมีช่วงการตัดทอนเท่ากับจำนวน

เท่าของคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.6 ฟังก์ชันถ่วงน้ำหนักข้อมูล (data weighting function)

จากหัวข้อที่แล้วการตัดทอน (truncation) อาจจะทำให้ผลจากการแปลงฟูเรียร์คลาดเคลื่อน การประมวลผลข้อมูลจำนวนจำกัด N ของฟังก์ชันคาบเวลาที่ไม่รู้เวลานแน่นอน จำเป็นต้องใช้วินโดว์ของข้อมูลหรือที่เรียกว่าฟังก์ชันถ่วงน้ำหนักของข้อมูลในหัวข้อนี้เราจะได้อธิบายเทคนิคการใช้ฟังก์ชันถ่วงน้ำหนักเพื่อลดผลเสียจากการตัดทอนให้เหลือน้อยที่สุด

6.6.1 ฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยม (rectangular weighting function)

จากรูปที่ 6.3 เราสามารถสังเกตุการณ์โดยการคูณด้วยอิมพัลส์ฟังก์ชัน (impulse function) ซึ่งได้แสดงที่รูปที่ 6.3(b) นำผลที่ได้ (รูปที่ 6.3 c) ไปคูณด้วยฟังก์ชันตัดทอนสี่เหลี่ยม ซึ่งแสดงไว้ในรูปที่ 6.3 (d) เพื่อจำกัดจำนวนของจุดสุ่มเพื่อให้เหลือเท่ากับ N ในที่นี้การตัดทอนของโดเมนเวลา (time domain) ก็คือการถ่วงน้ำหนักของข้อมูลโดยการคูณด้วยฟังก์ชันสี่เหลี่ยม ผลของการตัดทอนเวลาได้แสดงไว้ในรูปที่ 6.2 (e)

เรารู้ว่าฟังก์ชันโดเมนความถี่ (frequency domain) ของฟังก์ชันสี่เหลี่ยม คือฟังก์ชัน $[\sin(f)/f]$ ดังนั้นการแปลงเป็นโดเมนความถี่ของฟังก์ชันเวลาที่ตัดทอนแล้ว จึงเป็นการทำคอนโวลูชัน-

(convolution) ของฟังก์ชันเวลากับฟังก์ชันสี่เหลี่ยม ทำให้ผลที่ได้มีองค์ประกอบของความถี่เพิ่มขึ้นมาที่เรียกว่า ซ้ายข้าง (side lobe) ของฟังก์ชันสี่เหลี่ยม องค์ประกอบที่เพิ่มขึ้นมานี้เรียกว่า ส่วนรั่วไหล (leakage) ทำให้อิมพัลส์ของความถี่ของฟังก์ชันรั่วออกไปที่ลอนข้างของฟังก์ชัน $[\sin(f)/f]$

ถึงแม้ในฟังก์ชันเวลาในโดเมนเบื้องต้นเป็นสัญญาณชายนี้น แต่เมื่อผ่านการสุ่มแล้ว รูปคลื่นของสัญญาณที่ผ่านในโดเมนเวลาจะไม่ใช่สัญญาณชายนี้อีก เพราะว่าช่วงของการตัดทอน (truncation interval) ไม่เท่ากับคาบเวลาหรือจำนวนเท่าของคาบเวลา (เป็นจำนวนเต็ม) ทำให้คอนโวลูชันในรูปที่ 6.2(e) และ (f) ไม่ตรงกับฟังก์ชันคาบเวลาเบื้องต้น เนื่องจากการทำคอนโวลูชันของฟังก์ชันเวลาเป็นแบบไม่ต่อเนื่อง (discontinuous) จึงได้แสดงผลของการเกิดการกระเพื่อม (rippling effect) ไว้ในแบบไม่ต่อเนื่องในรูปที่ 6.2 (g) ด้วย

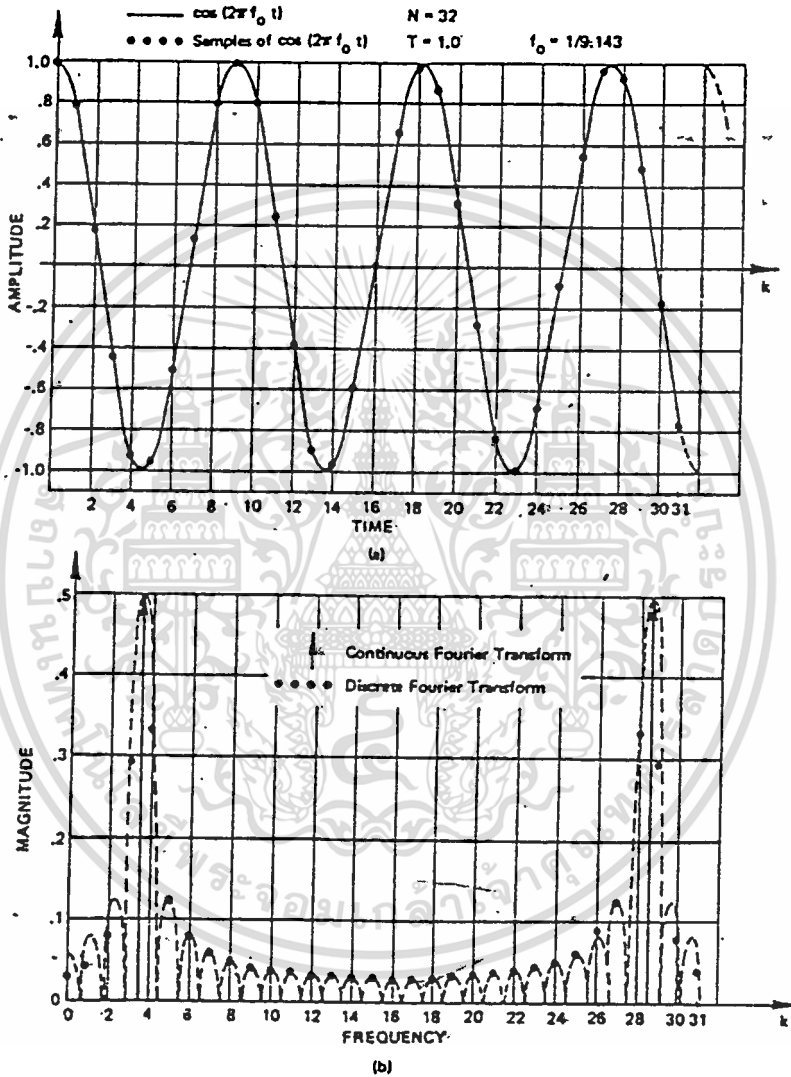
ต่อไป เราสาธิตตัวอย่างของการเกิดของฟังก์ชันถ่วงน้ำหนักของสี่เหลี่ยมดังนี้

สมมุติว่า เราต้องคำนวณ FFT ของฟังก์ชันโคไซน์ (cosin function) ซึ่งแสดงไว้ในรูปที่ 9.7 โดยมีคาบเวลา $T = 1.0$ และ $N = 32$ ในรูปที่ 9.7 (b) เราได้แสดงขนาดของการแปลงดิสครีตฟูเรียร์ (discrete fourier transform) ของจุดสุ่มในรูปที่ 9.7 (a) FFT จะทำให้ผลเป็น

-ความถี่ไม่ต่อเนื่องทางด้านบวกทั้งหมด จากที่ได้อธิบายในหัวข้อที่แล้ว องค์ประกอบของความถี่เพิ่มขึ้นมาที่เรียกว่าส่วนรัวไหลเป็นผลมาจากการมีลอนข้างของฟังก์ชัน $[\sin(f)] / f$

เพื่อลดผลจากการรัวไหลเราจำเป็นต้องใช้วิธีการตัดทอนโดเมนเวลาหรือฟังก์ชันถ่วงน้ำหนักที่มีลักษณะข้างในโดเมนความถี่เล็กกว่าฟังก์ชันสี่เหลี่ยมลอนข้างที่เล็กกว่านี้ทำให้การรัวไหลของผลลัพธ์ FFT มีค่าน้อยลง เพื่อให้เขาใจจุดนี้ได้ชัดเจนยิ่งขึ้น ขอให้พิจารณารูปที่ 6.2 อีกครั้งหนึ่ง ถ้าเปลี่ยนฟังก์ชันตัดทอนในรูปที่ 5.2(d) เป็นฟังก์ชันที่มีการตัดทอนลอนข้างต่ำ จะทำให้การประมาณค่าการแปลงของฟูเรียร์ดีขึ้น การใช้ฟังก์ชันถ่วงน้ำหนักของข้อมูลต้องทำกับจุดสุ่ม N ก่อนคำนวณ FFT





รูปที่ 6.8 ผลจาก FFT ของฟังก์ชันคาบเวลาโดยมีช่วงการตัดทอนไม่เท่ากับจำนวนเท่าของคาบเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

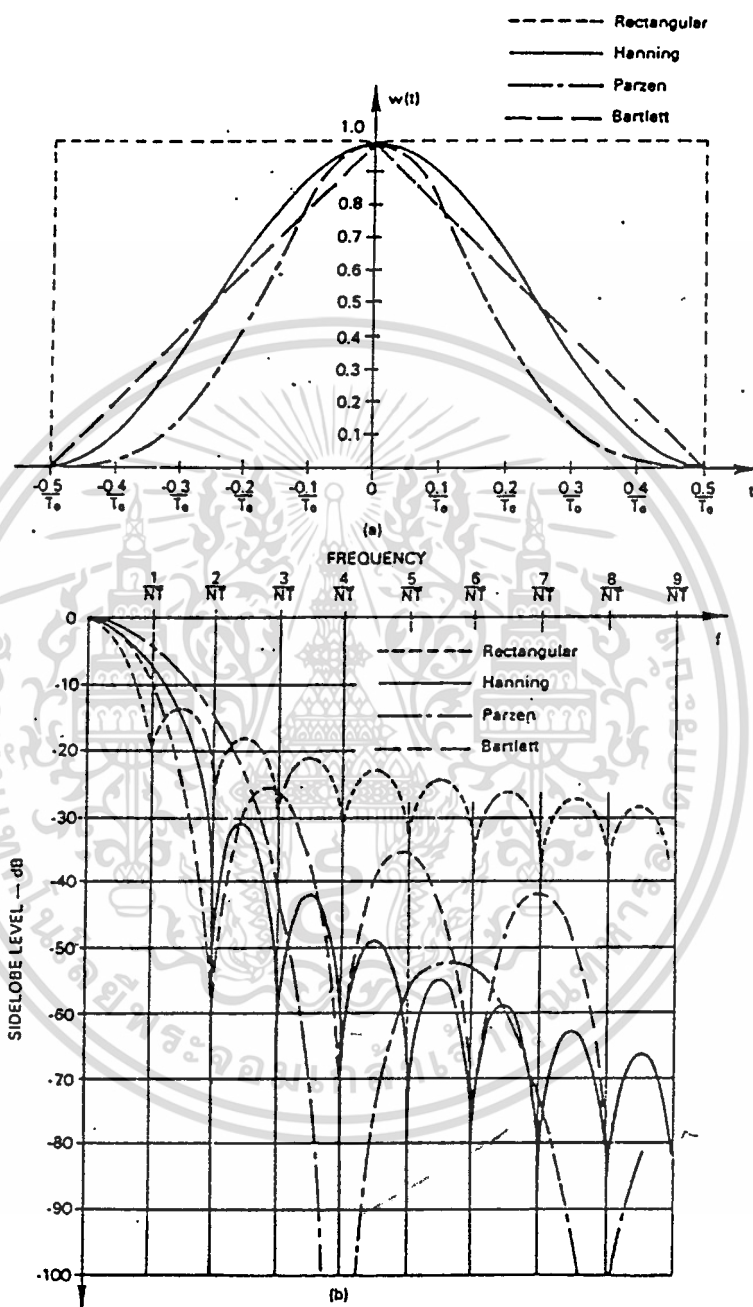
6.6.2 ลักษณะของฟังก์ชันถ่วงน้ำหนักชนิดต่าง ๆ

ในรูปที่ 5.9(a) ได้แสดงฟังก์ชันตัดทอนหรือถ่วงน้ำหนักที่นิยมใช้กับ FFT ผลของการตอบสนองของความถี่ของฟังก์ชันได้แสดงไว้ในรูปที่ 5.9(b) และในตารางที่ 5.2 แสดงข้อมูลลักษณะข้อเปรียบเทียบของฟังก์ชันถ่วงน้ำหนักแต่ละตัวทั้งในโดเมนเวลาและโดเมนความถี่โดยมีจุดศูนย์กลางอยู่ที่ศูนย์เพื่อง่ายต่อการสังเกต

จากรูปที่ 5.9(b) จะเห็นว่าฟังก์ชันถ่วงน้ำหนักมีลอนข้างในโดเมนความถี่ที่มีขนาดเล็กฟังก์ชันถ่วงน้ำหนักสี่เหลี่ยมทั้งนั้น ซึ่งจะทำให้ผลของการรั่วไหลลดน้อยลง แต่อย่างไรก็ตามฟังก์ชันถ่วงน้ำหนักทั้งหมดนี้มีลักษณะของลอนหลัก (main lobe) ที่กว้างกว่า ลองดูรูปที่ 6.4 (d) และ (e) อีกครั้ง ลองคิดเปรียบเทียบผลที่ได้จากการใช้ฟังก์ชันถ่วงน้ำหนักตามรูปที่ 5.9 (b) จะเห็นว่ายิ่งลอนหลักมีมากขึ้นเท่าใด ผลที่ได้จาก FFT ยิ่งมีความคลุมเคลือมากขึ้นเท่านั้น นั่นก็หมายความว่าลอนหลักมีความกว้าง จะทำให้แยกความแตกต่างยาก

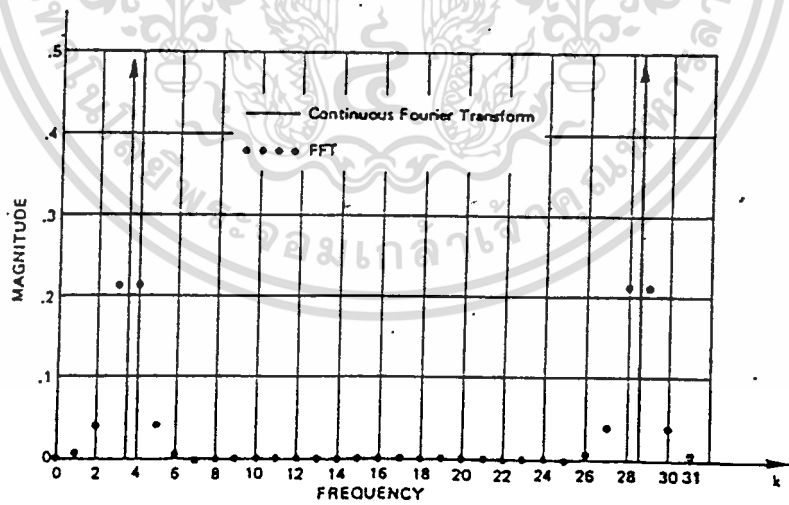
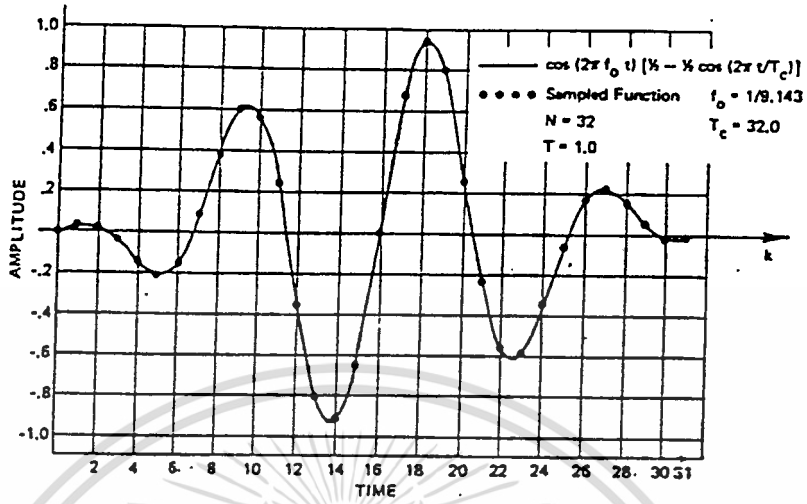
ข้อได้เปรียบเสียเปรียบระหว่างการรั่วไหล (ระดับการรั่วไหล) และความละเอียด (แบนด์วิดท์ของลอนหลัก) เป็นที่รู้จักกันดีในงานวิทยาศาสตร์สาขาต่างๆ ในตารางที่ 5.2 แสดงระดับสูง สุดของลอนข้าง และ แบนด์วิดท์ (bandwidth) 3-dB สำหรับแต่ละฟังก์ชันถ่วงน้ำหนักสำหรับงานการทดลองทั่วไป นิยมใช้ฟังก์ชันแฮนนิ่ง (Hanning function) เพราะโครงสร้างของมันง่ายกว่าแบบอื่น การเลือกใช้ฟังก์ชันถ่วงน้ำหนักที่ให้ผลดีที่สุดต้องขึ้นอยู่กับลักษณะของงานที่นำไปใช้งานด้วยเป็นสำคัญ

ผลที่ได้จาก FFT จะมีช่วงของความถี่ $f_0 = 1/NT$ ซึ่งอาจทำให้สับสนกับค่า $1/NT$ ของฟังก์ชันถ่วงน้ำหนักที่เลือกใช้ ผลของความละเอียดของผลลัพธ์ที่ได้



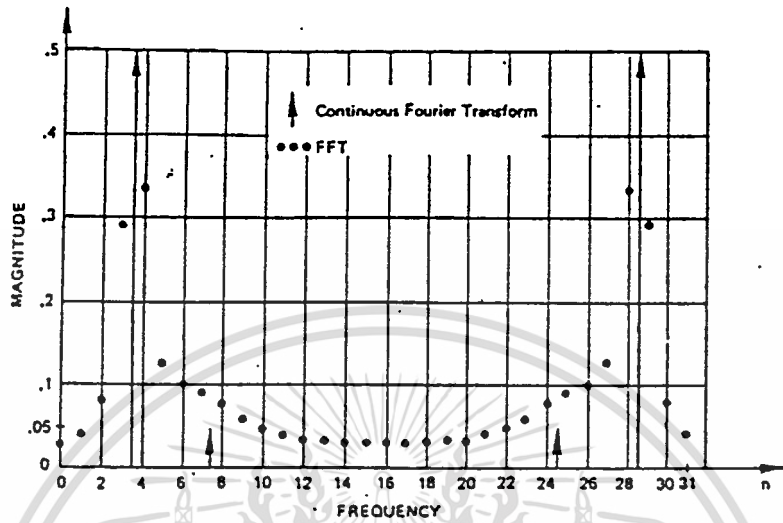
รูปที่ 6.9 แสดงฟังก์ชันถ่วงน้ำหนักหรือวินโดว์ของ FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

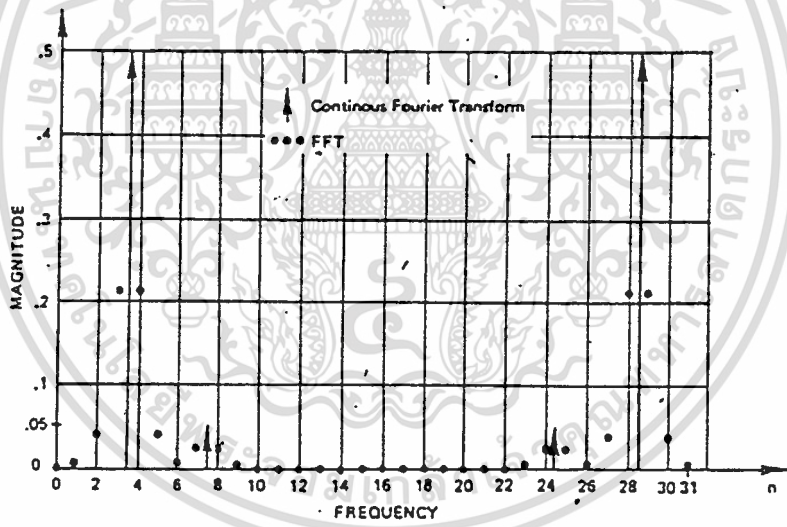


รูปที่ 6.10 แสดงตัวอย่างการใช้แฮนนิ่งฟังก์ชันเพื่อลดส่วนรั่วไหลของการคำนวณด้วย FFT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ 6.11 (a) แสดงตัวอย่างที่เลือนลางเนื่องจากส่วนที่รั่วไหลของลอนข้าง
 (b) สัญญาณที่ตรวจจับได้หลังจากที่ผ่านแชนเนลฟังก์ชันแล้ว

จาก FFT เป็นฟังก์ชันของแบนวิดธ์ของฟังก์ชันถ่วงน้ำหนักแต่ละตัวเท่านั้น (ดูรูปที่ 5.9(b)) ดังนั้นการใช้ค่านิยามของความละเอียด $1/NT$ ต้องทำอย่างระมัดระวังและรำลึกอยู่เสมอว่า มันเป็นช่องว่างระหว่างความถี่ที่เกิดจากผลที่ได้จาก FFT เท่านั้นและไม่เกี่ยวกับ $1/NT$ ของวินโดว์ที่ใช้

ต่อไป เราจะแสดงผลที่ได้จากฟังก์ชันถ่วงน้ำหนักที่มีลอนข้างต่ำเพื่อให้เห็นลักษณะการนำไปใช้ลดการรั่วไหลของผลการตัดทอนในโดเมนเวลาในรูปที่ 5.10(a) เราได้แสดงรูปที่ 9.7 (a) ที่คูณด้วยฟังก์ชันถ่วงน้ำหนักแชนนิงที่แสดงไว้ในรูปที่ 5.9(a) แล้ว

รูปที่ 5.10(b) แสดงสุมจาก FFT ของรูปที่ 5.10(a) จากรูปจะเห็นว่าส่วนรั่วไหลลดลงไปเป็นอย่างมาก แต่องค์ประกอบหลักของความถี่ มีระยะกว้างขึ้นหรือคลุมเคลือเมื่อเปรียบเทียบกับอิมพัลส์ฟังก์ชันเพราะผลจากการคอนโวลูชันความถี่ของอิมพัลส์ฟังก์ชันกับผลการแปลงฟูเรียร์ของฟังก์ชันถ่วงน้ำหนัก

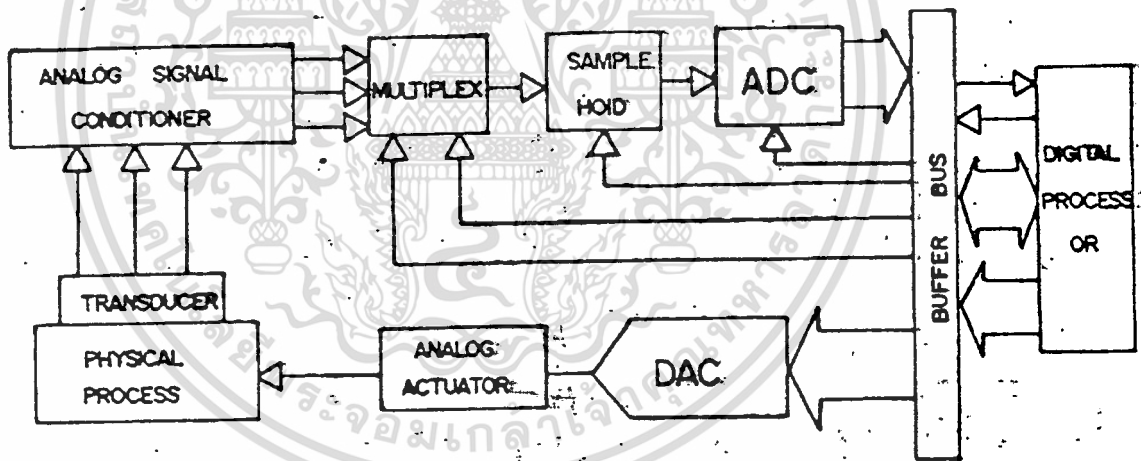


Weighting Function Nomenclature	Time Domain	Frequency Domain	Highest Side-Lobe Level (db)	3-dB Bandwidth	Asymptotic Roll-off (dB/Octave)
Keelangular	$w_n(t) = 1 \quad t \leq \frac{T_0}{2}$ $= 0 \quad t > \frac{T_0}{2}$	$W_n(f) = \frac{T_0 \sin(\pi f T_0 / 2)}{\pi f T_0}$	-13	$\frac{0.85}{T_0}$	4
Bartlett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

ตาราง พิกัดขั้ววงนํ้าหนักข้อยูล

ทฤษฎีของ Data Acquisition and Conversion

รูปแบบสัญญาณไฟฟ้าที่เราพบเห็นและคุ้นเคยในชีวิตประจำวันจะอยู่ในรูปแบบของสัญญาณที่ต่อเนื่อง หรือเรียกว่าสัญญาณอนาลอก ซึ่งแต่เดิมการนำเอาสัญญาณไฟฟ้าดังกล่าวมาประมวล (Processed) ใช้งานในรูปแบบที่มีประโยชน์จะกระทำในแบบอนาลอก แต่เมื่อเทคนิคการประมวลสัญญาณ (conversion) จึงได้มีความจำเป็นขึ้น จากสัญญาณอนาลอกที่มีตามธรรมชาติถูกเปลี่ยนมาเป็นดิจิทัลโดย Analog to digital converters (ADC) และประมวลโดยตัวประมวลทางดิจิทัล (Digital Processor) เช่น คอมพิวเตอร์ จากนั้นจะถูกนำมาแสดงผลเลย หรือถูกเปลี่ยนกลับมาอยู่ในรูปแบบอนาลอกที่คุ้นเคยและใช้งานได้ง่ายกว่าโดย Digital Analog Convertors (DAC) ในรูปที่ 1 แสดงระบบควบคุมโดยการประมวลผลข้อมูลในระบบดิจิทัล



รูปที่ 1 ระบบควบคุมที่มีการประมวลผลข้อมูลทางดิจิทัล

การเปลี่ยนแปลงทางกายภาพในลักษณะต่างๆก็ตาม (Physical process เช่นความดันอุณหภูมิ ฯลฯ) จะถูกเปลี่ยนให้เป็นสัญญาณไฟฟ้าอนาลอก โดยทรานสดิวเซอร์ที่มี คุณสมบัติที่เหมาะสมกับรูปแบบทางกายภาพนั้น สัญญาณไฟฟ้าจะถูกป้อนให้อยู่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

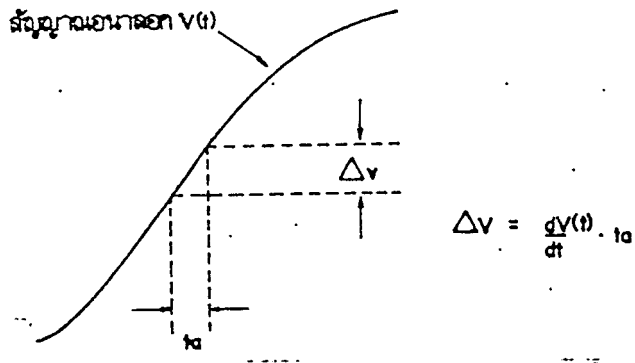
ในรูปและขนาดที่ที่เหมาะสมก่อนโดย Analog Signal conditioner เช่น วงจรขยายหรือ ฟิลเตอร์ เป็นต้น

ในระบบที่ข้อมูลที่ต้องประมวลในเวลาเดียวกันมีหลายข้อมูล แต่เนื่องจาก ADC หลาย ๆ ตัวทำงานแยกกัน แต่จะใช้วิธีแบ่งเวลา (Timesharing) โดยวิธี (Multiplexing) วงจร Sample and hold (S/H) จะสุ่ม Sample ขนาดของสัญญาณอนาลอกมาและ Hold ไว้ชั่วขณะเพื่อรอให้ ADC รับประทานไปเปลี่ยนให้เป็นกระแสดิจิทัลจนเรียบร้อยแล้วค่อยรับสัญญาณทั้งหมดนี้เพื่อไม่จำเป็นต้องใช้ ADC ที่ต้องทำงานได้เร็วมากนัก ข้อมูลดิจิทัลจะถูกส่งต่อไปยัง system bus ท้าการประมวลโดย Processor แล้วเปลี่ยนกลับมาเป็นสัญญาณอนาลอกโดย DAC เพื่อควบคุมกิจกรรมทางกายภาพของระบบผ่าน Analog actuator

ทฤษฎีการ Sampling

ในการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลนั้น ADC จะต้องใช้เวลาช่วงหนึ่งในการจัดการ ซึ่งช่วงเวลาดังกล่าวนั้นขึ้นอยู่กับหลายแพคเตอร์ เช่น ความละเอียดของการเปลี่ยนสัญญาณ เทคนิคของการเปลี่ยนแปลงสัญญาณและความเร็วในการทำงานของอุปกรณ์ร่วมอื่นๆ ความเร็วของการแปลงสัญญาณนี้ จำเป็นของการประยุกต์ใช้งาน เฉพาะอย่าง และความแม่นยำที่ต้องการ

Aperture time : ช่วงเวลาในการแปลงสัญญาณบางครั้งอาจเรียกว่า Aperture time ซึ่งคำว่า Aperture time โดยทั่วไปหมายถึงช่วงเวลาที่เกิดความไม่แน่นอนในการวัด และผลก็คือเกิด Error ต่อค่าที่วัดได้



ในรูปที่ 2 สัญญาณอนาล็อก $v(t)$ มีอัตราการเปลี่ยนแปลง dv/dt ในช่วง Aperture time t_a ดังนั้นช่วงการเปลี่ยนแปลงอนาล็อกจะเท่ากับ V โดย

$$\Delta V = \frac{t_a \cdot dv(t)}{dt}$$

ดังนั้นหากเวลาที่ ADC ใช้ในการเปลี่ยนสัญญาณในเวลา t_a นี้ รหัสดิจิทัลที่ได้ อาจจะไม่ตรงกับขนาดของสัญญาณอนาล็อก ค่าวัดค่าหนึ่งในช่วงนี้ และส่วนอื่นๆที่เหลือคือ Error ที่เกิดขึ้นซึ่งบางครั้งเป็นบางครั้งที่รหัสดิจิทัลจะตรงกับค่าอนาล็อกที่ถูกต้อง ตัวอย่างในกรณีสัญญาณอินพุตเป็นรูปไซน์ อัตราการเปลี่ยนแปลงบนรูปคลื่นจะเกิดที่จุดตรงบริเวณจุดตัดแกนเวลารอบๆ จุดศูนย์โวลท์ (Zero crossing) ที่เกิดขึ้น และ Aperture time error คือ

$$\begin{aligned} \Delta V &= t_a \frac{d}{dt} (A \sin \omega t) = 0 \\ &= t_a A \omega \end{aligned}$$

และ error รวมคิดจากอัตราส่วนที่คิดเต็มสเกล คือ

$$\Sigma = \frac{\Delta V}{1A} = \pi f t_a$$

ดังนั้นหากต้องการเปลี่ยนสัญญาณเป็นรูปไซน์ความถี่ 1 KHz ให้เป็นสัญญาณดิจิทัล 10 บิต ซึ่งยอมให้ error ไม่เกินกว่า resolution คือ $1/2^{10}$ LSB หรือ 0.001 ดังนั้นเวลา Aperture time จะต้องอยู่ในช่วง

$$t_a = \frac{\Sigma}{\pi f} = \frac{0.001}{\pi * 10^3} = 310 * 10^{-9} \text{ Sec}$$

จะเห็นว่าแม้สัญญาณ 1 KHz จะไม่ซับซ้อนสูงก็จริงแต่ ADC ที่ใช้ต้องการเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาในการเปลี่ยนนาน 230 นาโนวินาทีเป็นรหัส 10 บิต นั้นไม่สะดวกง่าย ๆ ซึ่งวิธี
ที่ใหม่จำเป็นต้องใช้ ADC ความเร็วสูงเช่นนั้นคือ การใช้ Sample and hold ซึ่ง
ที่มี Aperture time น้อยๆนั้นทาง่ายและราคาถูกลงกว่า

Sample and hold Aperture error

วงจร Sample and hold จะทำการสุ่มสัญญาณอินพุต และนำสัญญาณนั้นมา
เก็บหรือ hold ไว้ในระยะเวลาหนึ่ง ซึ่งส่วนใหญ่จะใช้การประจุแรงดันนั้นในตัวเก็บ
ประจุที่รั่วไหล ดังนั้นเมื่อแรงดันอินพุตสามารถคงอยู่ได้ดังนั้น Aperture time ของ

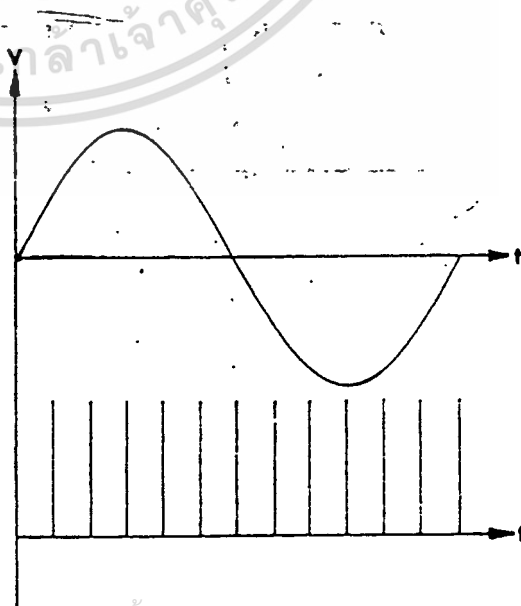
Sample & hold คือเวลาตั้งแต่เริ่มสุ่มสัญญาณจนเก็บประจุค่าแรงดันจนถึงค่าที่สุ่ม ซึ่ง
สำหรับ Sample & hold แล้ว Aperture time ขึ้นอยู่กับแบนด์วิธ และ Switch
ing time ของอุปกรณ์แอกทีฟ ที่ใช้ในวงจรซึ่งหาและสร้างได้ง่ายและราคาถูกลงกว่า
การสร้าง ADC ความเร็วสูง

ระบบการ Sample ข้อมูลและทฤษฎีการ Sample

ในระบบการ Sampling สัญญาณอนาลอกจะถูกสุ่มเป็นระยะๆ คงที่ตามรูปที่
3 กลุ่มของสัญญาณสุ่มจะแบนด์วิธที่ทำงานด้วยความเร็วสูง ซึ่งจะทำการตัดต่อสัญญาณ
อนาลอกในช่วงเวลาอันสั้น

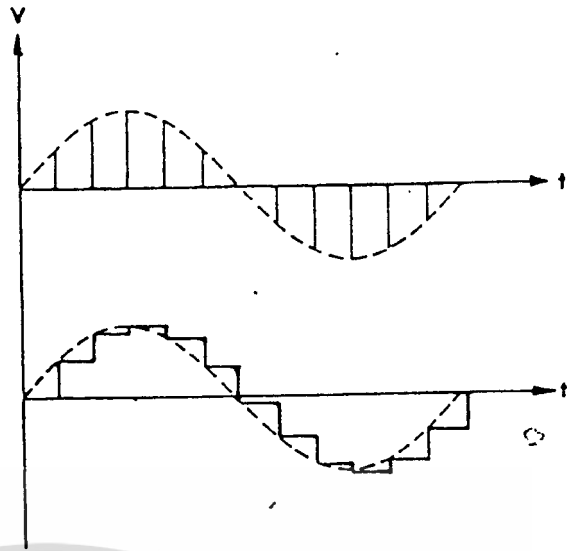
ก. สัญญาณ

ข. พัลส์ sampling



ค. สัญญาณที่ถูก sampling.

ง. สัญญาณที่ถูก Hold



รูปที่ 3 การ Sampling สัญญาณ

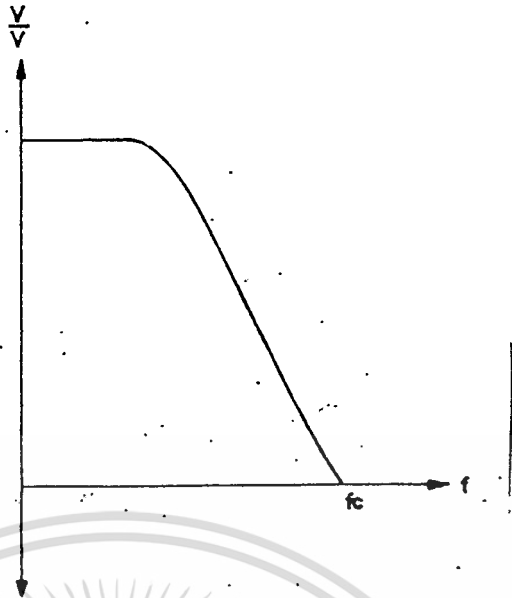
ผลของการสุ่มสัญญาณด้วยความเร็ว จะเสมือนกับการคูณขบวนสัญญาณพัลส์แคบๆ กับสัญญาณอนาล็อก ซึ่งจะได้เป็นสัญญาณที่มอดูเลตระหว่างขบวนพัลส์กับสัญญาณอนาล็อก ดังแสดงในรูป โดยสัญญาณอนาล็อกจะเข้ามาบนขบวนพัลส์ ถ้าหากเอาสวิทช์และตัวเก็บประจุ แทนสวิทช์แล้วสัญญาณอนาล็อกที่ถูกสุ่มจะถูก hold ไว้ในตัวเก็บประจุจนกว่าสัญญาณค่าใหม่จะถูกสุ่มเข้ามา ซึ่งลักษณะของเข้าที่พูดที่ได้แสดงในรูป ๓

มีปัญหาที่ว่า การสุ่มสัญญาณนั้นควรมีขนาดเท่าใด จึงจะไม่ทำให้ข้อมูลนั้นเสียหาย เมื่อสัญญาณนั้นถูกเปลี่ยนกลับมาเป็นเช่นเดิม คำตอบก็คือ ขึ้นอยู่กับความถี่ของสัญญาณอนาล็อก และทฤษฎีของการสุ่มกล่าวไว้ว่า "ถ้าสัญญาณต่อเนื่องที่มีความถี่และฮาร์มอนิกไม่เกิน f_c แล้วสัญญาณดังกล่าว จะสามารถเปลี่ยนกลับมาได้ดังเดิมโดยไม่สูญเสียรายละเอียดหรือผิดเพี้ยนไป ถ้าอัตราการสุ่มไม่น้อยกว่า $2f_c$ ต่อวินาที"

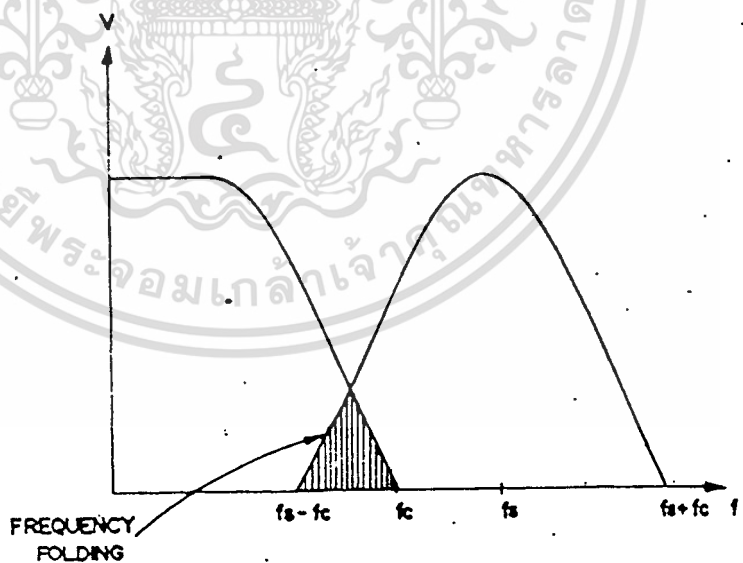
Frequency folding and Aliasing

จากทฤษฎีของการสุ่มสามารถอธิบายด้วยลักษณะรูปสเปกตรัมของสัญญาณในรูปที่ 4 รูป 4 ก แสดงให้เห็นสเปกตรัมของสัญญาณที่ถูกสุ่มซึ่งแบนด์วิธไม่เกิน f_c ในขณะที่สัญญาณนี้จะถูกสุ่มด้วยความถี่ f_s ขบวนการมอดูเลชันจะทำให้แถบสเปกตรัมของสัญญาณสุ่มจะขยายกว้างออกจาก f_s เป็น $2f_s$ $3f_s$ ได้เป็นดังรูปที่ 4 ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4 ก แสดงสเปกตรัมของสัญญาณอนาล็อกที่ถูกสุ่ม



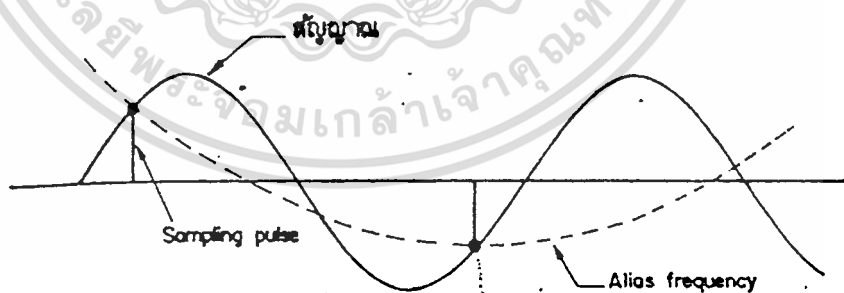
รูปที่ 4 ข หลังจากการสุ่มเกิด Frequency folding

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าความถี่ของสัญญาณสุ่ม f_s ไม่สูงพอหลังจากการสุ่มสเปกตรัมบางส่วนของ f_s จะหาซ็อนกลับสเปกตรัมของสัญญาณซึ่งเรียกว่า Frequency Folding หากเป็นเช่นนี้ก็จะทำให้เกิดความเพี้ยนแก่สัญญาณนอก จากการซ็อนกลับของสเปกตรัม เมื่อสัญญาณได้เปลี่ยนให้กลับป้อนรูปรเดิม และถ้าเลื่อนความถี่ของการสุ่มให้สูงขึ้น จนรอการซ็อนกลับของสเปกตรัมหมดไป ($f_s - f_c$) $> f_c$ และการเปลี่ยนกลับของสัญญาณหลังจากถูกสุ่มยังคงเหมือนเดิมได้

จากที่กล่าวมาแสดงการสนับสนุนทฤษฎีการสุ่มที่ว่าให้ $f_s > 2f_c$ นั้นคือการกำจัดซ็อนกลับของสเปกตรัมได้สองวิธีคือ หนึ่งด้วยการใช้อัตราสุ่มที่สูงพอ และสอง การทำการ filter ความถี่ของสัญญาณนอกก่อนการสุ่ม เพื่อให้แบนด์วิธไม่สูงไปกว่า $f_s/2$ ในทางปฏิบัติแล้วยังคงเกิด Frequency folding ได้เสมอจากส่วนฮาร์โมนิกของสัญญาณสเปกตรัม ของสัญญาณรบกวนที่ยังคงอยู่แม้ว่าจะทำการฟิลเตอร์ก่อนหน้ามาแล้วก็ตาม การกำจัดสเปกตรัมดังนี้วิธีที่ได้ผล คือ พยายามให้การสุ่มตัวอย่างเป็นไปอย่างรวดเร็วมากที่สุด ซึ่งตามปกติจะสูงกว่าความถี่ต่ำสุดตามทฤษฎี Sampling คือ $2f_c$ เสมอ

ผลของการใช้อัตราสุ่มที่ไม่เหมาะสมอีกประการหนึ่ง ที่ทำให้เกิดสัญญาณรูปซายน์ แสดงในรูปที่ 5 เรียกว่า alias frequency ซึ่งเกิดกับสัญญาณที่เปลี่ยนกลับมาเช่นเดิมหลังจากถูกสุ่มแล้ว



รูปที่ 5 การเกิด alias frequency จากการสุ่มความถี่ต่ำกว่า 2 เท่าของความถี่สัญญาณอนาลอก

ในกรณีการสุ่มนี้ในอัตราที่ต่ำกว่า 2 เท่าต่อรอบสัญญาณอินพุทจะทำให้ได้สัญญาณชายน้ความถี่ต่ำ แสดงเป็นจุดไขปลาดังรูปที่ 5 ซึ่งจะเห็นว่าความถี่ alias อาจ จะแตกต่างจากความถี่เดิมไปมาก



สัญญาณต่างๆบนสล็อตของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรอินเทอร์เฟซเข้าไปในภายหลังได้โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ด (Main Board) สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต (สำหรับใน IBMPC/XT) จะมี 8 สล็อต; จะกล่าวถึงภายหลัง) ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62ขา แบ่งออกเป็น 2 ข้างๆละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อตโดยขาที่อยู่ทางด้านซ้ายของสล็อตจะเรียกรโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขา เช่น ขา B16 ก็คือขาทางด้านซ้ายของขา สล็อตที่ 16 (นับจากทางด้านท้ายของเครื่อง) ส่วนขาที่อยู่ทางด้านขวาของสล็อตจะเรียกรโดยใช้อักษร "A" นำหน้าเลขตำแหน่งของขา เช่น ขา A11 ก็คือขาทางด้านขวาของสล็อตขาที่ 11 (นับจากทางด้านซ้ายของเครื่อง)

แต่ละขาของสล็อตเหล่านี้จะเชื่อมต่อกับสัญญาณต่างๆบนเมนบอร์ด ทำให้การสร้างวงจรอินเทอร์เฟซกับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus), บัสข้อมูล (Data Bus), บัสสำหรับการควบคุมการเขียน/อ่านข้อมูลจากหน่วยความจำ หรือพอร์ท I/O, เส้นสัญญาณสำหรับการขออินเทอร์รัพท์ของวงจรอินเทอร์เฟซ, เส้นสัญญาณสำหรับการขอ DMA, สัญญาณฐานเวลา (Timing Signal) ต่างๆที่ใช้ในระบบ, เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHECK)

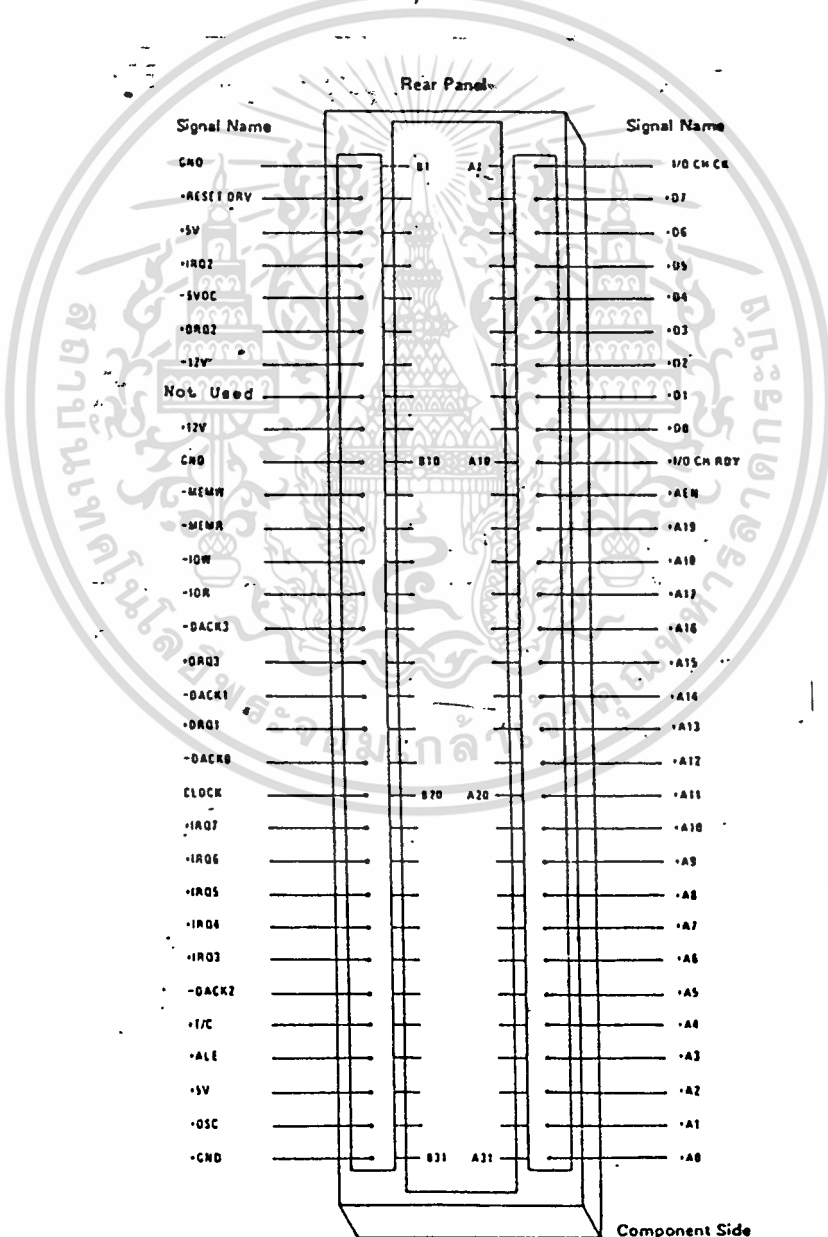
นอกจากเส้นสัญญาณเหล่านี้แล้ว สล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่างๆที่ใช้ในระบบอีกด้วย คือ +5Vdc, -5Vdc, +12Vdc, -12Vdc

รายละเอียดเกี่ยวกับสัญญาณต่างๆ

(OSC: Oscillator; ขา B30)

ขานี้เป็นเอาต์พุตที่เชื่อมต่อกับสัญญาณคล็อกที่มีค่าความถี่สูงสุดบนเมนบอร์ดคือ 14.31616 MHz ซึ่งมีคาบเวลาประมาณ 70 nanosec. และมี Duty Cycle (ช่วงเวลาใน 1 คาบที่สัญญาณคล็อกมีลอจิกเป็น "1" หารด้วยคาบเวลาทั้งหมด) ประมาณ 50% สัญญาณคล็อกอื่นๆของระบบ เช่น คล็อกที่ป้อนให้กับ 8088 หรือ ชิพ

ซัพพอร์ทที่ต่าง ๆ นั้นจะถูกสร้างขึ้นโดยการผลิตสัญญาณคล็อกนี้ อย่างไรก็ตามสิ่งหนึ่งที่ต้องคำนึงถึงในการใช้งานสัญญาณ OSC ก็คือ สัญญาณนี้จะไม่ Synchronize กับสัญญาณอื่น ๆ บนบอร์ดของระบบ ดังนั้น จึงไม่ควรที่จะนำสัญญาณจากขา OSC นี้ไปใช้เป็นสัญญาณคล็อกสำหรับวงจรภายนอกอื่น ๆ ที่ทำงานร่วมกับระบบ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLK(Clock: ขาB20):

ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งต่อกับสัญญาณคล็อกที่ถูกสร้างขึ้นโดยการหารสัญญาณ OSC ด้วย 3 ทำให้ได้ความถี่ประมาณ 4.77 MHz (14.31818 MHz/3) หรือมีช่วงเวลาใน 1 คาบ (ช่วงเวลาของคล็อก 1 ลูก) เท่ากับ 210 nanosec. (1/4.77 MHz) สำหรับค่า Duty Cycle ของสัญญาณนี้จะมีค่าประมาณ 1/3 คือใน 1 คาบจะมีช่วงเวลาที่เป็นลอจิก "1" เท่ากับ 1/3 ของคาบเวลาทั้งหมด หรือประมาณ 70 nanosec. และช่วงเวลาที่เป็นลอจิก "0" เท่ากับ 2/3 ของคาบเวลาทั้งหมด หรือประมาณ 140 nanosec. สัญญาณนี้เป็นสัญญาณที่ถูกใช้ เป็นคล็อกของระบบ

RESET DRV (ขา B2)

ขาสัญญาณนี้เป็นเอาต์พุต ซึ่งจะแอกทีฟ (logic "1") ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอกทีฟ ไปจนกว่าระบบต่าง ๆ ภายใน IBM/PC จะพร้อมที่จะทำงานได้ จากนั้นสัญญาณนี้จะเปลี่ยนกลับเป็นลอจิก "0" นอกจากนี้ในระหว่างการทำงานของ IBM/PC ถ้าระดับแรงดันของแหล่งจ่ายไฟตกลง สัญญาณนี้ก็จะถูกทำให้แอกทีฟเช่นกัน ภายที่ดับแล้วสัญญาณนี้จะถูกนำไปใช้ในการรีเซ็ตวงจรอินเทอร์เฟซหรืออุปกรณ์ I/O ต่าง ๆ ในช่วงที่เริ่มจ่ายไฟให้กับระบบ ซึ่งจะเป็นการทำให้วงจรหรืออุปกรณ์เหล่านี้ถูกปรับให้อยู่ในสภาวะที่แน่นอน ก่อนที่จะเริ่มดำเนินการ

ทำงานในระบบ (สถานะนี้เป็นสถานะที่เราทราบ และต้องการให้วงจรทำงานในขณะที่ระบบถูกรีเซ็ต)

A0-A19 (Address Bus; ขา A31-A12)

ขาสัญญาณทั้ง 20 ขานี้เป็นเอาต์พุต ซึ่งใช้สำหรับกำหนดแอสแตรของหน่วยความจำ หรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อกัน โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (Least Significant Bit) และ A19จะมีนัยสำคัญสูงสุด (MOST SIGNIFICANT BIT) สำหรับค่าแอสแตรระบบบัสแอสแตร A0-A19 นี้จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน/เขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA-CONTROLLER จะเป็นผู้กำหนดค่าแอสแตรระบบบัสแอสแตรเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ)

จะเห็นว่าจำนวนเส้นแอสแตรนี้จะมีอยู่ 20 เส้นซึ่งสามารถที่จะอ้างแอสแตรของหน่วยความจำได้ถึง Mbyte แต่อย่างไรก็ตามจะมีแอสแตรบางแอสแตรที่ถูกใช้งานโดย IBM/PC อยู่ก่อนแล้ว คือแอสแตรของหน่วยความจำ RAM โดยเมนบอร์ดที่ถูกใช้โดยระบบจำนวน 64 Kbyte (สำหรับ IBM PC/XT จะเป็นจำนวน 256Kbyte) และแอสแตรสำหรับหน่วยความจำ ROM อีก 48 Kbyte ซึ่งจะถูกจัดในช่องแอสแตรบนสุด 1Mbyte คือ OFCOOH จนถึง OFFFFFH (สำหรับ IBM PC/XT จะเป็น 64 Kbyte)

สำหรับการอ้างแอสแตรของพอร์ท I/O นั้นจะใช้เส้นแอสแตรเพียง 16 เส้น คือ A0-A15 ซึ่งจะทำการอ้างแอสแตรของพอร์ทได้ 64K พอร์ทโดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอสแตรที่เหลือคือ A16-A19 นั้นจะไม่ถูกใช้งานอย่างไรก็ตาม IBM PC จะใช้เส้นแอสแตรในการอ้างแอสแตรของพอร์ทเพียง 10 เส้น คือจาก A0-A9 และค่าแอสแตรที่ใช้งานต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น D0-D7 (Data Bus; ขา A9-A2) :

ขาสัญญาณนี้เป็นแบบ Bi-directional ซึ่งต่อกับบัสข้อมูลของระบบเพื่อทำหน้าที่ในการส่งผ่านข้อมูลระหว่างพอร์ท I/O กับ IBM/PC โดยบิต D0 จะมีนัยสำคัญต่ำสุด และบิต D7 จะมีนัยสำคัญสูงสุด

สำหรับในบัสไซเคิลของการเขียนข้อมูลที่สร้างขึ้นโดย 8088 นั้นข้อมูลจะส่งออกมาที่บัสข้อมูล ก่อนที่สัญญาณ \overline{IOW} (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ท) หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

\overline{MEMW} (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ท) หรือ \overline{MEMW} (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็น "1" (ขอขาขึ้น) ซึ่งโดยทั่วไปขอขาขึ้นของสัญญาณ \overline{IOW} หรือ \overline{MEMW} นี้จะถูกใช้เพื่อสั่งให้พอร์ท I/O หรือหน่วยความจำที่มีแอสแตรอสค่าแอสแตรอสบนบัสแอสแตรอสนั้นรับข้อมูลไปเก็บไว้

สำหรับบัสสาขาเกิดของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ท I/O หรือ หน่วยความจำที่ถูกอ้างถึง จะต้องส่งข้อมูล ก่อนที่สัญญาณ \overline{IOR} (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ท) หรือ \overline{MEMR} (ในกรณีที่ต้องการอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอขาขึ้น)

ALE (Address Latch Enable; ขา B28) :

ขานี้เป็นสัญญาณเข้าที่พอร์ทที่ 8288 Bus-controller สร้างขึ้นเพื่อใช้สำหรับ แสดงการเริ่มต้นของบัสสาขาเกิด และแสดงให้อุปกรณ์ภายนอกทราบว่าแอสแตรอสที่ 8088 ต้องการจะติดต่อด่วนนั้นถูกส่งออกมาบนบัสแอสแตรอสแล้ว โดยที่สัญญาณ ALE นี้จะเปลี่ยนจากลอจิก "1" เป็นลอจิก "0" เมื่อค่าแอสแตรอสที่ถูกต้อง ถูกส่งออกมาบนบัสข้อมูลเรียบร้อยแล้ว ดังนั้นขอขาลงของสัญญาณ ALE นี้จะถูกใช้ในการแลทช์ค่าแอสแตรอสจากบัสแอสแตรอส/ข้อมูล (Address/Data Bus; AD0-AD7) ออกจากกัน อย่างไรก็ตามสัญญาณ ALE จะแอกทีฟเฉพาะในบัสสาขาเกิดที่สร้างขึ้นโดย 8088 เท่านั้น โดยจะไม่แอกทีฟในระหว่างขบวนการ DMA

$\overline{I/O\ CHCK}$ (I/O Channel Check; ขา A1) :

ขาอินพุตนี้เป็นสัญญาณที่ใช้ในการแสดงความคิดพลาดเกี่ยวกับพาริตี้ ที่เกิดขึ้นในการทำงานของวงจรรีเฟรชหรืออุปกรณ์ I/O เมื่อขาสัญญาณนี้ได้รับลอจิก "0" จะทำให้ 8088 ถูกอินเทอร์รัพแบบ Non-Maskable (NMI) อย่างไรก็ตามเราสามารถที่จะกำหนดให้วงจรรายานของ IBM/PC ทำการขออินเทอร์รัพ (เมื่อได้รับสัญญาณ $\overline{I/O\ CHCK}$) หรือไม่ก็ได้ โดยการกำหนดลอจิกของข้อมูล ของพอร์ทที่ควบคุมการอินเทอร์รัพแบบ NMI คือบิตที่ D7 ของพอร์ท 00AAH ในกรณีที่ D7 ของพอร์ทที่ 00A0H ถูกเซตเป็น "1" ก็จะทำให้วงจรรายานนอกขออินเทอร์รัพแบบ NMI ได้ (Enable) แต่ถ้าบิต D7 ของพอร์ท 00A0H ถูกเซตเป็น "0" ก็จะเป็นการดิสเอเบิล (Disable) การขออินเทอร์รัพแบบ NMI ดังนี้

Enable : ใช้คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

Disable : ใช้คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

และเนื่องจากยังมีอุปกรณ์อื่นที่สามารถขออินเทอร์รัพแบบ NMI ได้อีกดังนั้นซอฟต์แวร์ที่ทำงานจะต้องสามารถตรวจสอบว่า การขออินเทอร์รัพนั้นเกิดขึ้นจากแหล่งใดได้ด้วย

I/O CHRDY (I/O Channel Ready; ขา A10) :

ขาสัญญาณนี้เป็นอินพุตที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อยู่ I/O หรือ หน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น ไม่สามารถทำงานทันในช่วงเวลาปกติของบัสไซเคิลนั้นๆได้ (ช่วงเวลาของบัสไซเคิลที่เกี่ยวข้องกับหน่วยความจำใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 4 ลูก หรือ 840 nanosec. ในขณะที่บัสไซเคิลที่เกี่ยวข้องกับ I/O จะใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 5 ลูก หรือ 1.05 usec)

เมื่ออุปกรณ์ I/O หรือหน่วยความจำต้องการที่จะเพิ่มช่วงเวลาในบัสไซเคิลให้ยาวนานขึ้นอีกนั้น จะสามารถทำได้โดยการป้อนลอจิก "0" ให้กับขา I/O CHRDY ในช่วงเวลาที่ I/O หรือ หน่วยความจำที่กำหนดนั้น ได้รับสัญญาณจากการตีรัทแอสเตอร์และสัญญาณ $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ หรือ $\overline{\text{IOW}}$ แอคทีฟ สำหรับรายละเอียดเกี่ยวกับการเพิ่มช่วงเวลาในบัสไซเคิลนั้นจะกล่าวถึงอีกครั้งในบทที่ 10 "การสร้าง Wait State "

IRQ2-IRQ7 (Interrupt Request 2 Through 7; ขา B4 และ B25-B21) :

ขาสัญญาณทั้ง 6 นี้เป็นขาอินพุตที่ใช้สำหรับการขออินเทอร์รัพจาก 8088 โดยสัญญาณเหล่านี้จะต่อเข้ากับ 8259A บนเมนบอร์ดโดยตรง โปรแกรมในส่วนของ BIOS ของ IBM/PC จะทำการโปรแกรม 8259A ให้ IRQ2 มีลำดับความสำคัญสูงสุด (Highest priority) และ IRQ7 มีลำดับความสำคัญต่ำสุด ในกรณีที่มีการขออินเทอร์รัพเกิดขึ้นคือ ระดับลอจิกที่ขา IRQ ขาดขาหนึ่งจากลอจิก "0" เป็นลอจิก "1" (ขอขาขึ้น) 8259A ก็จะทำการส่งสัญญาณ INT ให้กับ 8088 เพื่อทำการขออินเทอร์รัพ

สิ่งสำคัญในการขออินเทอร์รัพผ่านทาง IRQ2-IRQ7 นี้ ก็คืออุปกรณ์ที่ทำการขออินเทอร์รัพผ่านทาง IRQ ขาดก็จะต้องรักษาระดับสัญญาณที่ขา IRQ นั้น ให้แอคทีฟ (ลอจิก "1") อยู่จนกว่าจะได้รับสัญญาณ INTA (Interrupt Service Routine) จาก 8088 เสียก่อน ถ้าไม่เช่นนั้นการขออินเทอร์รัพจะถูกยกเลิก และอินเทอร์รัพ Level 7 (IRQ7) ก็จะถูกสร้างขึ้นโดยอัตโนมัติ ไม่ว่าการขออินเทอร์รัพที่ถูกเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMW (ขา B11)

ขานี้เป็นเข้าที่พหุแอกทีฟที่ลอจิก "0" ซึ่ง 8238 Bus-Controller สร้างขึ้นในานระหว่างบัสฯเคิล ในการผลิตข้อมูลลงในหน่วยความจำของ 8088 สัญญาณ MEMW นี้จะถูกส่งออกมาเพื่อให้หน่วยความจำที่แอสแตรตรงกับค่าแอสแตรบนบัสแอสแตรสนั้น หากการรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้ อดยทั่วไปหน่วยความจำจะรับข้อมูลในช่วงขอบขาขึ้นของสัญญาณ MEMW

สำหรับในขบวนการระหว่าง DMA นั้น 8237A-5 DMA Controller จะทำการควบคุมบัสต่างๆของระบบแทน 8088 และสัญญาณ MEMW จะถูกใช้าในบัสฯเคิลของการเขียนข้อมูลลงในหน่วยความจำ (ข้อมูลถูกส่งจากอุปกรณ์ I/O ไปให้กับหน่วยความจำ)

MEMR (Memory Read; ขา B12) :

ขานี้เป็นเข้าที่พหุจาก 8238 ซึ่งสัญญาณนี้จะแอกทีฟ(ลอจิก "0") ในระหว่างบัสฯเคิลของการอ่านข้อมูล จากหน่วยความจำของ 8088 เพื่อให้หน่วยความจำที่มีแอสแตรตรงกับค่าแอสแตรบนบัสแอสแตรสนั้น หากการส่งข้อมูลลงมาจากบัสข้อมูล อดยหน่วยความจำนั้นจะต้องส่งข้อมูลออกมาในช่วง 30 nanosec ก่อนที่สัญญาณ MEMW จะกลับเป็นลอจิก "1" ทั้งนี้ก็เพื่อให้ 8088 ด้รับข้อมูลที่ต้องการ

สำหรับในระหว่างขบวนการ DMA นั้น DMA-Controller จะควบคุมบัสต่างๆของระบบแทน 8088 และสัญญาณ MEMR จะถูกใช้าในบัสฯเคิลของการอ่านข้อมูลจากหน่วยความจำ (ข้อมูลถูกส่งจากหน่วยความจำไปให้กับอุปกรณ์ I/O)

DRQ1-DRQ3 (DMA Reques 1-3; ขา B18, B6 และ B16) :

ขาสัญญาณทั้งสามนี้เป็นสัญญาณอินพุทแอกทีฟที่ลอจิก " 1 " ซึ่งอุปกรณ์ภายนอกสามารถใช้ในการขอ DMA จากระบบ อดยการป้อนระดับสัญญาณลอจิก "1" ให้กับขา DRQ ขาใดขาหนึ่ง

(ขา DRQ ทั้งสามจะต่อเข้ากับ DRQ1-DRQ3 ของ 8237A-5)

เมื่อ 8237A-5 ด้รับสัญญาณนี้แล้วก็จะตรวจสอบว่ามีารขอ DMA ในแชนแนลที่ลำดับความสำคัญ (Priority) สูงกว่าหรือไม่ ถ้าไม่มีก็จะทำการขอ DMA จาก 8088 และตอบรับการขอ DMA จากอุปกรณ์ภายนอก (สัญญาณ DACK ของแชนแนลที่ขอ DMA แอกทีฟ) แต่ถ้ามี 8237A-5 ก็จะทำการขอ DMA ให้กับแชนแนลที่มีลำดับความสำคัญสูงกว่า ภายใน ROM BIOS ของ IBM/PC จะโปรแกรม 8237A-5 ให้ DRQ1 มีลำดับความสำคัญสูงสุดและ DRQ3 มีลำดับความสำคัญต่ำสุด ดังนั้นถ้ามีการขอ DMA ของอุปกรณ์นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีภายนอกผ่านทางแชนแนลที่ (CRQ1) และแชนแนลที่ 2 (DRQ2) 8237A-5 ก็จะทำ การขอ DMA ให้กับแชนแนลที่ 1 ก่อน จากนั้นเมื่อเสร็จจากขบวนการ DMA ของแชนแนลที่ 1 แล้ว จึงจะทำการขอ DMA ให้กับตัวแชนแนลที่ 2

อย่างไรก็ตาม 8237A-5 ยังมีแชนแนลสำหรับการขอ DMA อยู่อีก 1 แชนแนล คือ แชนแนลที่ 0 (DRQ0) ซึ่งในความเป็นจริงแล้วแชนแนลนี้จะมีลำดับความสำคัญที่สูงกว่าแชนแนลที่ 1 แต่จะไม่ถูกต่อออกมายังขาของสล็อต เนื่องจาก IBM/PC จะใช้แชนแนลที่ 0 นี้ในการรีเฟรชหน่วยความจำที่เป็น Dynamic RAM

ในการขอ DMA นั้นสัญญาณ DRQ นี้ จะต้องแอกทีฟอยู่ในช่วงระยะเวลาหนึ่งเท่านั้น ถ้าสัญญาณนี้แอกทีฟอยู่นานเกินไป จะทำให้เกิดขบวนการ DMA ขึ้นมากกว่า 1 ขบวนการได้ สำหรับวงจรที่ขอ DMA ไปด้วยแล้ว จะใช้สัญญาณตอบรับการขอ DMA หรือสัญญาณ \overline{DACK} ของแชนแนลที่ขอ DMA นั้น ในการรีเซ็ตสัญญาณ DRQ เช่น อุปกรณ์ภายนอกที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) ก็จะตรวจสอบการตอบรับการขอ DMA จากสัญญาณ \overline{DACK} ของแชนแนลที่ 1 ($\overline{DACK1}$) เมื่อได้รับสัญญาณจาก $\overline{DACK1}$ แล้ว ก็จะรีเซ็ตสัญญาณ DRQ1 (เปลี่ยนจากลอจิก "1" เป็น "0")

$\overline{DACK0}$ – $\overline{DACK3}$ (DMA Acknowledge 0–3; ขา B19, B17, B26 และ B15) :

สัญญาณทั้ง 4 นี้ เป็นเข้าที่พุทแอกทีฟที่ลอจิก "0" ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้วงจรภายนอกที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMA เพื่อให้การส่งผ่านข้อมูลระหว่างอุปกรณ์ I/O ที่ขอ DMA กับหน่วยความจำเกิดขึ้นโดยตรง (คือไม่ต้องผ่าน 8088) โดยสัญญาณ \overline{DACK} ที่จะแอกทีฟที่แชนแนลใดก็ขึ้นอยู่กับว่าขบวนการ DMA ที่จะเกิดขึ้นนั้น เป็นการตอบสนองต่อการขอ DMA ในแชนแนลใดเช่นถ้าขบวนการ DMA ที่จะเกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ในแชนแนลที่ 2 (DRQ2) สัญญาณ $\overline{DACK2}$ ก็จะแอกทีฟเป็นต้น

ดังที่ได้กล่าวแล้วว่าสัญญาณ DRQ0 นั้น จะไม่ถูกต่อออกมายังขาสล็อต ดังนั้น วงจรอินเทอร์เฟซจึงไม่สามารถที่จะขอ DMA ผ่านทางแชนแนลที่ 0 ได้ แต่สัญญาณ $\overline{DACK0}$ จะถูกต่อออกมายังสล็อตด้วย (ขา B19) ทั้งนี้ก็เพื่อแสดงให้วงจรอินเทอร์เฟซต่างๆทราบว่า ขบวนการ DMA ที่เกิดขึ้นในช่วงเวลาที่ $\overline{DACK0}$ แอกทีฟนั้น เป็นขบวนการที่ใช้สำหรับการรีเฟรชหน่วยความจำที่เป็น Dynamic RAM ซึ่งวงจรอินเทอร์เฟซที่ใช้หน่วยความจำประเภทนี้ สามารถที่จะนำมาใช้ในการรีเฟรช Dynamic RAM ที่อยู่ในวงจรได้โดยที่การรีเฟรชหน่วยความจำนั้นจะเกิดขึ้นในทุกๆ 15,12 usec. หรือ

+12Vdc (ขา B9) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC+12V ของระบบ โดยที่จะมีค่าความเที่ยงตรง (Regulated) \pm 5% คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

-5Vdc (ขา B5) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) \pm 10% คืออยู่ในช่วง -5.5 ถึง -4.5 Vdc

-12Vdc (ขา B7) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) \pm 10% คืออยู่ในช่วง -13.2 ถึง -10.8 Vdc

GND (ขา B1, B10 และ B31) :

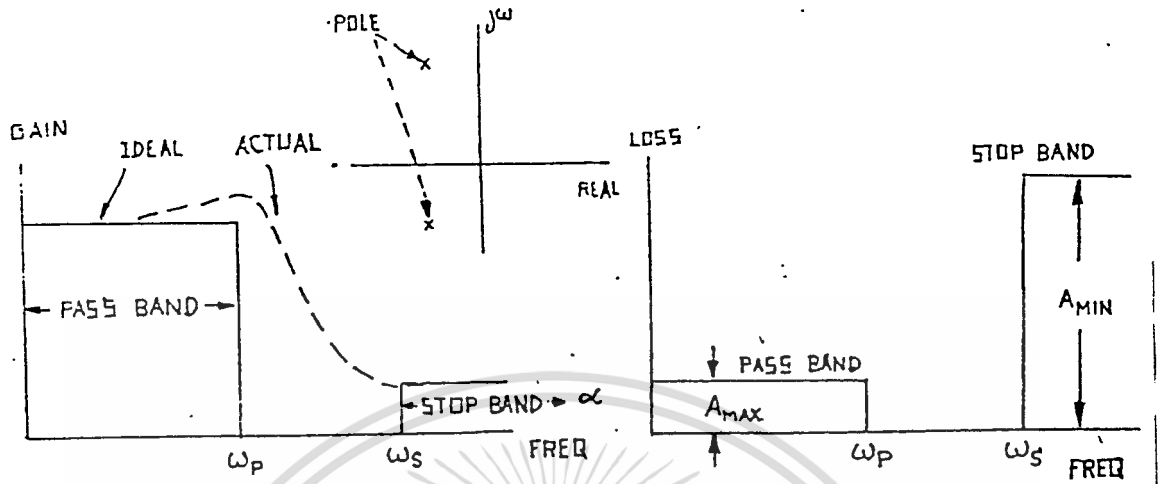
ขาทั้งสามนี้จะต่อเข้ากับกราวด์ (Ground) ของระบบ

การจัดสัญญาณบนสล็อตของ IBM PC/XT

สำหรับใน IBM PC/XT นั้นจะมีสล็อตสำหรับเชื่อมต่อกับวงจรภายนอกได้มากขึ้นคือใน IBM PC/XT จะทำการเพิ่มจำนวนสล็อตบนเมนบอร์ดเป็น 8 สล็อตจากเดิมที่มีอยู่เพียง 5 สล็อตบน IBM PC เพียงแต่สัญญาณต่างๆที่จะถูกส่งออกมายังขาของสล็อตที่ 8 นั้นจะถูกต่อผ่านวงจรขับกระแส (Buffer) ก่อน และในสล็อตที่ 8 นี้ขา BS จะถูกใช้งานด้วยโดยจะถูกใช้เป็นขา CARD SELECT (หรือ Card Selected) ซึ่งขาสัญญาณนี้จะ เป็นสัญญาณอินพุต จากวงจรภายนอกที่เสียบอยู่ที่สล็อตที่ 8 เพื่อให้วงจรบนเมนบอร์ดทราบว่าการ์ดที่อยู่บนสล็อตนี้ถูกเลือกใช้งานอยู่ ซึ่งจะทำให้ Driver บนเมนบอร์ดทำการอ่านหรือส่งข้อมูลไปยังสล็อตที่ 8

Low pass Filter : LPF

คุณสมบัติพื้นฐานของ LPF คือ ผ่านความถี่ต่ำ และบั่นทอนความถี่สูง ดังรูปที่ 1 ที่แสดงตามภาพข้างล่าง



รูปที่ 1

LPF จะต้องผ่านสัญญาณ DC จนถึง Cut off Frequency ω_p ซึ่งในช่วงของ Pass Band จะต้อง Loss ไม่เกิน A_{max} dB ความถี่ที่สูงกว่า ω_b จนถึง infinity จะต้อง Loss อย่างต่ำ A_{min} dB ความถี่จาก $\omega_b - \infty$ เรียกว่า stop band

Second order LPF จะมีฟังก์ชันอยู่ในรูป

$$LPF (Gain) = \frac{K\omega_p^2}{s^2 + \frac{\omega_p}{Q}s + \omega_p^2} ; Gain = k$$

จากการประมาณค่าโดย Bode plot Loss ของ stop band จะมีค่าตามกำลังสูงสุดของ s หรือ order ของ filter ดังต่อไปนี้

ตารางที่ 1 Stop band loss ของ Filter order ต่างๆ

order ของ Filter	Lowpass or Highpass		Band pass* Loss	
	OCTAVE	DECADE	OCTAVE	DECADE
1	6 dB	20 dB	-	-
2	12 dB	40 dB	6dB	20 dB
3	18 dB	60 dB	-	-
4	24 dB	80 dB	12 dB	40 dB
5	30 dB	100 dB	-	-
6	36 dB	120 dB	18 dB	60 dB

หมายเหตุ Band Pass Filter มีเฉพาะ order คู่เท่านั้น

Butterworth Filters

Butterworth จะให้คุณสมบัติของ Flattest Amplitude ซึ่งเป็นชนิดของ Filter ที่ถูกใช้งานมากที่สุด

Loss ของ Butterworth LPF จะสอดคล้องกับสมการ

$$\text{LPF Loss} = 1 + E \left(\frac{S}{W_p} \right)^n$$

E คือ Butterworth constant หาได้จาก

ตารางที่ 2 Normalizer Butterworth FPF Loss Fuction

n Normalizer Fuction ($w_p = 1$)

1 $(S + 1)$

2 $(S + 1.414 S + 1)$

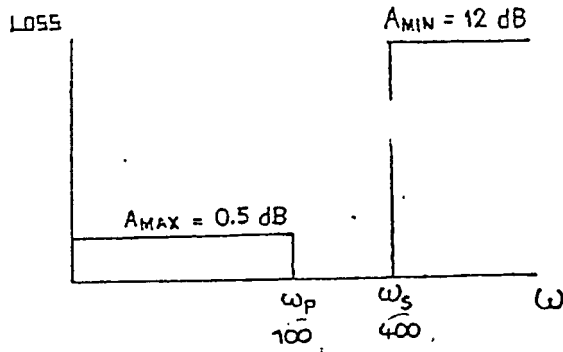
3 $(S + 1)(S + S + 1)$

4 $(S + 0.76537 S + 1)(S + 1.84776 S + 1)$

5 $(S + 1)(S + 0.61803 S + 1)(S + 1.61803 S + 1)$

ตัวอย่าง หา Butterworth LPF Loss Fuction ของ LPF ที่มีคุณสมบัติดังนี้ผ่านความถี่จาก DC-100 RAD/sec โดยมี Loss ไม่เกิน 0.5 dB จะมีความถี่ 400 RAD/Sec จะต้อง Loss อย่างน้อย 12 dB

จากคุณสมบัติของ LPF ที่ต้องการได้



หาค่า E จากสมการ 9

$$E = \sqrt{10^{0.1 \times 0.5} - 1}$$

$$= 0.349$$

$$\approx 0.35$$

รูปที่ 3

แทนค่า $A_{min} = 12, E = 0.35, \omega_p = 100, \omega_s = 400$ ได้ $n = 1.73$

ใช้ Butterworth LPF order 2

จากตารางที่ 2 order 2 Loss = $S + 1.414 S + 1$

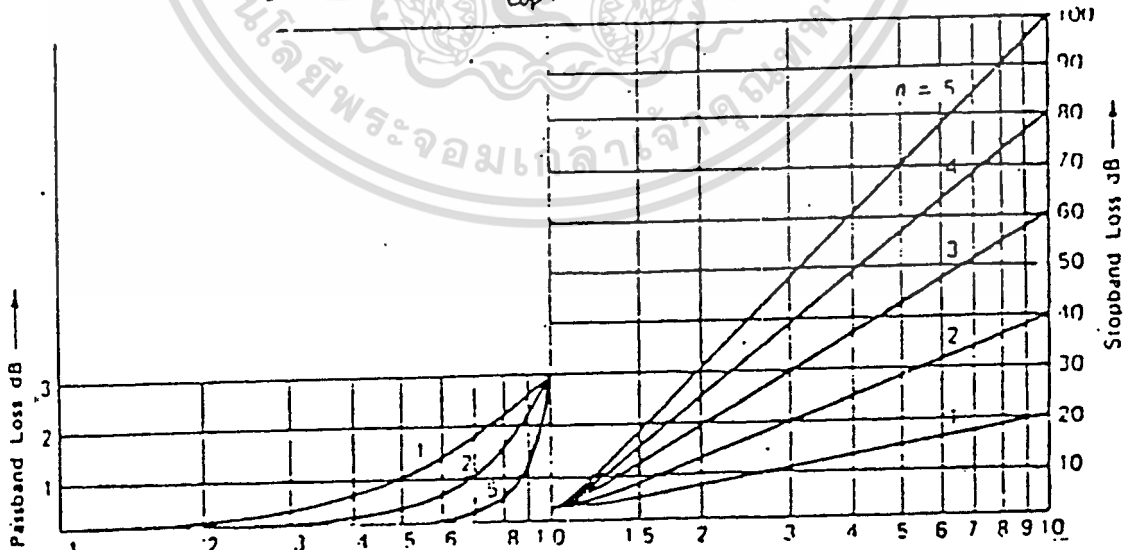
denormalized โดยการแทน S ด้วย $S\left(\frac{\omega}{\omega_p}\right)^{1/n} = 0.00595$ จะได้ที่

$\omega = \omega_p$ ค่า Loss จะเป็น

$$A(\omega_p) = 10 \log(1 + E^2) \text{ dB}$$

ในการออกแบบวงจร จะเป็นการสะดวกมากถ้าทำการคำนวณและ Plot ค่า Loss ของ Butterworth LPF ที่ order n ต่างๆ โดย normalized Frequency : Ω

$$\Omega_{BUT} = E^{1/n} \left(\frac{\omega}{\omega_p}\right) \quad (11)$$



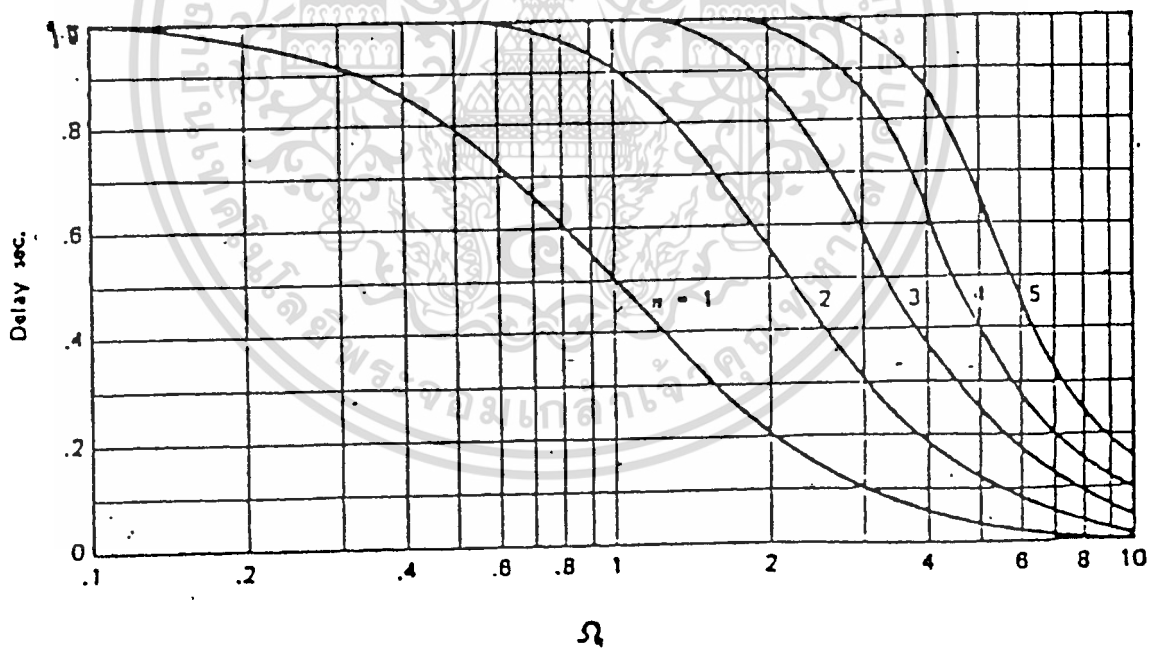
รูปที่ 4

ค่าของ Loss ของ normalized Butterworth LPF จะเป็นดังรูปที่ 4 และใน
 รางที่ 2 คือ normalized Butterworth Loaa Fuction ที่ order ต่างๆใน
 การหาค่า Butterworth Fuction ที่ความถี่ ω_p ใดๆ สามารถทำได้โดยการหา
 denormalized อดยการแทนค่า s ในตารางที่ 2 ด้วย $s(j\omega/\omega_p)^{1/n}$

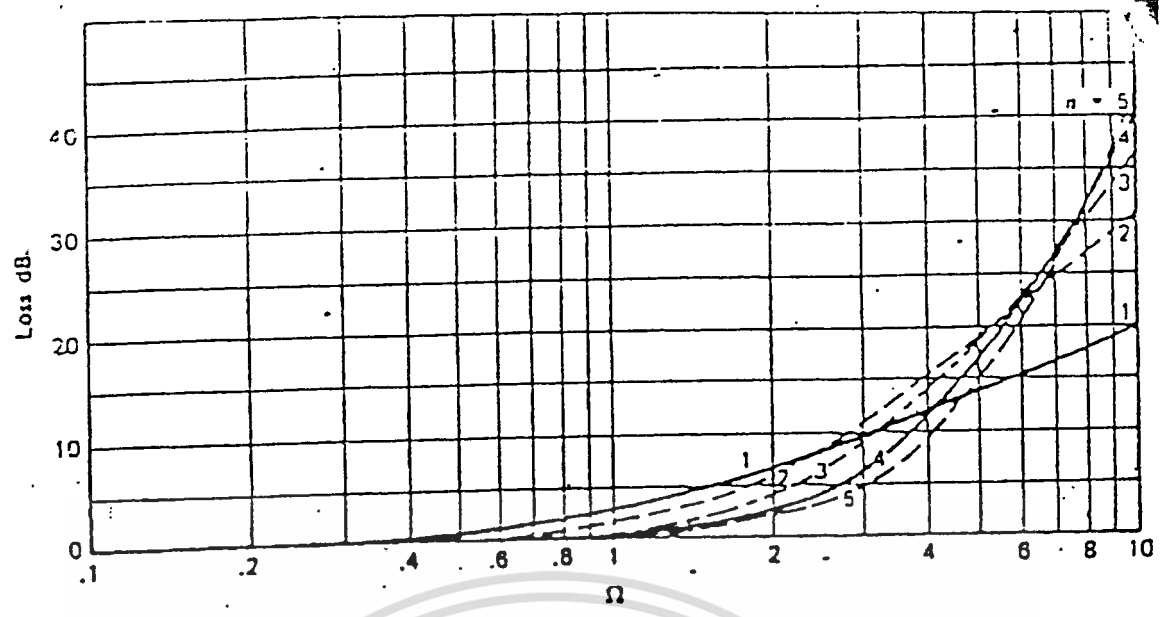
จากตารางที่ 2 จะพบว่าค่า Q ของ Butterworth LPF คือ LPF $n=2$
 $Loss = S + 1.414 S + 1$ (normalized $\omega_p = 1$)

Loss Fn ของ LPF คือ $S^2 + \frac{wps}{Qp} S + \omega_p^2$

จะได้ค่า Q ของ Butterworth LPF ที่ order 2 คือ $1/1.414 = 0.707$



รูปที่ 6



รูปที่ 7

ตัวอย่าง คำนวณหาค่า order ของ LPF ที่มีคุณสมบัติดังต่อไปนี้

ก) Delay จะต้อง Flat 1% จาก DC 2 KHz และ

ข) Loss ที่ 6 KHz ต้องมากกว่า 25 dB

Delay flat 1% จากรูปที่ 6 Delay = 0.99

n = 3 จะได้ค่า 1.5 หากค่าได้จาก

ความถี่ 2 KHz มีค่า normalized Frequency เป็น 1.5

$$\begin{aligned} \text{" 6 KHz " " " " } &= \frac{1.5 * 6}{2} = 4.5 \end{aligned}$$

พิจารณารูปที่ 7 ที่ = 4.5 , n = 3 ได้ Loss ประมาณ 16 dB : n = 4 จะ

ได้ 2 ดังนั้น 6 KHz จะมีค่า normalized Freq = $\frac{2}{2} * 6 = 6$

จากรูปที่ 7 = 6, n = 4 ได้ Loss ประมาณ 22 dB, n = 5 ได้

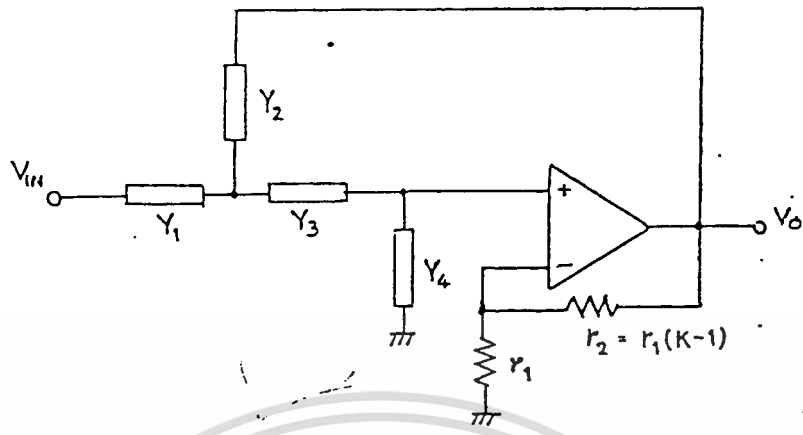
= 2.5 ดังนั้น 6 KHz จะมีค่า เป็น $2.25 * 6 = 7.5$

2

จากรูปที่ 7 = 7.5, n = 5 ได้ Loss ประมาณ 30 dB ดังนั้นต้องใช้ Bessel

LPF order 5

Positive Feedback Topology



รูปที่ 8 Generalized PFB Filter circuit

รูปที่ 8 เป็นวงจรทั่วไปของ Active Filter ที่ใช้ OP AMP เพียงตัวเดียวและใช้วิธีการป้อนกลับแบบบวก จากวงจรรูปที่ 8 จะได้

$$T = \frac{V_o}{V_{in}} = \frac{KY_1Y_3}{Y_2Y_3 + Y_4(Y_1 + Y_2 + Y_3) + (1-K)Y_2Y_3}$$

โดยการใช้ $Y_1 = 1/R_1$, $Y_2 = sC_1$, $Y_3 = 1/R_2$, $Y_4 = sC_2$ ได้

$$T = \frac{V_o}{V_{in}} = \frac{K}{s + \left(\frac{1}{R_1C_1} + \frac{1}{R_2C_2} + \frac{1-K}{R_2C_2} \right) + \frac{1}{R_1C_1R_1C_2}}$$

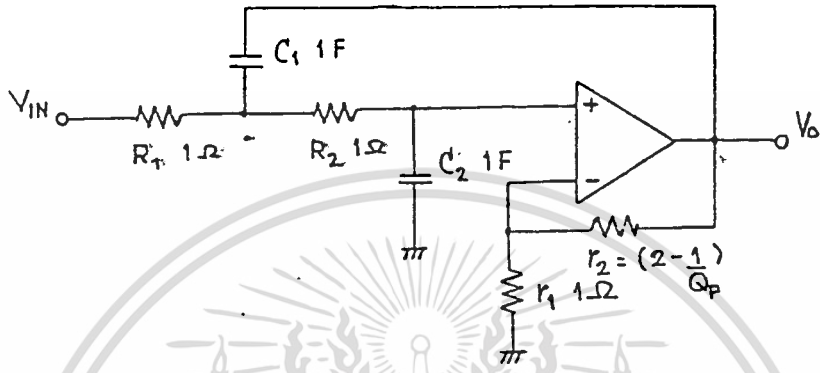
ซึ่งจะเห็นว่าอยู่ในรูปของสมการที่ 4 หรือ LPF

$$T_{LPF} = \frac{KW_p^2}{s^2 + \frac{s}{Q_p} + W_p^2}$$

ซึ่งจะได้ $\omega_p = \frac{1}{R_1 C_1 R_2 C_2}$

และ $R_1 C_1 R_2 C_2$

$$Q_p = \frac{1}{\frac{1}{R_1 C_1} + \frac{1}{R_2 C_1} + \left[\frac{1-K}{R_2 C_2} \right]}$$



รูปที่ 9

Normalized LPF circuit

ถ้าเรากำหนดค่า $C = C_2 = 1F, R_1 = R_2 = R$ จะได้

$$R_2 = R_2 = \frac{1}{\omega_p} \text{ และ Gain } K = 3 - \frac{1}{Q_p}$$

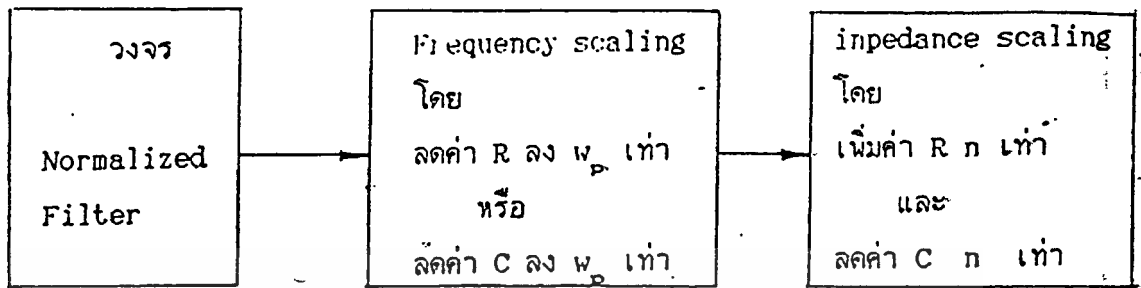
และถ้าให้ cut off Frequency $\omega_p = 1 \text{ RAD/sec}$ จะได้วงจรของ Normalized LPF ดังรูปที่ 9

Impedance and Frequency scaling

จากสมการ 16, 17 และ 18 จะเป็นได้ว่าค่า RC ใน Transfer Function จะอยู่ในลักษณะ RC Time Constant ถ้าเราเพิ่มค่า R ขึ้นทุกตัว n เท่า และลดค่า C ทุกตัวลงด้วยอัตราส่วน n เท่าเช่นเดียวกัน ผลคูณ RC ยังคงมีค่าเท่าเดิม ซึ่งจากวงจร Normalized ก็จะสามารถเปลี่ยนเป็นวงจรที่เราต้องการได้โดยง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในทานองเดียวกับ impedance scaling เราสามารถเปลี่ยนค่า $w_p=1$ ของวงจร Normalized เป็นค่า w_p ใดๆ ได้โดยลดค่า Time constance ลงนั้น คือ



รูปที่ 11

ตัวอย่าง ออกแบบวงจร LPF ที่มีคุณสมบัติดังต่อไปนี้

- Amplitude ของ Passband จะต้อง Flat ที่สุด
- ส่งผ่านความถี่ที่ต่ำกว่า 4 KHz โดยมี Loss ไม่เกิน 3 dB และมีค่า Passband Gain เป็น 10 เท่า
- กาลัดความถี่ที่มากกว่า 8 KHz อย่างน้อย 15 dB

จากความต้องการ Passband Flat ที่สุด (Maximum) ใช้ Butterworth

$f_p = 4 \text{ KHz}, A_{\max} = 3 \text{ dB}, f_B = 8 \text{ KHz}, A_{\min} = 15 \text{ dB}$ จะต้องหา order ของ Filter

$$\text{หาค่า } E \text{ จากสมการที่ 9 } E = 2 \cdot 10^{0.1 \cdot 3 - 1} = 1$$

หา order n จากสมการที่ 12

$$n = \frac{\text{LOG} \left[\frac{10^{0.1 \cdot 15} - 1}{1} \right]}{\text{LOG} \left(\frac{8}{4} \right)^2} = 2.46$$

ใช้ filter order 3 หรืออาจหาค่า order ได้จากกราฟในรูปที่ 4 ที่ $n=3$
 ได้ Loss = 20 dB จากตารางที่ 2 จะได้ normalized Butterworth LPF
 Function เป็น

$$\frac{1}{(s+1)(s^2+s+1)}$$

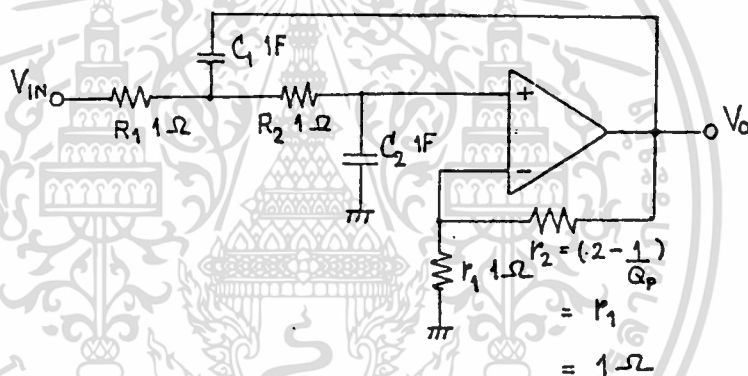
TLPF Gain 1 = (s + 1)(s²+s+1)

สามารถสร้างได้ filter order 3 โดยการ cascade กันของ

$$\frac{1}{s+1} \cdot \frac{1}{s^2+s+1}$$

พิจารณาส่วน Second order T = $\frac{1}{s^2+s+1}$

จะได้ $Q_p = 1$ ดังนั้นจะได้ Normalized LPF circuit เป็น



วงจรในรูปที่ 13 คือ Normalized LPF ที่มีค่า $Q_p = 1$ และ Gain
 มีค่าเป็น $3-1 = 2$ เท่า แต่จากที่เราต้องการ Gain = 10 ในส่วนของ second
 Q_p

order มี Gain = 2 ดังนั้นในส่วนของ First order จะมี Gain = $\frac{10}{2} = 5$

2

จะได้ Normalized LPF ที่ต้องการดังรูปที่ 14


```

#include "stdio.h"

#include "dos.h"

main(void)

{
    int detect,x;

char ans;

clrscr();

outp(0x303,0x98);

printf(" speaking (y/n) :");

ans=getchar();

if((ans=='y')||(ans=='Y'))
{
    printf(" write data to ram \n");
    outp(0x302,0x6);/*clear*/
outp(0x302,0x0);/*write*/
detect=inp(0x302);
while(((detect)&(0x10))!=0)
{
    detect=inp(0x302);
    printf(" end of convert \n");
    printf(" press any key \n");
    getch();
}
}
else
{
    printf(" error \n");
    printf(" press any key \n");
    getch();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include"stdio.h"

#include"dos.h"

FILE *fp;

main(void)
{
    int data,detect,z;

    char name[11];

path1:clrscr();

    outp(0x303,0x98);

    printf(" you want save data to file name : ");

    scanf("%s",name);

    if((fp=fopen(name,"wb"))==NULL)
    {
        printf(" can't open file %s \n",name);

        printf(" press any key to continuous ");

        goto path1;
    }
    else
    {
        z=0;

        printf(" save data to %s \n",name);

        printf(" wait,please \n");

        outp(0x302,0x5);/*clear*/

        delay(1);

        outp(0x302,0x1);/*negative clock*/

        delay(1);

        putc(inp(0x300),fp);/*put data to file*/

        outp(0x302,0x3);/*positive clock*/

        delay(1);

        detect=inp(0x302);

        while((detect&0x10)!=0)/*detect end*/

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิได้ทำซ้ำ, คัดลอก, ผลิตซ้ำ, หรือเผยแพร่ข้อมูลใดๆไปยังผู้อื่นโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
outp(0x302.0x3);  
  
z++;  
  
detect=inp(0x302);  
  
}  
  
fclose(fp);  
  
printf(" save is valid \n");  
printf(" data= %d \n",z);  
printf(" press any key \n");  
getch();  
  
}  
}
```



```

#include "dos.h"

#include "stdio.h"

FILE *fp;

main(void)
{
    int i,j,nub;
    char name[11];
    clrscr();
    outp(0x303,0x98);
    printf(" enter file name :");
    scanf("%s",name);
    if((fp=fopen(name,"rb"))==NULL)
    {
        printf(" can't open file %s \n",name);
        printf(" press any key \n");
        getch();
    }
    else
    {
        printf(" file %s is valid \n",name);
        nub=0;
        while(!feof(fp))
        {
            nub++;
            outp(0x301,getc(fp));
            for(i=0;i<4;i++)
        for(j=0;j<18;j++):
            }
        fclose(fp);
        printf(" end of convert \n");
        printf(" data= %d \n",nub);
        printf(" press any key ");
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "dos.h"

#include "graphics.h"

#include "stdio.h"

#include "conio.h"

FILE *fp;

int x1,x2,y1,y2;

main(void)

{
    int i,z,x,y;

    int data,plot,driver,mode,show,detail[3];

    char name[11];

    char conti;

    mode=DETECT;driver=DETECT;

    initgraph(&driver,&mode,"c:\\lang\\tc");

    path1:clrscr();

    rectangle(4,4,270,65);

    setfillstyle(1,BLACK);

    floodfill(5,5,WHITE);

    gotoxy(3,2);

    printf(" ENTER FILE NAME : ");

    scanf("%s",name);

    if((fp=fopen(name,"rb"))==NULL)

    {
        gotoxy(3,3);

        printf(" CAN'T OPEN FILE %s \n",name);

        gotoxy(3,4);

        printf(" PRESS ANY KEY TO CONTINUOUS ");

        getch();goto path1;

    }

    }

}

printf( " ENTER NUMBER OF DETAIL " );

```

```

scanf("%d",detail);

setcolor(RED);

line(36,240,610,240);/*hor*/
line(36,113,42,113);/*+2.5*/
line(42,90,42,390);/*ver*/
line(36,367,42,367);/*-2.5*/

setcolor(BLUE);

outtextxy(2,112,"+2.5");
outtextxy(15,239,"0");
outtextxy(2,366,"-2.5");
outtextxy(2,75,"AMPLITUDE (V)");
outtextxy(570,250,"TIME (t)");
x1=42;y1=240;show=0;
while(!feof(fp))
{
    data=getc(fp);
    show++;
    if(data>0x85)
    {
        plot=data-0x85;
y2=240-plot;
x2=x1+2;
line(x1,y1,x2,y2);
    }

    if(data<0x85)
    {
        plot=0x85-data;
y2=240+plot;
x2=x1+2;
line(x1,y1,x2,y2);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลง y2=240 และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<2=x1+2;
line(x1,y1,x2,y2);
}
if(detail!=0)
{
for(i=0;((i<^detail)&(data!=EOF));i++)
data=getc(fp);
x1=x2;y1=y2;
}
if(detail==0)
{
x1=x2;
y1=y2;
}
if(^detail==0)
{
gotoxy(3,4);
printf(" DATA = %d POINT ",show);
}
else
{
show=20482;
gotoxy(3,4);
printf(" DATA = %d POINT ",show);
}
}
}
fclose(fp);
path2:gotoxy(26,24);
printf(" PRESS KEY TO CONTINOUS (Y/N)");
conti=getch();
if((conti=='Y')||(conti=='y'))
{
setcolor(WHITE);
goto path1;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }  
    if(!((conti=='n')||(conti=='N')))  
goto path2;  
    else  
closegraph();  
}
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้คุณเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น. อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include "dos.h"

#include "stdio.h"

#include "graphics.h"

FILE *fp;

main(void)

{
    int data,rob,driver,mode,total,plot,x1,x2,y1,y2;

    char name[11],cont1,cont2;

    driver=DETECT;

    mode=DETECT;

    initgraph(&driver,&mode,"c:\\lang\\tc");

path1:clrscr();

    rectangle(4,4,270,65);

    setfillstyle(1,BLACK);

    floodfill(5,5,WHITE);

    gotoxy(3,2);

    printf(" ENTER FILE NAME : ");

    scanf("%s",name);

    if((fp=fopen(name,"rb"))==NULL)

    {
        gotoxy(3,3);

        printf(" CAN'T OPEN FILE %s ",name);

        gotoxy(3,4);

        printf(" PRESS ANY KEY TO CONTINUOUS ");

        getch();

        goto path1;

    }

    else

    {
        total=0;

        data=getc(fp);

        path2:setcolor(RED);

```

```

line(36,113.42,113);/*+2.5*/
line(42,90,42,390);/*ver*/
line(36,367.42,367);/*-2.5*/
setcolor(BLUE);
outtextxy(2,112,"+2.5");
outtextxy(15,239,"0");
outtextxy(2,366,"-2.5");
outtextxy(2,75,"AMPLITUDE (V)");
outtextxy(570,250,"TIME (t)");
x1=42;y1=240;rob=0;
while((rob<260)&(data!=EOF))
{
    data=getc(fp);
rob++;
if(data>0x85)
{
    plot=data-0x85;
    y2=240-plot;
    x2=x1+2;
    line(x1,y1,x2,y2);
    x1=x2;y1=y2;
}
if(data<0x85)
{
    plot=0x85-data;
    y2=240+plot;
    x2=x1+2;
    line(x1,y1,x2,y2);
    x1=x2;y1=y2;
}
if(data==0x85)
{
    y2=240;x2=x1+2;
    line(x1,y1,x2,y2);
}
}

```

```

        x1=x2;y1=y2;
    }
}

total=total+rob;

gotoxy(3,4);

printf(" DATA = %d POINT",total);

gotoxy(24,24);

printf(" PRESS KEY TO CONTINUOUS (Y/N)");

    path3:conti1=getch();

    if((conti1=='y')|(conti1=='Y'))
    {
        setcolor(WHITE);

clrscr();

rectangle(4,4,270,65);

setfillstyle(1,BLACK);

floodfill(5,5,WHITE);

gotoxy(3,2);

printf(" %s",name);

if(data!=EOF)

    goto path2;

else

{
    gotoxy(3,3);

    printf(" DATA = %d POINT",total);

    gotoxy(3,4);

    printf(" END OF FILE");

    goto path5;

}

}

}

if(!(conti1=='n')|(conti1=='N'))

    goto path3;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            fclose(fp);
            path5:setcolor(WHITE);
            clrscr();
            rectangle(4,4,270,65);
            setfillstyle(1,BLACK);
            floodfill(5.5,WHITE);
            gotoxy(3,2);
            path4:printf(" PRESS KEY TO CONTINUOUS (Y/N)");
            conti2=getch();
            if((conti2=='y')|(conti2=='Y'))
            {
                setcolor(WHITE);
                goto path1;
            }
            if(!(conti2=='n')|(conti2=='N'))
                goto path4;
            else
                closegraph();
        }
    }
}

```

หนังสืออ้างอิง

- ธานีรินทร์ ภาวะศาสนวงศ์ , ทินกร ดูก " การอินเทอร์เฟส IBM PC" สำนักพิมพ์ พิกส์-เซ็นเตอร์
- ธันวา ศรีประโมง " การวิเคราะห์เสียงภาษาไทยแทนในความถี่ฮาร์โมนิกส์ " เอกสารประกอบการประชุมทางวิศวกรรมไฟฟ้า ครั้งที่ 15 , 2535.
- " คู่มือไอซีซีพีพีพอร์ท และหน่วยความจำ " บริษัท ซีเอ็ดยูเคชั่น จำกัด , 2529
- ดร. วัลลภ สุรกำพลธร " การประมวลสัญญาณเชิงเลข " สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง , 2533
- วารสารสำนักวิจัย และบริการคอมพิวเตอร์ . สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง , ปีที่ 2 , ฉบับที่ 6
- Trevor J Terrair "Introduction to Digital Filter " The Macmillan Press Ltd.,1980
- Thomas W Parsons "Voice and speech processing."McGraw-Hill Back Company,1987
- D.O. Shaughnessy., "Speech Communication " , Addison Wesleypublishing company,1987
- Roman Kuc., " Introduction to digital signal procession " McGRAW-Hill company Inc, 1988.

กิตติกรรมประกาศ

ในการทำปริญญาโทฉบับนี้ สำเร็จได้ด้วยดีนั้นได้รับความช่วยเหลือจากผู้มีพระคุณหลายท่าน ขอกราบขอบพระคุณ คุณพ่อและคุณแม่ของผู้จัดทำที่ให้การสนับสนุนในเรื่องการศึกษา และ ขอกราบขอบพระคุณผู้ช่วยศาสตราจารย์ ดร. รัตติกร วรากุลศิริพันธ์ อาจารย์ที่ปรึกษา พี่สมศักดิ์ จันทัน , พี่ศังกรศรัณย์ ล่องชุมผล ผู้ซึ่งให้ความอนุเคราะห์ทางด้านความรู้แนวทาง และอุปกรณในการทำงาน

ขอบคุณสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , สำนักวิจัยและบริการคอมพิวเตอร์ ตลอดจนเพื่อนๆ ที่คอยให้กำลังใจดิฉัน นายอัคร ช่วยพิมพ์อีกแรง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้