



# เครื่องปิด-เปิดหลอดไฟด้วยเสียงพูด

## LIGHTING ON/OFF CONTROLLED BY SPEECH RECOGNITION

โดย

นางสาวพรพรรณ สุวรรณวงศ์

นางสาวพิมพ์ฉวี พวงลำไย

วัน เดือน ปี..... 18.ค.ค. 2541  
เลขทะเบียน..... 039073  
เลขเรียกหนังสือ..... 140094 พ.ม.ศ.

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เครื่องปิด-เปิดหลอดไฟด้วยเสียงพูด

LIGHTING ON/OFF CONTROLLED BY SPEECH RECOGNITION

โดย

นางสาวพรพรรณ สุวรรณวงศ์ 37014280

นางสาวพิมพ์ฉัตร พวงลำไย 37014289

อาจารย์ที่ปรึกษา

ผศ. นิภา ลีสารุจิ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมโทรคมนาคม

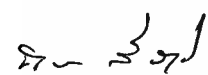
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมการเปิด-ปิดหลอดไฟด้วยเสียงพูด

LIGHTING ON/OFF CONTROLLED BY SPEECH RECOGNITION

ผู้จัดทำ

1. นางสาวพรพรรณ สุวรรณวงศ์ 37014280
2. นางสาวพิมพ์ภิลัย พวงคำไช 37014289



(ผศ. นิภา ลีลาจุจิ)

อาจารย์ที่ปรึกษา



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมการปิด-เปิดหลอดไฟด้วยเสียงพูด

LIGHTING ON/OFF CONTROLLED BY SPEECH  
RECOGNITION

โดย นางสาวพรพรรณ สุวรรณวงศ์ 37014280

นางสาวพิมพ์กัญญา พวงลำไย 37014289

อาจารย์ที่ปรึกษา ผศ. นิภา ลีลารุจิ

บทคัดย่อ

เครื่องควบคุมการปิด-เปิดหลอดไฟด้วยเสียงพูดนี้เป็นโครงการที่ประยุกต์มาจากเรื่องระบบการรู้จำเสียงพูด(Speech Recognition) ซึ่งเป็นการนำสัญญาณเสียงพูดในเวลาจริงจากไมโครโฟนซึ่งเป็นสัญญาณอนาล็อกมาเปลี่ยนเป็นสัญญาณดิจิทัล แล้วนำมาวิเคราะห์หาองค์ประกอบทางความถี่ ซึ่งในโครงการนี้วิเคราะห์โดยวิธีการหาค่าอัตราตัดผ่านศูนย์และเปรียบเทียบโดยวิธีทางคณิตศาสตร์ เพื่อนำผลที่ได้จากการวิเคราะห์นี้มาต่อกับอุปกรณ์เชื่อมต่อเพื่อใช้ควบคุมการปิด-เปิดอุปกรณ์ไฟฟ้า

ABSTRACT

Speech-controlled lighting system applies the speech recognition theory which captures speech signal in real time from microphone, digitize, then find the zero-crossing rate. Finally, compare with pre-calculated template by mathematic theory and applies the output to control on/off sequence of the lamps to demonstrate speech recognition application.

## สารบัญ

หน้า

บทที่ 1	บทนำ	1
บทที่ 2	ทฤษฎีและหลักการ	
2.1	ลักษณะของเสียงพูด	4
2.2	ทฤษฎีการสุ่ม	5
2.3	การเข้ารหัสแบบพัลส์โค้ดมอดูเลชัน	9
2.4	การวิเคราะห์เสียงพูด	12
2.5	การตรวจจับจุดสิ้นสุด	22
บทที่ 3	การคำนวณและการสร้าง	24
บทที่ 4	การทดลองและผลการทดลอง	29
บทที่ 6	สรุปผลการทดลอง	39
บรรณานุกรม		
ภาคผนวก ก	โปรแกรม	
ภาคผนวก ข	รายละเอียดของอุปกรณ์	



# บทที่ 1

## บทนำ

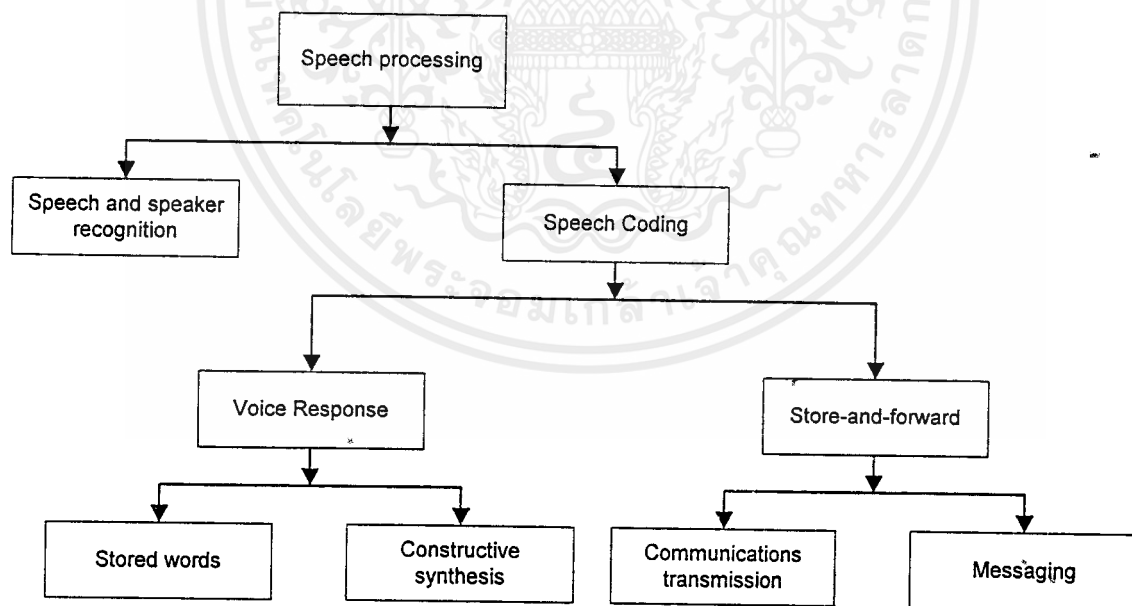
เสียงพูดนับเป็นตัวกลางการติดต่อสื่อสารที่สำคัญที่สุดของมนุษย์ และได้มีการศึกษารูปร่างลักษณะของเสียงพูดมาช้านานแล้ว จนมาถึงยุคของคอมพิวเตอร์ การสนใจเกี่ยวกับเสียงพูดจึงได้แบ่งออกเป็น 2 ลักษณะคือ

1. การสื่อสารระหว่างมนุษย์กับเครื่องจักร (machine) ด้วยเสียงพูด
2. การสื่อสารที่มีประสิทธิภาพมากขึ้นระหว่างมนุษย์ด้วยกันเองโดยผ่านคอมพิวเตอร์ ซึ่งคำว่า

ประสิทธิภาพนี้อาจหมายถึง

- การบีบ(compress) คำพูดหลายคำให้สามารถส่งออกไปได้พร้อมๆกันในช่องสัญญาณช่องเดียว ซึ่งทำให้ประหยัดช่องสัญญาณ
- การแก้เสียงพูดที่ผิดเพี้ยนไปให้เป็นคำพูดที่ดีขึ้น เข้าใจได้มากขึ้น
- ทำให้การติดต่อสื่อสารด้วยเสียงพูดมีระบบความปลอดภัย

การติดต่อสื่อสารระหว่างมนุษย์กับเครื่องจักรโดยเสียงพูดนั้น เป็นวิชาใหม่ที่มีจุดประสงค์ลดต้นทุนและเพิ่มความรวดเร็วในการส่งงานเครื่องกลและสามารถส่งงานได้ โดยไม่จำเป็นต้องใช้สายคาและมือ กระบวนการเกี่ยวกับเสียงพูดมีหลายประเภท แสดงดังบล็อกไดอะแกรมข้างล่าง



รูปที่ 1.1 บล็อกไดอะแกรมแสดงประเภทต่างๆของกระบวนการทางเสียง

การจดจำเสียงพูด เป็นการจดจำรูปแบบอย่างหนึ่ง เพื่อเป็นการเปรียบเทียบรูปแบบตรวจสอบ (test pattern) ซึ่งเป็นตัวแทนของสิ่งที่ไม่รู้เพื่อแยกแยะ หรือจดจำกับรูปแบบอ้างอิง (reference pattern) ซึ่งได้รู้ลักษณะมาก่อนแล้ว

ในแต่ละรูปแบบจะอยู่ในรูปของเวกเตอร์ แต่ละสมาชิกของเวกเตอร์ เป็นค่าที่วัดได้ของลักษณะเด่น (feature) ซึ่งเป็นลักษณะที่วัดได้ของสัญญาณอินพุต และสามารถนำไปใช้ประโยชน์ในการจดจำรูปแบบได้

ในการจดจำเสียงพูดแบบแยกคำ รูปแบบต่างๆ จะเก็บอยู่ในรูปของกลุ่มฟังก์ชันเวลา เพราะค่าของแต่ละลักษณะที่วัดได้นั้น ไม่ได้วัดที่จุดใดจุดหนึ่งของผู้พูด แต่มาจากการวัดทั้งคำ รูปแบบเหล่านี้เราเรียกว่า ต้นแบบ (template)

การจดจำเสียงพูดโดยทั่วไป นิยมแบ่งออกเป็น 4 กลุ่ม ดังนี้

1. การจดจำแบบแยกคำ (Isolated-word recognition) เป็นการจดจำโดยจะต้องมีช่วงหยุดของคำแต่ละคำ
2. การจดจำเฉพาะคำ (Word spotting) ตรวจจับคำเฉพาะใดๆ ที่ปรากฏอยู่ในประโยคนั้น
3. การจดจำแบบต่อเนื่อง (Connected speech recognition) จดจำคำพูดได้ทั้งประโยค โดยไม่ต้องมีช่วงหยุดแต่ละคำ
4. การเข้าใจเสียงพูด (Speech understanding) เป็นแบบสมบูรณ์ของกลุ่มที่ 3 โดยสร้างแหล่งข้อมูลของการออกเสียงและหลักการของภาษานั้น จุดประสงค์หลักก็คือ การพูดที่ไม่ต้องถูกต้องมากนัก แต่ให้เข้าใจความหมายของประโยคนั้นได้ ในทางทฤษฎี ระบบเข้าใจเสียงพูด จะตัดทอนเอาเฉพาะความหมายของประโยค โดยไม่เข้มงวดเรื่องไวยากรณ์

ในโครงการนี้ได้ทำการทดลองในกลุ่มที่ 1 คือการจดจำเสียงแบบแยกคำ ซึ่งการจำแบบแยกคำนั้นเป็นจุดเริ่มต้นของการพัฒนาการจดจำเสียงพูดจนเป็นการจำเฉพาะคำและการจำแบบต่อเนื่องตามลำดับ ช่วงหยุดของแต่ละคำพูดทำให้การจดจำคำง่ายขึ้น เพราะสามารถแยกแยะจุดสิ้นสุดของคำ (หมายรวมถึง จุดเริ่มต้นและจุดสิ้นสุดของคำนั้น) ทำให้ผลที่เกิดจากความไม่ชัดเจนของการพูดลดน้อยลงไปได้มาก การจำแบบแยกคำขณะที่ออกเสียงจะต้องระมัดระวังอย่างมาก เพราะต้องการช่วงหยุดระหว่างการพูดแต่ละคำ ทำให้การพูดขาดความไพเราะของภาษาไป ไม่เหมือนกับวิธีการอื่นๆ ซึ่งสามารถพูดเป็นธรรมชาติได้มากกว่า และใช้ความระมัดระวังในการพูดน้อยกว่า

การจดจำแบบแยกคำ มีหลักการจดจำโดยใช้ลักษณะของสัญญาณเสียงพูด ดังต่อไปนี้

1. พลังงานเสียงต่อเวลา (power versus time)
2. อัตราตัดผ่านศูนย์ (zero crossing rate)
3. สมดุลกลุ่มความถี่รวม (gross spectrum balance) เป็นการเปรียบเทียบพลังงานของกลุ่มความถี่สูงกับกลุ่มพลังงานความถี่ต่ำ
4. รายละเอียดทางสเปกตรัม เช่น

ก. การแปลงฟูเรียร์แบบไม่ต่อเนื่อง (discrete fourier transform; DFT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ข. การทำนายเชิงเส้น ( linear predictive coding;LPC)
- ค. สัมประสิทธิ์พาร์คอร์ ( parcor efficiency)
- ง. เอาท์พุทของฟิลเตอร์แบงก์ ( filter bank) วิธีนี้เป็นวิธีที่นิยมกันที่สุดเพราะราคาถูกและความเร็วสูง ซึ่งในโครงการนี้ ใช้วิธีหาค่าอัตราตัดผ่านศูนย์ ในการแยกแยะเพื่อรู้จำคำ



## บทที่ 2

### ทฤษฎีและหลักการ

#### 2.1 ลักษณะของเสียงพูด

คนเราเปล่งเสียงพูดด้วยอวัยวะที่ใช้ในการออกเสียง(organs of speech) โดยเปล่งเสียงสำคัญตามที่มีอยู่ในระบบภาษาของตน แม้ว่าคนที่อยู่ในสังคมเดียวกันจะใช้ภาษาเดียวกัน แต่ถ้าเราพิจารณาเสียงที่เราเปล่งออกมาจริงๆ ในแต่ละครั้ง ก็อาจจะสังเกตเห็นลักษณะที่แตกต่างกันได้ เราจึงสามารถจำเสียง จำผู้พูด จำคนที่เรารู้จักได้ เสียงพูดนี้มีลักษณะที่จะอธิบายได้ ด้วยหลักเกณฑ์ทางวิทยาศาสตร์ ซึ่งเป็นสากลแม้ภาษาหนึ่งๆจะมีเสียงที่แตกต่างกันไป แต่ละเสียงก็สามารถที่จะนำมาพิจารณาและอธิบายให้รู้ลักษณะการออกเสียง และตำแหน่งที่เกิดของเสียงได้ คำอธิบายนี้จะทำให้เข้าใจลักษณะเสียงทุกเสียง ซึ่งวิชาที่ว่าด้วยเสียงพูดนี้เรียกว่า “วิชาสัทศาสตร์ (Phonetics)”

ในการศึกษาเรื่องเสียงพูดจะสามารถแบ่งได้เป็น 2 ลักษณะคือ

1. สรีระสัทศาสตร์ (Articulatory Phonetics) - เป็นการศึกษาเสียงพูดจากอวัยวะและการเคลื่อนไหวของอวัยวะที่ทำให้เกิดเสียงพูด การอธิบายก็จะอธิบายโดยอาศัยลักษณะและอาการเคลื่อนไหวของอวัยวะที่เกี่ยวข้องในการเปล่งเสียงพูดนั้น
2. กลศาสตร์ (Acoustic Phonetics) - เป็นการศึกษาเสียงพูดจากลักษณะคลื่นเสียงที่ผู้พูดเปล่งออกมาแล้วและผู้ฟังได้ยิน ว่ามีลักษณะทางกลศาสตร์เป็นอย่างไร การศึกษาตามแนวนี้ต้องอาศัยความรู้ทางฟิสิกส์และคณิตศาสตร์ เข้าช่วยอธิบายลักษณะของคลื่นเสียง

#### 2.1.1 อวัยวะที่ใช้ในการออกเสียง

อวัยวะที่ใช้ในการออกเสียงพูดมีอยู่หลายส่วน แต่ละส่วนสามารถทำให้เสียงพูดแตกต่างกันไป อวัยวะเหล่านี้ได้แก่ ปากและส่วนต่างๆ ในปาก ช่องคอ กล่องเสียง ช่องปากและช่องจมูก อวัยวะที่ใช้ในการออกเสียงแบ่งได้เป็น 2 ชนิด ได้แก่

1. อวัยวะที่ใช้ในการกระทำอาการ (Articulator) หมายถึง อวัยวะส่วนที่เคลื่อนไหว เพื่อผลัดลมไปยังส่วนต่างๆ อวัยวะตัวกระทำอาการที่สำคัญ คือ ลิ้น ซึ่งเป็นส่วนที่เคลื่อนไหวได้มากที่สุด
2. อวัยวะที่เป็นตำแหน่งที่เกิดเสียงต่างๆ (Point of Articulator) หมายถึง ตำแหน่งหรือฐานที่เกิดของเสียงต่างๆ เช่น ริมฝีปาก ฟัน เพดานปาก เป็นต้น

#### 2.1.2 การเกิดของเสียง(Speech Production)

ขั้นตอนการเกิดของเสียงแบ่งได้เป็น 3 ขั้นตอนใหญ่ๆ คือ

1. ขั้นเริ่มต้น (Initiation) - ขั้นตอนนี้เป็นขั้นตอนที่ลมเริ่มถูกขับออกจากปอดเพื่อเข้าสู่ขั้นตอนที่สองต่อไป

2. ขั้นตอนคัดแปลงลมที่เสียนเสียง (Phonetic) - เป็นขั้นตอนที่ลมจากปอดจะผ่านมายังหลอดลม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือที่สงวนลิขสิทธิ์ไว้เพื่อการค้าเท่านั้น เมื่อผู้ใดเห็นประโยชน์ในการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และกล่องเสียง ซึ่งที่กล่องเสียงนี้ เส้นเสียงจะทำหน้าที่เป็นลิ้นเปิด-ปิด ทำให้เกิดเสียงได้สองชนิด คือ เสียงก้อง (Voice Sounds) และเสียงที่ไม่ก้อง (Unvoiced Sounds) อวัยวะที่ใช้ในตอนนี้เป็นคือ ส่วนที่ต่อจากปอดขึ้นมาจนถึงริมฝีปาก

3. ขั้นตอนเปลี่ยนแปลงลักษณะเสียง (Articulation) - ในขั้นตอนนี้ลมที่ผ่านออกมาจากกล่องเสียงจะถูก แปลงให้เกิดเสียงในลักษณะต่างๆ อวัยวะที่ใช้ในขั้นตอนนี้คือส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก เสียงที่เกิดจากขั้นตอนนี้ดังกล่าว จะถูกแบ่งเป็น 2 ประเภทใหญ่ๆ คือ

1. เสียงก้อง (Voice Sounds) เกิดเนื่องจากการออกเสียงในลักษณะที่เส้นเสียงถูกดึงเข้ามาใกล้กันจนเกือบปิดช่องทางลมเสียดสีกัน ลมที่ดันขึ้นมาจากปอดจะทำให้เส้นเสียงสั่น ลมที่ออกมาไม่สะดวกเพราะต้องบีบตัวผ่านช่องแคบเป็นจังหวะ จึงทำให้เกิดเป็นเสียงขึ้นเรียกว่า “ เสียงก้อง “

2. เสียงไม่ก้อง (Unvoiced Sounds) เป็นการออกเสียงในลักษณะที่เส้นเสียงยังเปิดกว้างช่องระหว่างเส้นเสียงหรือช่องคอหอย (Glottis) ให้ลมหายใจผ่านเข้าออกสะดวกเรียกเสียงเหล่านี้ว่า “เสียง ไม่ก้อง”

โดยรวมแล้วระบบที่ทำแล้วทำให้เกิดเสียงขึ้นมา แบ่งได้เป็น 2 ขั้นตอนคือ ส่วนที่ทำให้การกระตุ้นให้เกิดเสียง (Excitation) และ ส่วนที่ทำให้เกิดการผสมของเสียง (Modulation) ดังแสดงในรูปที่ 2.1



รูปที่ 2.1 รูปแบบของการกระตุ้น-การผสม เพื่อกำเนิดเสียงพูด

การกระตุ้นจะเกิดได้จากหลายอย่าง เช่น เกิดจากการอัด การสั่นสะเทือน โดยการกระตุ้นจะเกิดที่ช่องคอหอยเป็นส่วนใหญ่ แต่ก็สามารถเกิดได้เล็กน้อยที่ส่วนอื่น ส่วนการผสมหรือการมอดูเลชันนั้นจะถูกกระทำโดยอวัยวะต่างๆของเส้นเสียง เสียงจะถูกมอดูเลทโดยการเคลื่อนที่ของอวัยวะต่างๆ ส่วนใหญ่เกิดที่ลิ้นและจะเปลี่ยนคุณสมบัติของเสียงไป การมอดูเลชันเปรียบเสมือนการกรองเสียง (filtering) เสียงที่ยังไม่ถูกมอดูเลชันจะมีฮาร์โมนิคมากมาย แต่เมื่อถูกมอดูเลทแล้วจะมีเพียงความถี่ธรรมชาติซึ่งมีผลต่อรูปเสียง การรีโซแนนซ์ที่ความถี่ธรรมชาตินี้ เรียกว่า ฟอร์แมนท์ (Formants)

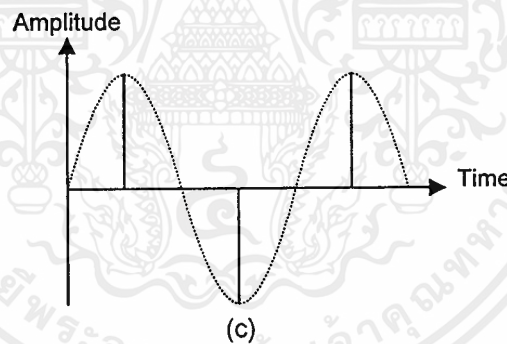
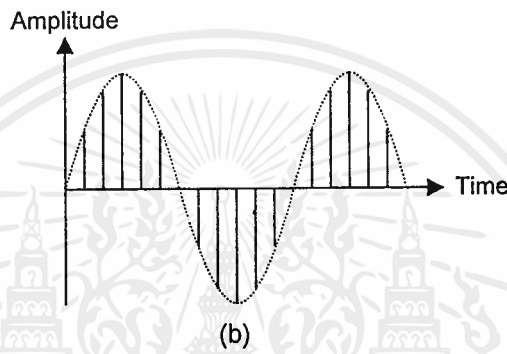
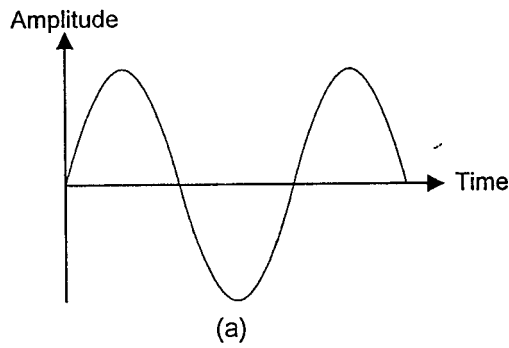
2.2 ทฤษฎีการสุ่ม (Sampling Theory)

ในความเป็นจริงนั้น ระบบที่ต่อเนื่อง อินพุท เอาท์พุท และตัวแปรอื่นๆ ที่เกี่ยวข้องในระบบทั้งหมด จะมีค่าที่เปลี่ยนแปลงไปตามเวลาแบบต่อเนื่อง ซึ่งเราสามารถที่จะหาค่าต่างๆโดยการวัดได้ ณ เวลาใดๆ แต่ไว้ในระบบแบบไม่ต่อเนื่องที่มีช่วงเวลาการทำงานเป็นช่วงๆ ที่แน่นอนนั้น เราอาจสามารถหาค่าต่างๆ ในช่วงเวลาใด

เวลาหนึ่งได้เป็นเพียงค่าที่ใกล้เคียงกับค่าจริงๆ ในระบบต่อเนื่องเท่านั้น เพราะถ้าจะให้มันมีค่าเท่ากับในระบบต่อเนื่องได้ ก็ต่อเมื่อมีการทำงาน ในลักษณะที่มีช่วงเวลางานที่สั้นมากจนใกล้เคียงค่าศูนย์ ซึ่งหมายถึงข้อมูลที่ใช้ต้องมีค่าเป็นอนันต์ด้วย เพราะฉะนั้นถ้าเราจะนำเอาเครื่องคอมพิวเตอร์ ซึ่งมีหลักการทำงานแบบไม่ต่อเนื่อง มาใช้กับงานแบบระบบต่อเนื่องจึงเป็นการยากลำบากและไม่เหมาะ ดังนั้นจึงมีความจำเป็นที่สัญญาณของระบบต่อเนื่องนั้น จำต้องถูกทำการสุ่ม (sampling) ตามช่วงเวลาแบบไม่ต่อเนื่องนั้นๆ โดยการสุ่มสัญญาณข้อมูล และทำการเปลี่ยนแปลงสัญญาณทางอนาลอก ให้เป็นข้อมูลเชิงตัวเลขหรือดิจิทัลนั้น เราเรียกว่าการดิจิไทซ์ จากที่กล่าวถึงทั้งหมดนี้ นั่น เรา รู้จักกันในรูปแบบของการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัล ส่วนในทางตรงข้ามกับที่กล่าวถึงมา นี้คือการแปลงสัญญาณดิจิทัลให้เป็นอนาลอก หรือการอนาไลซ์นั่นเอง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 (a) แสดงสัญญาณอนาล็อก

(b) การแปลงสัญญาณอนาล็อกในรูป (a) ให้เป็นสัญญาณดิจิทัล

(c) สัญญาณในรูป (a) ถูกสุ่มด้วยอัตราความเร็วในควิสต์

สัญญาณที่ถูกนำเข้ามาผ่านระบบทำการสุ่มบางทีก็มีความหมายเป็นลำดับ (sequence) แล้วแต่ว่าจะเป็นลำดับเข้าหรือลำดับออก โดยธรรมชาติแล้วสัญญาณใดๆ นั้นจะมีความยาวเป็นอนันต์ ดังนั้นเราจึงต้องใช้เวกเตอร์ (vector) เป็นตัวชี้แยกข้อมูลให้มีความยาวที่แน่นอน ซึ่งจะทำให้การวิเคราะห์นั้น สามารถทำได้ง่ายขึ้น โดยการคำนวณด้วยระบบดิจิทัล

จากทฤษฎีดังกล่าว ทำให้เราสามารถอ้างได้ว่า ข้อมูลเชิงตัวเลขที่เราสุ่มมาเพื่อเป็นตัวแทน สัญญาณทางอนาลอกต่าง ๆ นั้น ย่อมมีจุดที่สุดแน่นอน หรือเราสามารถใช้อัตราสุ่มข้อมูลเชิงตัวเลขต่าง ๆ ทางอนาลอกได้ เสมอและมีจำนวนที่แน่นอน เราเรียกความแตกต่างระหว่างข้อมูลทางตัวเลขที่เรากำหนด ได้นี้ว่าข้อมูลทางควอน ไตซ์(quantize)

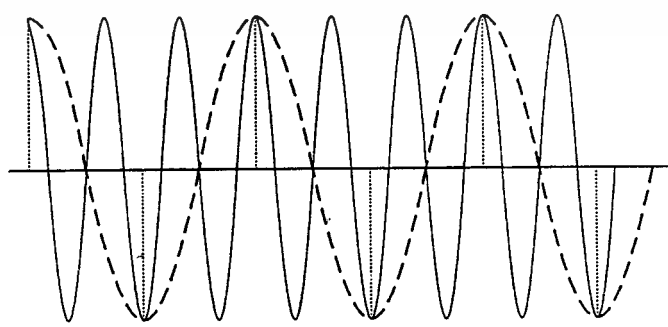
การสุ่มข้อมูลแบบนี้อาจจะเกิดความคลาดเคลื่อนอยู่บ้าง เราเรียกว่าความผิดพลาดทางควอนไตซ์ (quantize error) ซึ่งความผิดพลาดทางควอนไตซ์ นี้เป็นปัญหาในเรื่องของสัญญาณรบกวน ซึ่งเป็นเรื่องปกติ สำหรับการทำงานในระบบดิจิทัล และมักก่อให้เกิดความไม่เที่ยงตรงต่อระบบที่ทำงานเกี่ยวกับสัญญาณดิจิทัล

ในระบบที่เรากำลังศึกษานี้ จะใช้ช่วงเวลาในการสุ่มข้อมูลแต่ละครั้งเท่ากันเสมอ ดังนั้นอัตราการสุ่ม ข้อมูลของระบบจะมีค่าเท่ากับส่วนกลับของเวลาที่แตกต่างกันระหว่างการสุ่มข้อมูลแต่ละครั้ง

ในการสุ่มข้อมูลจากสัญญาณอนาลอกใดๆ ควรจะประมาณอัตราการสุ่มไว้ให้สูงพอสมควร เพราะถ้า ใช้อัตราสุ่มที่ค่อนข้างต่ำจนเกินไป อาจจะก่อให้เกิดปัญหาในระหว่างที่ทำการสุ่มข้อมูลจากสัญญาณต่อเนื่องขึ้น ได้ เพราะในสัญญาณนั้นๆ อาจมีข้อมูลที่สำคัญมาก และอาจก่อให้เกิดการเหลื่อมล้ำของข้อมูลที่ทำการสุ่มขึ้นมา ได้ ผลที่ได้คือเมื่อทำการแปลงข้อมูลที่สุ่มมาได้กลับไปในรูปแบบของสัญญาณทางอนาลอกแล้ว สัญญาณที่ได้จะมีความผิดเพี้ยนจากสัญญาณต้นฉบับเดิมมาก แต่ถ้าเราใช้ อัตราสุ่มที่สูงเกินไป ซึ่งอาจจะมากเกินไปจนเกินไป จะทำให้เกิดการสิ้นเปลืองเนื้อที่ของหน่วยความจำโดยใช่เหตุ จะเห็นได้ว่าการสุ่มที่มากเกินไปก็ไม่ใช่ว่าผลที่ดีนัก

ในทางทฤษฎีการสุ่มแล้วนั้น การสุ่มข้อมูลแบบต่อเนื่อง กำหนดให้มีการใช้อัตราการสุ่มอย่างน้อยที่สุด ที่ยอมรับได้นี้ว่า อัตราไนควิสต์ ( Nyquist rate ) โดยถ้าเราต้องการรายละเอียดของสัญญาณสูง ก็จะต้องเลือก อัตราการสุ่มที่มากกว่า 2 เท่าขึ้นไปของความถี่ของสัญญาณที่จะนำมาทำการสุ่ม เช่นถ้าสัญญาณที่จะนำมาทำการสุ่มมีความถี่ของสัญญาณเท่ากับ 3-4 กิโลเฮิรต์ อัตราการสุ่มที่เหมาะสมที่สุดจะเท่ากับ 8-9 กิโลเฮิรต์

ในระบบที่มีการสุ่มน้อยกว่าอัตราไนควิสต์นั้นเราเรียกว่า อัตราสุ่มที่น้อยเกินไป(Undersampling ) สำหรับสัญญาณที่ถูกสุ่มแบบอัตราสุ่มที่น้อยเกินไปนั้นจะได้ผลลัพธ์ที่มีความถี่เท่ากับผลต่างของความถี่ของ อัตราการสุ่ม กับความถี่ของสัญญาณที่ป้อนเข้ามา ซึ่งจะมีความผิดเพี้ยนมาก



รูปที่ 2.3 แสดงตัวอย่างของการกำหนดอัตราการสุ่มที่ต่ำกว่ามาตรฐาน

เส็นหนา แสดงถึง สัญญาณอินพุตที่มีความถี่ประมาณ 795 เฮิรต์

เส้นตรงที่บ แสดงถึง การสุ่มที่ความถี่ประมาณ 1 กิโลเฮิร์ต  
 เส้นประ แสดงถึง สัญญาณเอาต์พุตที่ได้ออกมาที่มีความถี่เท่ากับผล  
 ต่างของอัตราสุ่มกับสัญญาณอินพุต มีค่าประมาณ  
 1000-795=205 เฮิร์ต

ส่วนในระบบที่อัตราสุ่มที่มากจนเกินไป (Oversampling) นั้นอาจมาเกินความจำเป็น ทำให้มีการสิ้นเปลืองของหน่วยความจำมาก และยังลำบากต่อการนำข้อมูลที่ได้ไปวิเคราะห์เปรียบเทียบอีกด้วย

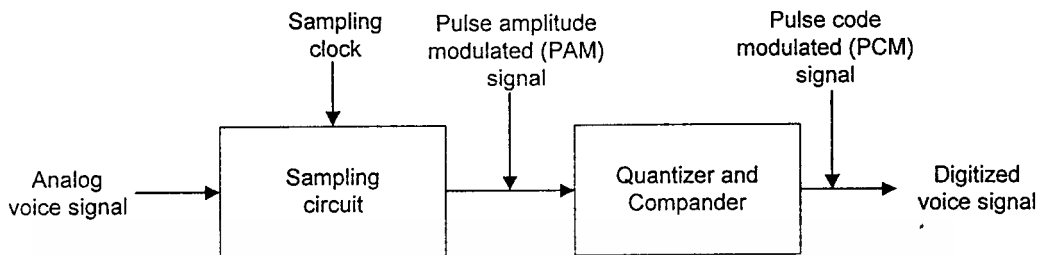
วิธีการแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลนั้นมียุหลายวิธีการ แต่โดยหลักวิธีแล้วมีหลักการในรูปแบบการทำงานที่ค่อนข้างคล้ายกัน โดยหลักการพื้นฐานและยังเป็นที่ยอมรับใช้กันอยู่คือ แบบพัลส์โค้ดมอดูเลชัน(Pulse Code Modulation : PCM)

### 2.3 การเข้ารหัสแบบพัลส์โค้ดมอดูเลชัน

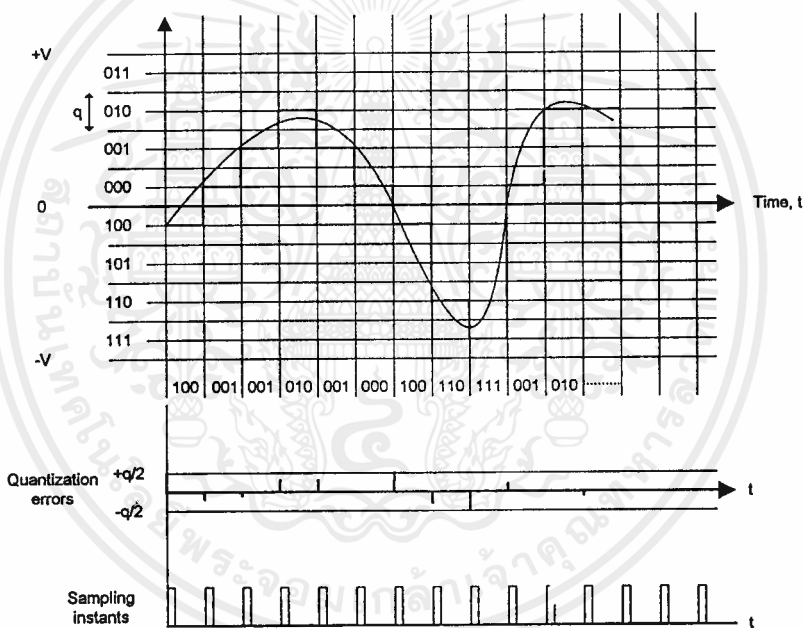
พัลส์โค้ดมอดูเลชัน คือวิธีการที่ใช้โดยทั่วไปในการแทนสัญญาณอนาล็อกให้อยู่ในรูปสัญญาณแบบดิจิทัล สำหรับจุดใดๆ บนสัญญาณอนาล็อกที่ต่อเนื่องกันจะถูกแทนได้เป็นข้อมูลทางดิจิทัลค่าหนึ่งในระบบเลขฐานสอง เมื่อใดที่ค่าหนึ่งบนสัญญาณอนาล็อกถูกทำการวัดค่าหนึ่งๆ จะถูกนำมาเปรียบเทียบว่าจะต้องแทนด้วยเลขไค้คอะไรในเลขฐานสอง ซึ่งเรียกว่า พัลส์โค้ด ถ้าหากว่าค่านั้นอยู่ในช่วงระหว่างสองค่าในเลขฐานสองจะถือว่าค่าเลขฐานสองที่ใกล้เคียงกันมากที่สุดจะเป็น ไค้คสำหรับแทนสัญญาณอนาล็อกที่เข้าไป วิธีการเหล่านี้เรียกว่า ควอนไทซ์เซชัน (Quantization) การแบ่งคลื่นออกเป็นช่วงย่อยๆ โดยแต่ละช่วงจะถูกกำหนดให้เท่ากับเลขฐานสองค่าหนึ่ง ดังนั้นในการวิเคราะห์สัญญาณเสียงใดๆ เมื่อสัญญาณเสียงผ่านกระบวนการเหล่านี้ชุดข้อมูลที่เป็นพัลส์โค้ดจะถูกส่งออกมาอย่างต่อเนื่อง ผลที่ได้คือสัญญาณพัลส์โค้ดมอดูเลชัน แต่เพราะสัญญาณเสียงที่จะนำมาวิเคราะห์ และทำการบันทึกไว้นั้นจะถูกนำมาเก็บในรูปของสัญญาณดิจิทัล ดังนั้นการที่เอาสัญญาณเสียงนั้นๆ มาผ่านกระบวนการต่างๆ เราจึงต้องทำการปรุงแต่งสัญญาณนั้นๆ ให้เหมาะสมเสียก่อน เช่นการนำสัญญาณนั้นไปผ่านวงจรกรองความถี่ต่ำผ่าน ซึ่งเป็นวงจรที่ใช้ในการกรองความถี่ข้อมูลที่มีระดับสูงจนเกินไปเกินกว่าค่าที่เราจะสนใจ หรือเกินกว่าระดับของการใช้งานโดยปกติ ทำให้ผลต่างระหว่างระดับสัญญาณ 2 จุดใดๆ ในขณะที่ทำการสุ่มมีค่ามากเกินกว่าค่าสูงสุดของพัลส์โค้ดที่เราที่กำหนดไว้ มักทำให้เกิดการผิดเพี้ยนของสัญญาณ (Compliance error) จึงต้องผ่านวงจรกรองความถี่ต่ำผ่านเพื่อลดปัญหา ยกตัวอย่าง เช่น ระดับความถี่ที่เสียงคนเราปกติก็มีค่าอยู่ในช่วง 3-4 กิโลเฮิร์ต แต่ในบางครั้งความถี่ที่เราพูดอาจจะมีความถี่ถึง 5 กิโลเฮิร์ต เกิดขึ้นได้บ้าง ดังนั้นเมื่อความถี่ข้อมูลถูกส่งเข้ามาผ่านวงจรกรองความถี่ต่ำผ่าน ส่วนของข้อมูลที่มีความถี่เกินกว่า 4 กิโลเฮิร์ต ก็จะถูกทำการตรวจสอบ(detect) เอาไว้ได้ ส่วนข้อมูลที่มีความถี่ต่ำกว่าก็จะสามารถผ่านไป

ขณะที่ทำการสุ่มสัญญาณที่มีค่าความถี่มากกว่าค่าสูงสุดของ พัลส์โค้ดที่เราได้กำหนดไว้ ซึ่งมักทำให้เกิดความผิดเพี้ยนอันเนื่องมาจากแถบคลื่นซ้อนกัน (Aliasing Distortion) ขณะที่มีการทำให้สัญญาณเดิมนั้นกลับคืนมา แต่อย่างไรก็ตามการผ่านตัวกรองความถี่ต่ำผ่านทางด้านรับนั้น ในทางปฏิบัติไม่สามารถที่จะกำจัดความถี่ซึ่งสูงกว่าค่าสูงสุดของพัลส์โค้ดได้ และฮาร์โมนิคต่างๆ ดังกล่าวเหล่านี้จะสอดแทรกเข้าไปปะปนอยู่กับ

สัญญาณที่ได้จากการแปลงกลับ หรือ ดิมอดูเลท (Demodulate) จึงเป็นผลทำให้เกิดการผิดเพี้ยนที่เรียกว่าการผิดเพี้ยนอันเนื่องมาจากการสอดแทรก (Interpolating Distortion)



รูปที่ 2.4 แสดงบล็อกไดอะแกรมของการสังเคราะห์เสียงโดยวิธีพัลส์โค้ดมอดูเลชัน



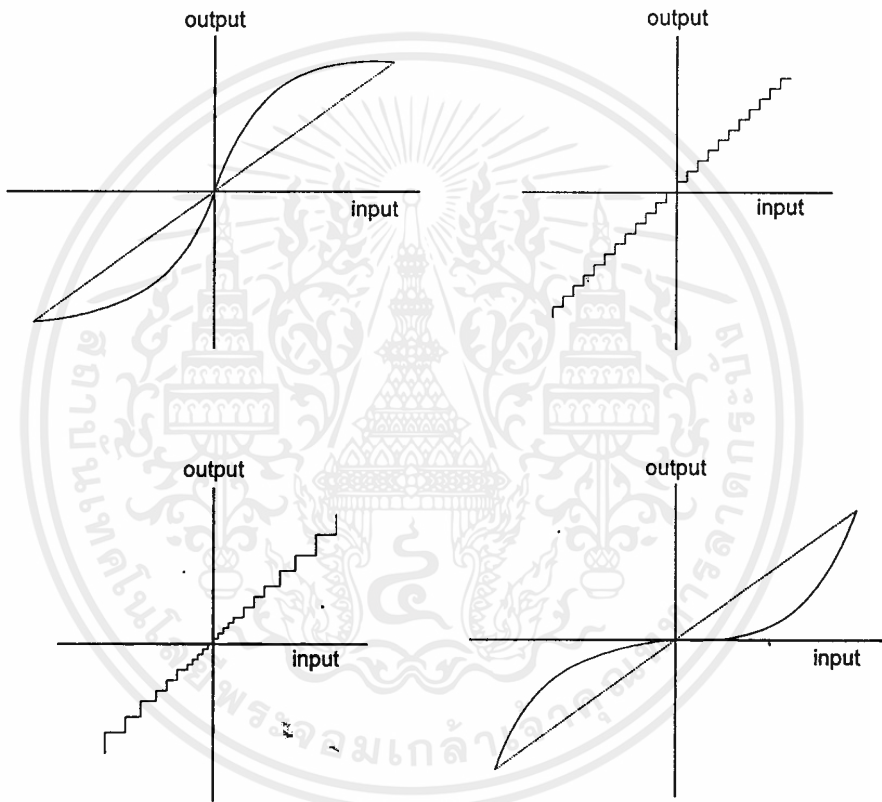
รูปที่ 2.5 การควอนไทซ์

- (a) ที่มาของความคลาดเคลื่อน(error)
- (b) ช่วงของความคลาดเคลื่อน

ส่วนการแก้ไขความผิดพลาดทางควอนไทซ์อันเนื่องมาจากสัญญาณรบกวนนั้น จะใช้วิธีการแบ่งระดับไม่เท่ากัน (Non-linear Quantizing) นั้น มีมาตรฐานหลักๆ อยู่ 2 แบบคือ แบบ U-Law และแบบ A-Law โดยทั้ง 2 ลักษณะที่กล่าวถึงนี้มีลักษณะการแบ่งแบบคอมแพนดิง (Companding) ซึ่งการแบ่งแบบนี้จะสามารถลดการรบกวนที่เกิดจากการควอนไทซ์(Quantizing noise) ที่ไม่สามารถหลีกเลี่ยงได้ โดยการลดช่วงของการควอนไทซ์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Quantizing Interval) หรือเป็นการเพิ่มจำนวนระดับนั่นเอง แต่เมื่อเพิ่มจำนวนระดับขึ้นแล้วจำนวนบิตที่ใช้ก็จะมากขึ้นด้วย จึงต้องใช้ความเร็วในการส่งสัญญาณดิจิทัลที่สูงขึ้น

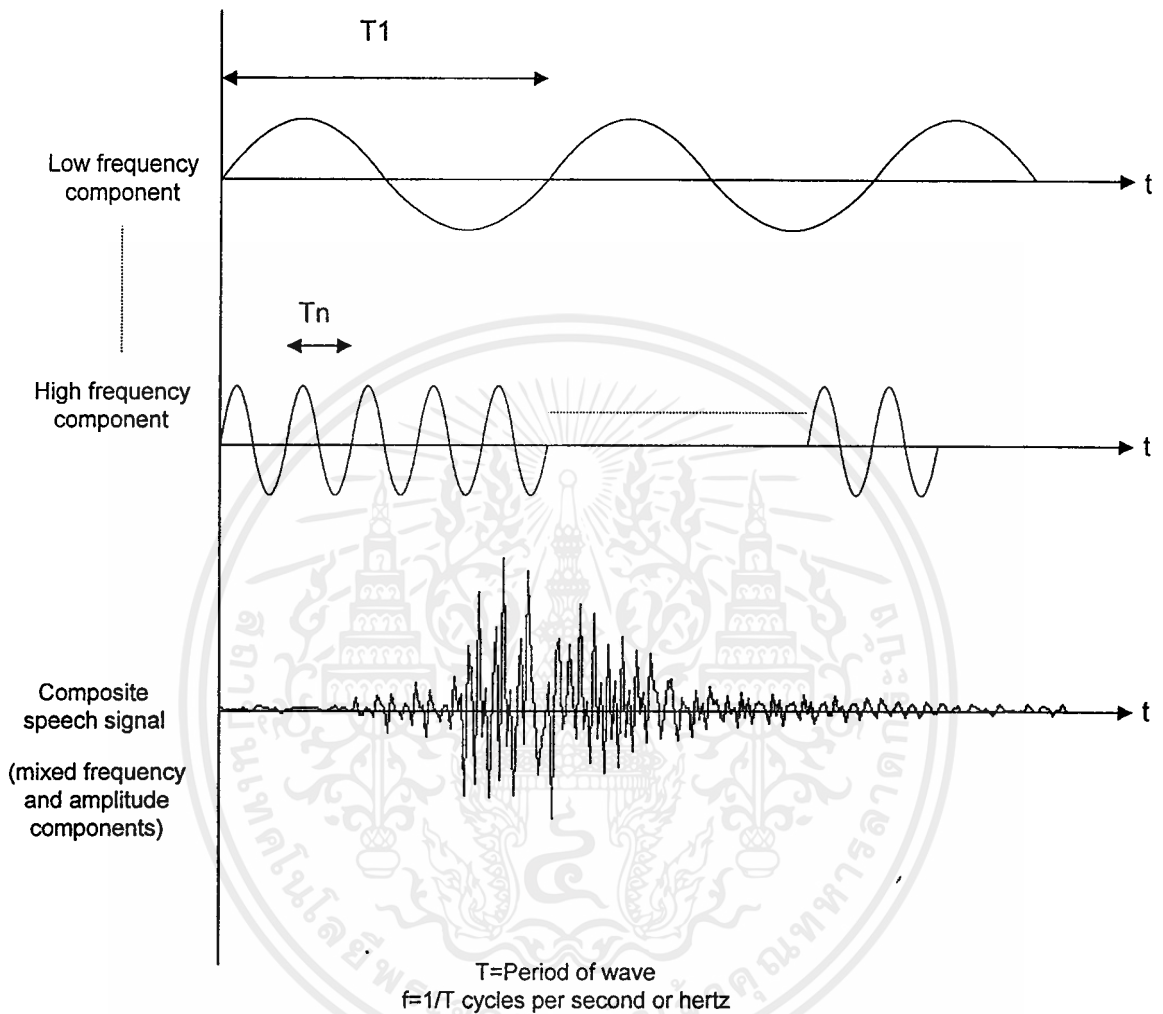
คอมแพนดิงนั้นเป็นชื่อรวมของวงจรคอมเพรสเซอร์ (Compressor) ซึ่งใช้สำหรับวงจรด้านส่ง กับวงจรเอ็กซ์แพนเดอร์ (Expander) ซึ่งใช้สำหรับด้านรับ ในบริเวณที่มีแอมพลิจูดต่ำๆ จะมีการใช้ช่วงการควอนไทซ์แคบๆ ส่วนในบริเวณที่มีสัญญาณแอมพลิจูดสูงๆ จะใช้ช่วงการควอนไทซ์กว้างๆ การแบ่งที่มีระดับไม่เท่ากันนั้น จะเรียกการแปลงสัญญาณแบบนี้ว่าอะแดปทีฟพัลส์โค้ดมอดูเลชัน (Adaptive Pulse Code Modulation : ADPCM)



รูปที่ 2.6 แสดงคุณลักษณะการจกระดับด้วยคอมเพรสเซอร์และเอ็กซ์แพนเดอร์

## 2.4 การวิเคราะห์เสียงพูด (Speech Analysis)

เสียงพูดนั้นประกอบไปด้วยความถี่หลายความถี่ดังรูป



รูปที่ 2.7 แสดงส่วนประกอบของสัญญาณเสียงพูด

ในการเก็บหรือจดจำเสียงพูด เราต้องพยายามลดช่วงของสัญญาณเสียงพูดที่เกิดซ้ำซ้อนกัน โดยใช้ตัวแทนของเสียงพูดที่สามารถแทนลักษณะสำคัญของเสียงพูดนั้นในเทอมของค่าพารามิเตอร์ (Parameter) เพื่อให้จัดการกับข้อมูลได้ง่าย

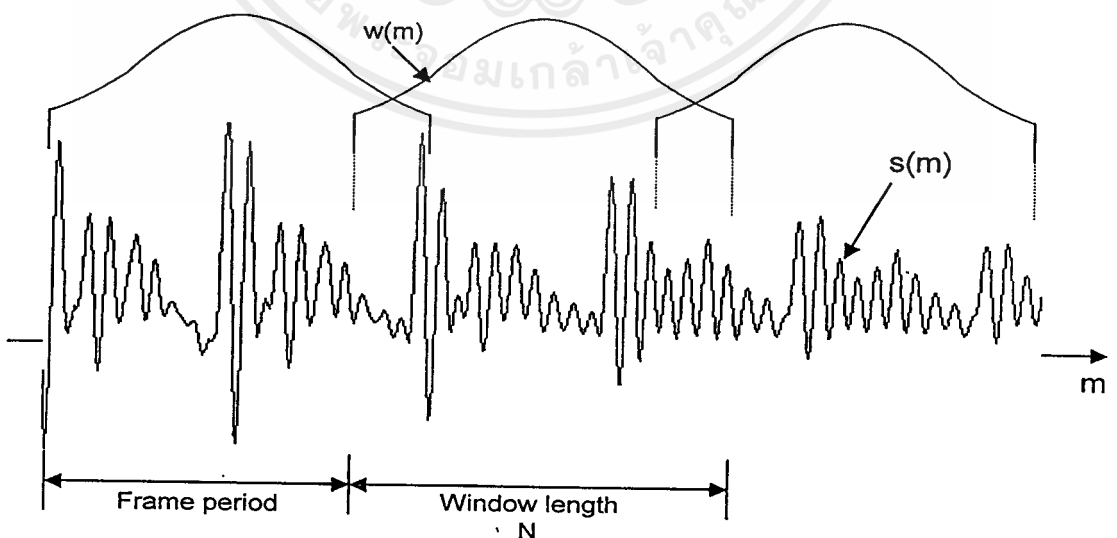
ต่อไปเราจะได้อธิบายถึงวิธีการที่ใช้ในการวิเคราะห์เสียงพูด ทั้งในโดเมนเวลา (กระทำกับเสียงพูดที่เก็บไว้โดยตรง) และ โดเมนความถี่ (ผ่านการแปลงเป็นความถี่ก่อน) เพื่อหาตัวแทนของสัญญาณเสียงพูดในรูปของพารามิเตอร์ของเสียงพูดที่เหมาะสมที่สุด สามารถนำไปใช้งานได้ และมีข่าวสารของข้อมูลครบถ้วน การวิเคราะห์ในโดเมนเวลาเราต้องการการคำนวณเพียงเล็กน้อย ทำให้ถูกจำกัดให้วัดได้ในลักษณะง่ายๆเท่านั้น เช่น เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การวัดพลังงานและความเป็นคาบเวลาของสัญญาณ ในขณะที่การวิเคราะห์ในโดเมนความถี่ จะให้ค่าพารามิเตอร์ที่มีประสิทธิภาพมากกว่า แต่การสุ่มสัญญาณเสียงพูดจะต้องมีความถี่ขงตรงสูง

เทคนิคที่ใช้ในการวิเคราะห์เสียงพูด อาจทำได้ทั้งแบบดิจิทัลและแบบอนาลอก การประมวลผลสัญญาณอนาลอกเป็นการใช้วงจรทางด้านอิเล็กทรอนิกส์ ซึ่งมีข้อดีคือที่มีความเร็วสูง แต่จำเป็นต้องใช้วงจรเฉพาะอย่าง ทำให้ต้องต่อสายใหม่และปรับค่าใหม่ทุกครั้งที่น่าไปประยุกต์ใช้กับงานอื่น ในขณะที่เทคนิคทางดิจิทัลจะสร้างและเปลี่ยนแปลงได้ง่ายกว่า เพียงแต่ต้องการซอฟต์แวร์หรือ โปรแกรม และฮาร์ดแวร์พิเศษ (ไมโคร โปรเซสเซอร์ และชิปต่างๆ) ถึงแม้ว่าอาจจะมีปัญหาในเรื่องความเร็วที่ไม่สามารถตอบสนองในเวลาจริงได้ แต่ในปัจจุบันเทคโนโลยีทางด้าน VLSI (Very Large Integrated Circuit) ได้พัฒนาไปอย่างมากทำให้ลดข้อเสียเปรียบของเทคนิคทางดิจิทัลไปได้มาก

สัญญาณเสียงพูดเป็นสัญญาณที่เปลี่ยนแปลงไปตามเวลา โดยเกิดในลักษณะแบบสุ่ม (Random) โดยต้องขึ้นอยู่กับควบคุมเสียงของผู้พูดด้วย เพราะเสียงที่เปล่งออกมาในช่วงระยะเวลาหนึ่งนั้น จะขึ้นอยู่กับรูปร่างของท่อกำทอนเสียง (Vocal Tract) และลักษณะการสั่นของเส้นเสียง (Vocal Cord) สัญญาณของเสียงพูดจึงเป็นสัญญาณที่เป็นคาบเวลาชั่วขณะ (quasi-periodic) หมายความว่าสัญญาณเสียงพูดมีคาบเวลาคงที่ในระยะเวลาสั้นๆ และมีการเปลี่ยนแปลงในระหว่างระยะเวลาสั้นๆนั้น

ถ้าเราพูดซ้ำๆ เสียงพูดที่ได้ อาจเปลี่ยนแปลงอยู่ในช่วงมากกว่า 200 มิลลิวินาทีก็ได้ แต่การพูดโดยทั่วไปลักษณะการเปลี่ยนแปลงของเสียงพูดจะอยู่ในช่วงค่าเฉลี่ยประมาณ 80 มิลลิวินาที ในการวิเคราะห์เรา จะใช้วินโดว์ (Window) วิเคราะห์สัญญาณเสียงเป็นช่วงๆ หรือเรียกว่าการวิเคราะห์เฟรม (frame) ดังรูปที่ 2.8 ช่วงเวลาของวินโดว์จะมีค่าไม่แน่นอน ในบางครั้งอาจต้องใช้ถึง 100 มิลลิวินาที ในบางครั้งเมื่อทำการวิเคราะห์สัญญาณเสียงพูดที่เร็วมากและไม่มีช่วงหยุดเลย เราอาจจะต้องใช้ช่วงเวลาของวินโดว์ต่ำถึง 5-10 มิลลิวินาที



รูปที่ 2.8 แสดงการแบ่งสัญญาณเสียงพูดออกมาเป็นเฟรม

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การใช้งานโดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย การใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 การใช้วินโดว์ (Windowing)

รูปแบบของการเฉลี่ย โดยปกติมีวิธีหลายอย่างเพื่อให้ได้เส้นของพารามิเตอร์ เป็นฟังก์ชันของเวลาแสดงได้อย่างถูกต้อง ทางเลือกทางหนึ่งที่ใช้คือ ขนาดของวินโดว์ที่เกี่ยวข้องดังนี้

1. วินโดว์จะต้องเล็กพอๆกับคุณลักษณะของคำพูด
2. วินโดว์จะต้องยาวพอที่จะใช้ในการคำนวณพารามิเตอร์
3. วินโดว์ที่ดีจะต้องไม่สั้นจนข้ามบางช่องของคำพูด การวิเคราะห์จะต้องทำเป็นคาบๆ ไป ซ้ำๆกัน

ตลอดที่ได้รับสัญญาณ

เงื่อนไขที่ข้านี้จะขึ้นกับจำนวนครั้งต่อวินาทีที่กระทำต่อสัญญาณวิเคราะห์ (frame rate) มากกว่าขนาดของวินโดว์ ปกติจะใช้จำนวนครั้งต่อวินาทีที่กระทำต่อสัญญาณวิเคราะห์ประมาณ 2 เท่าของความถี่เพื่อให้วินโดว์ทับกัน 50 %

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

วินโดว์อาจมีคาบเวลาแบบไม่จำกัด แต่ในทางปฏิบัติจะใช้แบบมีจุดสิ้นสุด เพื่อให้ง่ายในการคำนวณ โดยการใช้ฟังก์ชัน  $w(n)$  ตรวจสอบส่วนต่าง ๆ ของ  $S(n)$  โดยการเคลื่อนวินโดว์  
 วินโดว์ที่มีรูปแบบง่ายที่สุดคือรูปแบบสี่เหลี่ยม  $W(m)$  ซึ่งมีสมการดังนี้

$$W(m) = 1 \quad \text{เมื่อ } 0 \leq m \leq N-1$$

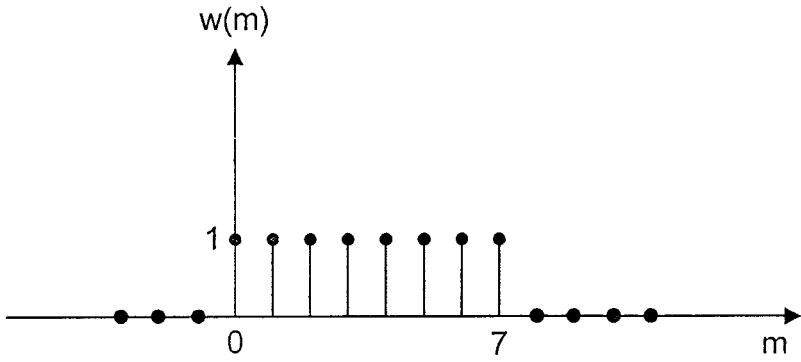
$$W(m) = 0 \quad \text{อื่นๆ} \quad \text{----- (2.1)}$$

การวิเคราะห์แบบนี้เป็นการจำกัดช่วงจุดสุ่มให้เหลือ  $N$  โดยแต่ละจุดมีการถ่วงน้ำหนักเท่าๆกัน วินโดว์ที่ใช้งานจริงนั้นต้องมีรูปร่างตามรูปที่ 2.9 สมมุติว่า เสียงพูดประมาณว่าคงที่ในช่วง 10 ms เราต้องใช้วินโดว์ที่ช่วงเวลาเท่ากับ 20 ms โดยให้จุดตรงกลางถ่วงน้ำหนักมากกว่าจุดต้นและจุดปลาย เหตุผลที่ต้องถ่วงน้ำหนักจุดกลางมากกว่าจุดปลาย เพราะรูปร่างของวินโดว์มีผลต่อเอาต์พุตต่อพารามิเตอร์ของเสียงพูด เมื่อเราเลื่อนวินโดว์ในโดเมนเวลาเพื่อวิเคราะห์เฟรมของสัญญาณเสียงพูด ผลของพารามิเตอร์อาจเปลี่ยนแปลงอย่างมากถ้าใช้ฟังก์ชัน  $W(m)$

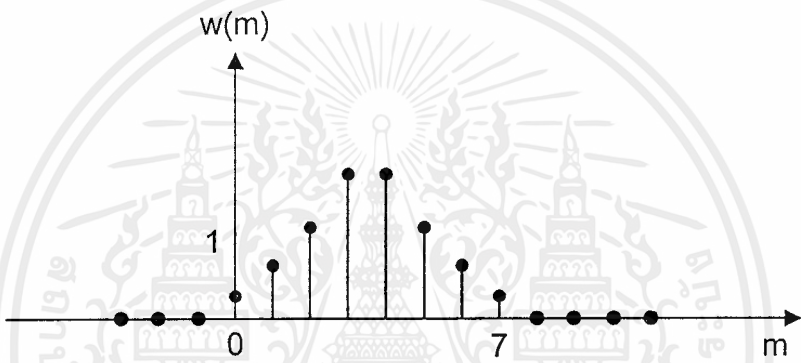
วินโดว์อีกอันหนึ่งที่นิยมใช้กันมากก็คือแฮมมิงวินโดว์ (Hamming Window) ซึ่งมีสมการดังนี้

$$W(m) = 0.54 - 0.46\cos(2\pi m/N) \quad \text{เมื่อ } 0 \leq m \leq N-1$$

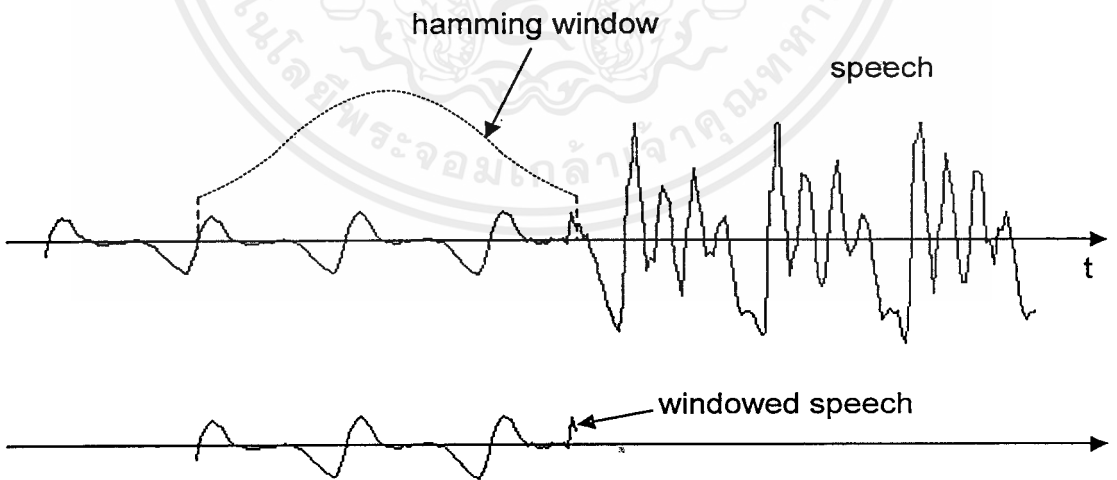
$$= 0 \quad \text{เมื่ออื่นๆ} \quad \text{----- (2.2)}$$



รูปที่ 2.9 วินโดว์รูปสี่เหลี่ยมที่มีความยาววินโดว์  $N=8$



รูปที่ 2.10 แฮมมิงวินโดว์ที่มีความยาววินโดว์  $N=8$



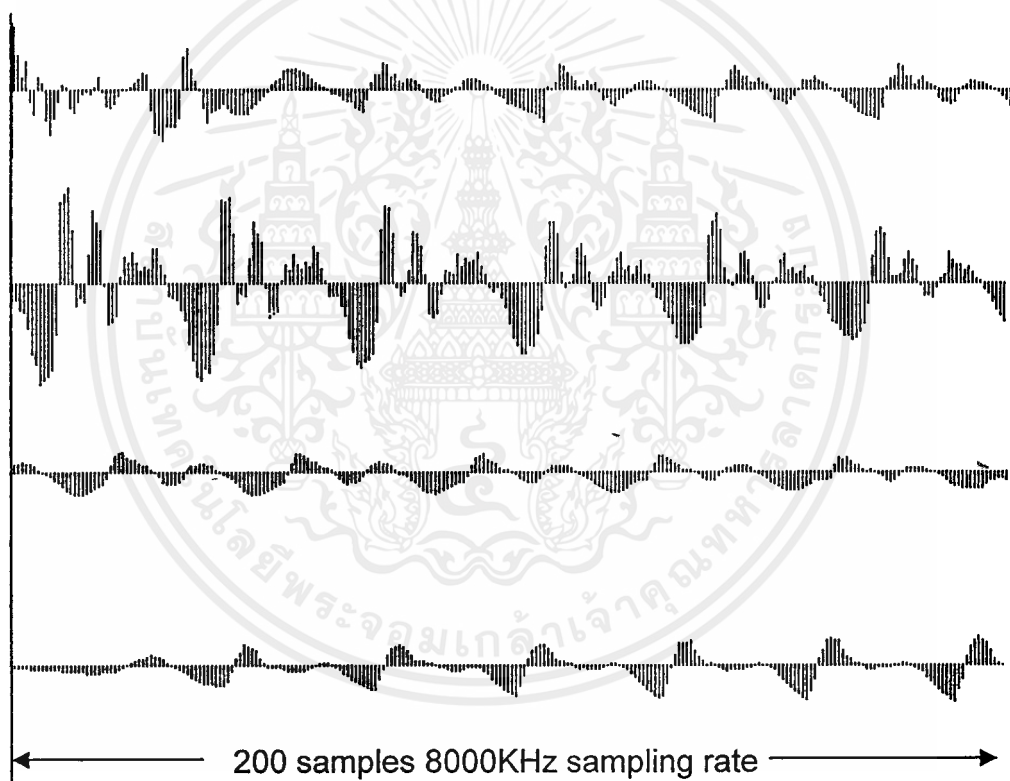
รูปที่ 2.11 การใช้แฮมมิงวินโดว์กับสัญญาณเสียงพูด

การทำให้ขอบวินโดว์ลาดลง ในการวิเคราะห์เฟรมโดยการเลื่อนตลอดแนวความยาวของสัญญาณทั้งหมด ไม่มีผลเสียต่อพารามิเตอร์เสียงพูด

## 2.4.2 การวิเคราะห์ในโดเมนเวลา

ในขบวนการของสัญญาณเสียงในฟังก์ชันเวลา จะมีความได้เปรียบในด้านความง่าย ค่าความเร็ว และง่ายในการแปลความหมาย พารามิเตอร์ของเสียงสำหรับการเข้ารหัสและการจำ สามารถทำได้จากการวิเคราะห์ฟังก์ชันเวลา เช่น การวัดค่าเฉลี่ยของอัตราตัดผ่านศูนย์ การวัดพลังงาน (amplitude) และการหาค่าฟังก์ชันอโตคอร์เรชัน (Autocorrelation function)

สัญญาณเสียงที่ถูกสุ่มด้วยอัตราสุ่ม 8000 ครั้งต่อวินาที แสดงดังรูป ที่ 2.12 ซึ่งแสดงให้เห็นว่าคุณสมบัติของเสียงนั้นเปลี่ยนแปลงไปตามเวลา ตัวอย่างเช่นการเปลี่ยนแปลงระหว่างสัญญาณเสียงก้องและเสียงไม่ก้อง ซึ่งมีขนาดแอมพลิจูดแตกต่างกันในส่วนของสัญญาณเสียงก้อง ซึ่งตัวแปรเหล่านี้สามารถหาได้จากการวิเคราะห์ในโดเมนเวลา



รูปที่ 2.12 ตัวอย่างการสุ่มสัญญาณเสียง(ด้วยอัตราสุ่ม 8 กิโลเฮิร์ต)

ถ้าสมมุติว่าสัญญาณเสียงเปลี่ยนแปลงไปอย่างช้าๆ ตามเวลา และเราตัดเอาช่วงหนึ่งของเสียงนั้นมาวิเคราะห์ เราจะเรียกว่า การวิเคราะห์ในโดเมนเวลาช่วงสั้นๆ (short-time analysis) โดยที่ส่วนที่ถูกตัดช่วงสั้นๆ นั้น เชื่อมต่อกัน โดยมีบางส่วนที่เหลื่อมซ้อนกัน ผลของการวิเคราะห์แต่ละส่วนซึ่งเราเรียกว่าเฟรม (analysis frame) นั้นอาจจะมีค่าเป็นตัวเลขค่าเดียวหรือหลายค่า

การหาค่าพารามิเตอร์ใน โดเมนเวลานั้นมีการทำได้หลายวิธี แต่จะนำเสนอเป็นตัวอย่างบางวิธีเท่านั้น ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.4.2.1 การหาพลังงานช่วงสั้นๆ และค่าเฉลี่ยของขนาดเสียงพูด

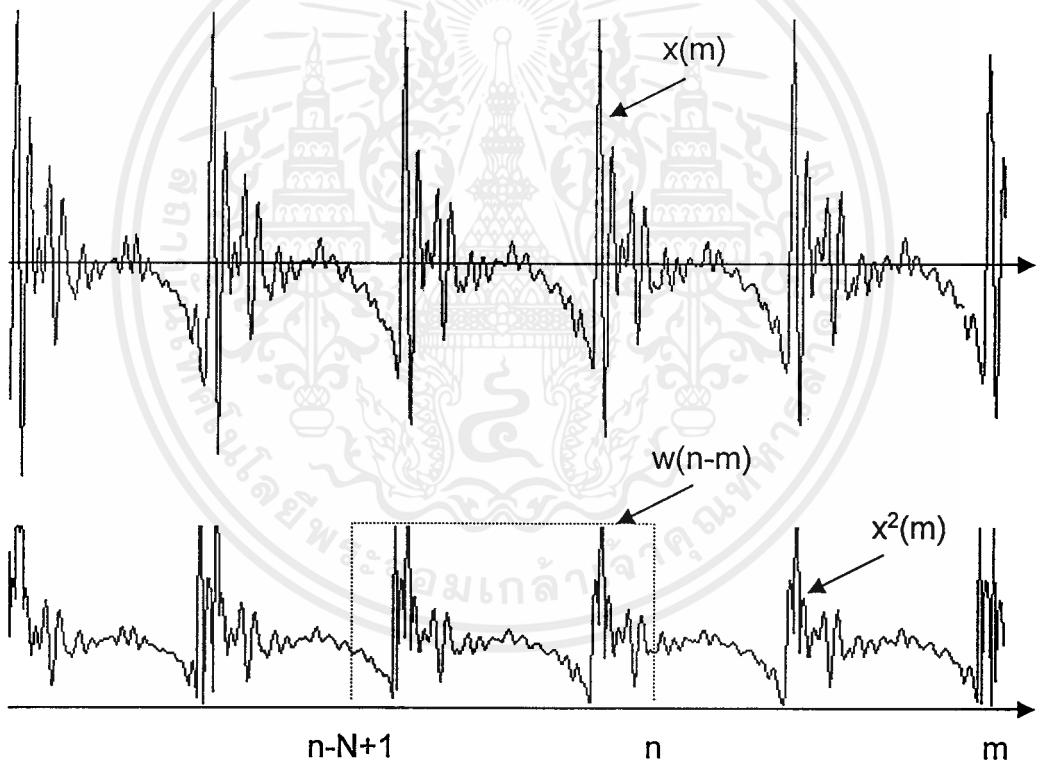
การหาค่าพลังงานของเสียงใน โดเมนเวลาช่วงสั้นๆ นั้น ก็ใช้ลักษณะของเฟรมดังกล่าวข้างต้น โดยที่พลังงานของเสียงใน โดเมนเวลาแบบแยก (discrete time) มีนิยามดังนี้

$$E = \sum_{m=-\infty}^{\infty} x^2(m) \quad \text{-----}(2.3)$$

และพลังงานของเสียงในโดเมนเวลาช่วงสั้นคือ

$$E_n = \sum_{m=n-N+1}^n x^2(m) \quad \text{-----}(2.4)$$

ซึ่งหมายถึงพลังงานช่วงสั้นๆ ที่ตัวอย่างการสุ่มที่  $n$  คือ ผลรวมของค่ายกกำลังสองของ  $N$  แซมเปิล (ตั้งแต่  $n-N+1$  ถึง  $n$ )



รูปที่ 2.13 แสดงการคำนวณพลังงานช่วงสั้น

เราจะสังเกตเห็นว่าแอมพลิจูดของสัญญาณเสียงจะเปลี่ยนไปตามเวลา ในกรณีของแอมพลิจูดของเสียงไม่ก้อง โดยทั่วไปจะต่ำกว่าแอมพลิจูดของเสียงก้อง

พลังงานช่วงสั้นๆ ของสัญญาณเสียงเป็นตัวแสดงถึงผลกระทบต่อ การเปลี่ยนแปลงของแอมพลิจูด

$$E_n = \sum_{m=-\infty}^{\infty} [x(m)w(n-m)]^2 \quad \text{-----}(2.5)$$

สามารถเขียนสมการเป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$E_n = \sum_{m=-\infty}^{\infty} x^2(m).h(n-m) \quad \text{----- (2.5)}$$

โดยที่

$$h(n) = w^2(n) \quad \text{----- (2.6)}$$

สมการที่ 2.4 สามารถเขียนแสดงดังรูปที่ 2.13 สัญญาณ  $X^2(n)$  จะถูกกรองสัญญาณ โดยตัวกรองเชิงเส้น(linear filter) ด้วยผลตอบสนองช่วงสั้น (impulse response)  $h(n)$

วินโดว์ที่สมมูลกันจะแสดงถึงธรรมชาติของพลังงานช่วงสั้นๆ อย่างไรบ้าง เรามาพิจารณาว่า ถ้าสมมุติ impulse response ยาวมาก และมีแอมพลิจูดที่ พลังงานจะเปลี่ยนแปลงค่อนข้างน้อย เมื่อเทียบกับเวลาดังนั้นวินโดว์ควรจะสมมูลกับวงจรกรองความถี่ต่ำผ่านที่มีแถบความถี่มาก ๆ สิ่งที่ต้องออกแบบให้ชัดเจนก็คือ วงจรกรองความถี่ต่ำผ่าน ซึ่งให้เอาที่พู่ที่คงที่นั่นคือเราต้องการพลังงานช่วงสั้นๆ ที่ส่งผลต่อการเปลี่ยนแปลงขนาดของสัญญาณเสียงนั่นคือ เรากำลังเผชิญกับปัญหาซึ่งจะถูกยกขึ้นยกขึ้นมาศึกษาบ่อยๆ ในการแสดงพลังงานช่วงสั้นๆ ของสัญญาณเสียง เนื่องจากเราต้องการวินโดว์ช่วงสั้นๆ ที่มีการตอบสนองต่อการเปลี่ยนแปลงของแอมพลิจูดอย่างรวดเร็ว แต่วินโดว์ที่สั้นมากๆ จะไม่เพียงพอกับการเฉลี่ยในการสร้างฟังก์ชันพลังงานที่ค่อนข้างราบเรียบ(smooth energy function)

### 2.4.2.2 การหาอัตราตัดผ่านศูนย์เฉลี่ยช่วงสั้นๆ

สำหรับสัญญาณที่ไม่ต่อเนื่องกันทางเวลา การตัดผ่านศูนย์จะเกิดขึ้นเมื่อตัวอย่างที่สุ่มได้แต่ละตัวมีเครื่องหมายต่างกัน อัตราตัดผ่านศูนย์เป็นวิธีการง่าย ๆ ที่ใช้ในการวัดองค์ประกอบทางความถี่เสียง โดยเฉพาะสัญญาณที่มีแถบความถี่แคบ ตัวอย่างเช่น สัญญาณรูปไซน์ความถี่  $F_0$  ถูกสุ่มตัวอย่างด้วยอัตรา  $F_s$  จะได้จำนวนตัวอย่างเท่ากับ  $F_s/F_0$  ตัวอย่างคือคลื่นรูปไซน์ 1 รอบ ในแต่ละรอบจะมีการตัดผ่านศูนย์ 2 ครั้ง

อัตราเฉลี่ยในช่วงเวลายาวนานของการตัดผ่านศูนย์คือ

$$Z = 2F_0/F_s \quad \text{crossing/sampling} \quad \text{----- (2.7)}$$

ดังนั้น อัตราเฉลี่ยของการตัดผ่านศูนย์ จะเป็นวิธีที่น่าเชื่อถือวิธีหนึ่งที่ใช้ในการประมาณค่าความถี่ของคลื่นรูปไซน์

สัญญาณเสียงเป็นสัญญาณที่มีช่วงกว้างและการแสดงอัตราเฉลี่ยของการตัดผ่านศูนย์มีความละเอียดหรือความแม่นยำค่อนข้างน้อย เราจึงจำเป็นต้องใช้ช่วงที่แคบๆ โดยการใช้วินโดว์

นิยามของอัตราตัดผ่านศูนย์คือ

$$Z_n = \sum_{m=-\infty}^{\infty} |\text{sgn}[x(m)] - \text{sgn}[x(m-1)]| w(m-n) \quad \text{----- (2.8)}$$

โดยที่

$$\begin{aligned} \text{sgn}[x(n)] &= 1 && \text{เมื่อ } x(n) \geq 0 \\ &= -1 && \text{เมื่อ } x(n) < 0 \end{aligned} \quad \text{----- (2.9)}$$

และ

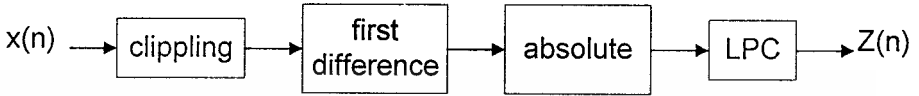


$$w(n) = \frac{1}{2n}$$

$$w(n) = \frac{1}{2n} \quad \text{เมื่อ} \quad 0 \leq n \leq n-1$$

$$= 0 \quad \text{อื่นๆ} \quad \text{-----} \quad (2.10)$$

สมการของอัตราตัดผ่านศูนย์แสดงดังรูป 2.14



รูปที่ 2.14 บล็อกไดอะแกรมแสดงการหาค่า Zero Crossing

อัตราตัดผ่านศูนย์เฉลี่ยช่วงสั้นๆ จะมีคุณสมบัติทั่วไปเหมือนกันกับค่าพลังงานช่วงสั้นๆ และขนาดเฉลี่ยในช่วงสั้นๆ แต่ตามสมการดังกล่าวทำให้การหาค่าอัตราตัดผ่านศูนย์มีความยุ่งยากมากเกินความจำเป็น เราจะนำอัตราตัดผ่านศูนย์เฉลี่ยช่วงสั้นๆ มาประยุกต์ใช้กับสัญญาณเสียงพูดได้อย่างไร รูปแบบของการสร้างเสียงพูด แนะนำว่า พลังงานของเสียงก้องจะหนาแน่นในช่วงที่ต่ำกว่า 3 กิโลเฮิร์ต ในขณะที่เสียงไม่ก้อง พลังงานส่วนใหญ่จะถูกพบที่ความถี่สูงกว่า เนื่องจากความถี่ค่าสูงจะมีค่าอัตราตัดผ่านศูนย์เฉลี่ย (โดยเฉลี่ยเกินกว่า 10 mm) ของเสียงก้องและ ไม่ก้อง ค่าเฉลี่ยของอัตราตัดผ่านศูนย์เฉลี่ยช่วงสั้น สำหรับเสียง ไม่ก้อง คือ 49 ต่อ 10 mm และสำหรับเสียงก้องคือ 14 ต่อ 10 mm

### 2.4.3 ค่าพารามิเตอร์ในโดเมนความถี่

ค่าพารามิเตอร์สำคัญจำนวนมากถูกค้นพบในโดเมนความถี่ เพราะท่อกำทอนเสียงให้กำเนิดสัญญาณ ที่สามารถวิเคราะห์เป็นความถี่ได้ง่ายกว่าทำในโดเมนของเวลาโดยตรง เสียงของคนพูดเพียงคนเดียวกันในแต่ละครั้ง เมื่อพิจารณาในโดเมนของเวลาจะเห็นผลของความแตกต่างของสัญญาณ แต่เมื่อพิจารณาผ่านในโดเมนความถี่สัญญาณที่ได้จะมีความคล้ายคลึงกันมาก

ระบบการได้ยินของมนุษย์จะตอบสนองต่อรูปร่างของสัญญาณเสียง หรือขนาดที่กระจายอยู่ในโดเมนความถี่ มากกว่าที่เฟส(phase) ของสัญญาณในโดเมนเวลาด้วยเหตุผลเหล่านี้ การวิเคราะห์ความถี่จึงถูกใช้ในการดึงค่าพารามิเตอร์สำคัญจากเสียงพูด ต่อไปจะได้อธิบายถึงวิธีการต่างๆ ที่จะนำไปใช้ในการวิเคราะห์ความถี่

#### 2.4.3.1 การวิเคราะห์ด้วยฟิลเตอร์แบงด์

การวิเคราะห์ด้วยฟิลเตอร์แบงด์ เป็นเทคนิคที่นิยมกันมาก เพราะสามารถตอบสนองได้ในเวลาจริง โครงสร้างง่ายไม่แพง โดยใช้ฟิลเตอร์แบงด์หรือกลุ่มของแบนด์พาสฟิลเตอร์ (bandpass filter) ซึ่งอาจจะเป็นอนาล็อกฟิลเตอร์หรือดิจิตอลฟิลเตอร์ก็ได้ แบนด์พาสฟิลเตอร์แต่ละตัวจะแยกวิเคราะห์ความถี่ของสัญญาณเสียงในช่วงๆ ในทางการค้า วิธีฟิลเตอร์ มีความยืดหยุ่นมากกว่า การวิเคราะห์ด้วยฟูเรียร์แบบไม่ต่อเนื่อง(discret

fourier) เพราะช่วงแบนด์วิธ สามารถปรับช่วงตามขอบเขตการรับฟังเสียงของมนุษย์ได้มากกว่าการกำหนดตายตัวว่า ต้องมีแบนด์วิธกว้างแคบเท่ากับค่าใดค่าหนึ่ง

ในการใช้งานจำนวนมาก ต้องการกลุ่มของพารามิเตอร์ความถี่ไม่มากในการอธิบายการกระจายของพลังงานความถี่ โดยทั่วไปใช้กลุ่มของแบนด์พาสฟิลเตอร์ 10-12 ตัว ก็สามารถให้ค่าตัวแทนของความถี่ที่ซับซ้อน และมีประสิทธิภาพมากกว่า การวิเคราะห์ด้วยฟูเรียร์แบบไม่ต่อเนื่อง ซึ่งให้รายละเอียดปลีกย่อยของแต่ละความถี่ได้มากกว่า

ในวิธีการทั่วไป ฟิลเตอร์จะมีการวางระยะให้มีระยะห่างความถี่ของฟิลเตอร์แต่ละตัวเท่าๆกัน และมีแบนด์วิธที่เพิ่มขึ้นแบบลอการิทึม แต่ไม่ควรเกิน 1 kHz โดยทั่วไปนิยมใช้ฟิลเตอร์ 1/3 ออกเตฟ (one-third-octave filter) ในระบบการจดจำเสียงพูดจริงๆนั้น อาจใช้ฟิลเตอร์แบนด์ถึง 2 ครั้ง ครั้งแรกเป็นการจำแนกเสียงอย่างคร่าวๆ ก่อน โดยใช้ฟิลเตอร์เพียงไม่กี่ตัว และวิเคราะห์หารายละเอียดปลีกย่อยอีกครั้ง โดยใช้กลุ่มของฟิลเตอร์มากขึ้น

#### 2.4.3.2 การวิเคราะห์ฟูเรียร์ช่วงสั้น (Short-time Fourier Transform)

การวิเคราะห์ฟูเรียร์ช่วงสั้นเป็นเทคนิคการหาความถี่ที่ใช้กันมานานแล้ว การวิเคราะห์ฟูเรียร์ให้ตัวแทนของสัญญาณเสียงพูดเป็นฟังก์ชันความถี่ในเทอมของขนาดและเฟส เนื่องจากเสียงพูดไม่ได้นิ่งอยู่ตลอดเวลา ดังนั้นจึงจำเป็นต้องใช้การวิเคราะห์ในช่วงสั้น โดยใช้วินโดว์

การใช้ฟูเรียร์ช่วงสั้นมีสมการว่า

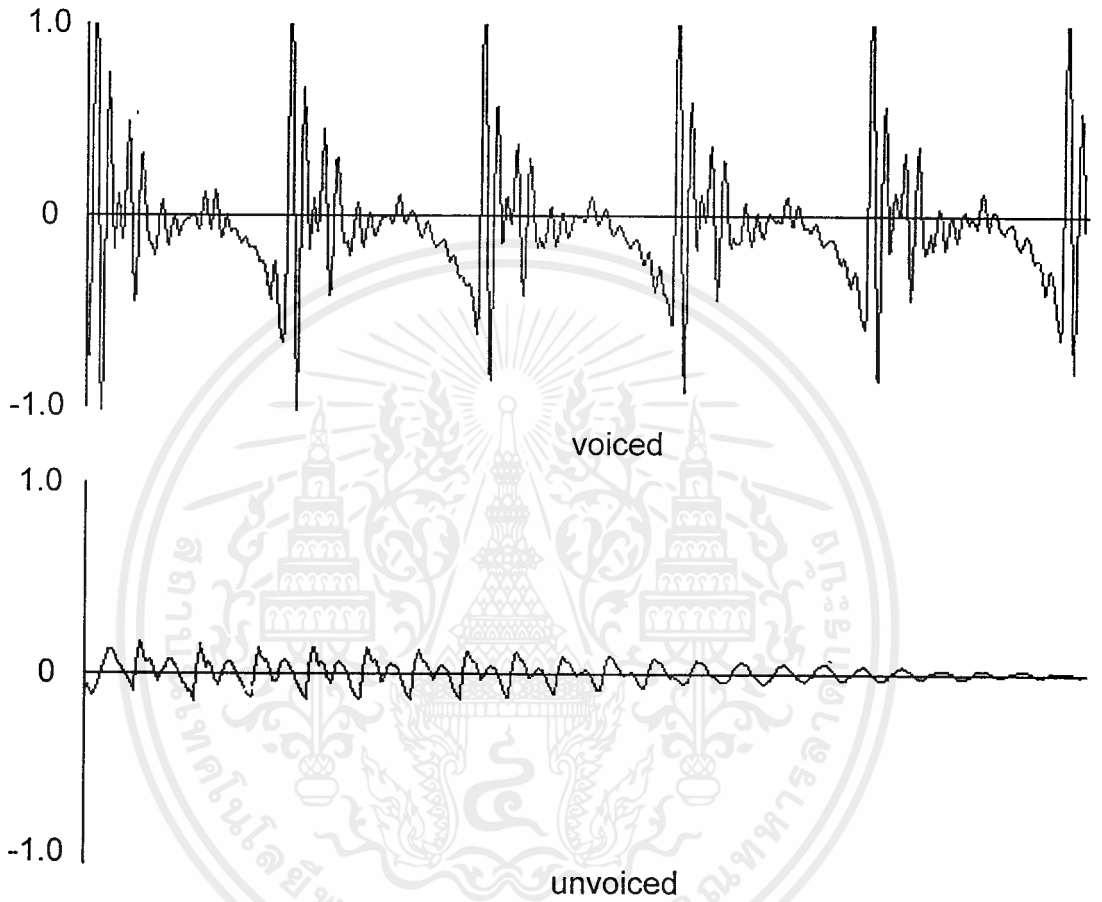
$$S_n(e^{-j\omega}) = \sum_{m=-\infty}^{m=\infty} s(m)[\text{exe}(-j\omega m)w(n-m)] \quad \text{-----}(2.11)$$

ในการคำนวณเราต้องใช้ DFT แทนการแปลงฟูเรียร์แบบต่อเนื่องโดยใช้ฟังก์ชันวินโดว์  $w$  ลดข้อมูลแบบไม่ต่อเนื่องทั้งหมดให้เหลือจำนวน  $N$  ตัว ( $N$  คือ ช่วงเวลาหรือขนาดของวินโดว์ ที่ใช้ในการแปลงฟูเรียร์แบบไม่ต่อเนื่อง) ข่าวดสารต่างๆ ในสมการที่ 2.11 จะไม่สูญหายไปจากข้อมูลเดิม ถ้าการแปลงนั้นสัมพันธ์ด้วยความถี่สูงเพียงพอ (คือช่วงระยะห่างระหว่าง  $N$ ) และวินโดว์  $w(n)$  ไม่มีจุดศูนย์กลางที่ศูนย์กลางตลอดช่วง  $N$  ตัวแปร  $N$  เป็นตัวที่ต้องระวังมากเป็นพิเศษในการวิเคราะห์ความถี่ช่วงสั้น ถ้าค่าของ  $N$  ต่ำ (ใช้วินโดว์ช่วงสั้น) จะทำให้ความละเอียดในโดเมนความถี่จะหยาบมาก เพราะจะให้ผลที่ดีในโดเมนเวลา เพราะการเฉลี่ยถูกทำในช่วงสั้นๆ เท่านั้น ในทางตรงกันข้าม ถ้า  $N$  มีขนาดใหญ่จะทำให้ผลความละเอียดในโดเมนเวลาที่ไม่ดี แต่จะทำให้โดเมนความถี่มีความละเอียดสูงกว่า

#### 2.4.3.3 การหาค่าสัมประสิทธิ์ส่วนประกอบทางเสียง

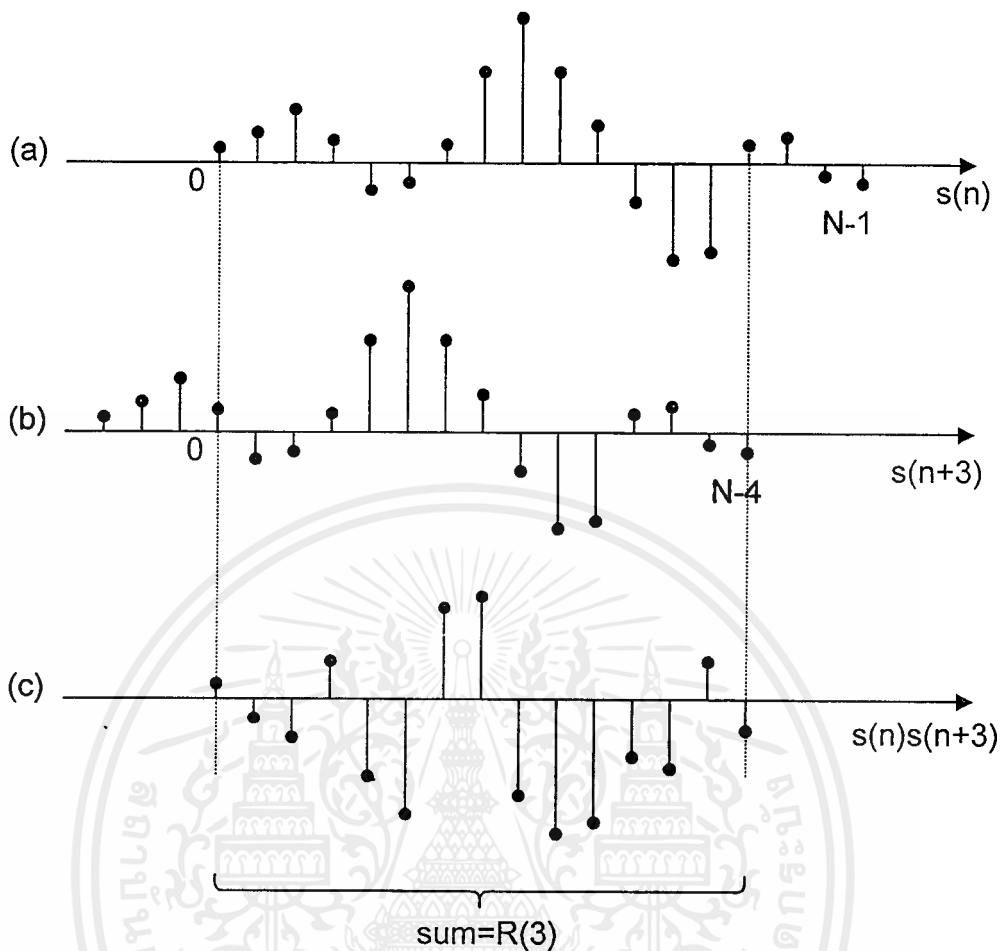
เมื่อ  $s(n)$  เป็นส่วนหนึ่งของเสียงพูดที่มีความยาว  $N$  แซมเปิล ใน 1 วินโดว์ ฟังก์ชันออโตคอร์เรชัน แสดงดังสมการที่ 2.12

$$R(k) = \sum_{n=0}^{N-1-k} s(n) s(n+k) \quad \text{เมื่อ } k=0,1,2,\dots \quad (2.12)$$



รูปที่ 2.15 แสดงตัวอย่างของฟังก์ชันออโตคอร์เรชันสำหรับเสียงก้องและเสียงไม่ก้อง

จากรูปจะสังเกตเห็นว่าฟังก์ชันออโตคอร์เรชันนี้ใช้ได้ทั้งเสียงก้องและเสียงไม่ก้อง ตัวอย่างการคำนวณหา  $R(3)$  แสดงดังรูป 2.17



รูปที่ 2.17 แสดงการคำนวณฟังก์ชัน R(3)

ฟังก์ชันออโตคอร์เรชันนั้นสามารถนำไปคำนวณหาค่าสัมประสิทธิ์พาร์คอร์ได้ ดังความสัมพันธ์ตามสมการต่อไปนี้

$$E(0) = r(0) \tag{3.13}$$

$$K_i = \frac{-r(i) + a_1^{(1-i)}r(i-1) + \dots + a_{i-1}^{(i-1)}r(1)}{E(i-1)} \quad \text{for } i = 1, \dots, p \tag{3.14}$$

$$a_i^{(i-1)} = K_i \tag{3.15}$$

$$a_j(i) = a_j^{(i-1)} + K_i a_{i-j}^{(i-1)} \quad \text{for } j = 1, \dots, i-1 \tag{3.16}$$

$$E(i) = (1 - K_i^2)E(i-1) \tag{3.17}$$

### 2.5 การตรวจจับจุดสิ้นสุด (Endpoint Detection)

ปัญหาพื้นฐานของระบบจดจำเสียงพูด คือ เสียงที่เปล่งออกมาในขณะที่ตรวจสอบนั้น อาจจะไม่เหมือนกับตอนที่เรียนรู้เสมอไป เสียงที่เปล่งออกมา 2 ครั้ง จะมีระยะเวลาที่แตกต่างกัน และระยะห่างของการออกเสียง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ไม่แน่นอน หมายความว่า คุณสมบัติที่ขึ้นกับเวลา จะทำให้ การเปรียบเทียบไม่สามารถทำได้เพราะค่าอ้างอิงกับค่าที่มาเปรียบเทียบเวลาจะต่างกัน ในกรณีเช่นนี้ ถึงแม้ว่าจะเปรียบเทียบกับค่า ที่ไม่รู้กับรูปแบบที่ถูกต้องของตัวเองแล้วก็ตาม แต่ก็อาจจะให้ผลแตกต่างได้พอๆ กับที่เปรียบเทียบกับรูปแบบอ้างอิงของค่าอื่น ซึ่งจะเป็นปัญหา ถ้าค่าที่ถูกเปรียบเทียบ เป็นการเปรียบเทียบทั้งค่าต้นแบบ แทนที่จะเปรียบเทียบกับบางส่วนของค่า โดยการเปรียบเทียบส่วนต่อส่วน ความจริงนั้นแม้ในระหว่างเสียงสองเสียง ที่ใช้ในการเรียนรู้ รายละเอียดของเวลายังต่างกัน เมื่อเก็บข้อมูลเสียงจากหลายๆ คน จึงต้องทำการเฉลี่ยในระหว่างการเรียนรู้ก่อนนำไปเปรียบเทียบ

ในหลายๆ กรณี ความแม่นยำของการปรับ จะขึ้นอยู่กับความแม่นยำของการแสดงจุดสิ้นสุดของค่า ความผิดพลาดในการจดจำจุดเริ่มต้น และจุดสิ้นสุดของค่า จะทำให้เกิดความยุ่งยากขึ้น โดยความผิดพลาดของจุดสิ้นสุด จะเกิดขึ้นในค่าที่เริ่มต้นและลงท้ายด้วยพยางค์ที่ออกเสียงเบา ผู้พูดบางคนอาจบ่นประโยคด้วยเสียงสั้น พยางค์เสียงสั้นนี้จะทำให้เกิดความผิดพลาดในการเปรียบเทียบค่าในห้องทดลอง การจดจำจุดสิ้นสุดอาจจะทำง่าย เพราะข้อมูลเสียงพูดถูกรวบรวมภายใต้การควบคุมสภาวะ เพื่อหาจุดสิ้นสุด ยิ่งกว่านั้น ผู้พูดที่บันทึกข้อมูลในห้องทดลอง จะทำด้วยความระมัดระวัง แต่นอกห้องทดลองไม่ได้เป็นเช่นนั้น เสียงพูดจะมีการรบกวนจากรอบข้างเพราะไม่มีการควบคุมสภาวะ และผู้พูดไม่มีแรงจูงใจที่จะช่วยในการจดจำ

คุณสมบัติหลักของจุดสิ้นสุด คือพลังงาน ดังนั้นวิธีง่าย ๆ ที่ใช้ในการตรวจจับจุดสิ้นสุดก็คือ การเปรียบเทียบพลังงานของสัญญาณกับค่าหรือระดับที่ตั้งไว้โดยตรวจสอบจุดเริ่มต้นของค่าจากตำแหน่งของค่าที่มีพลังงานเริ่มมากกว่าระดับที่ตั้งไว้ และจุดสิ้นสุดจากตำแหน่งของค่าที่มีพลังงานต่ำกว่าระดับที่ตั้งไว้ ค่าระดับที่ตั้งไว้นี้ต้องมีความเข้มระดับหนึ่ง ถ้าความแรงของสัญญาณเสียงพูดอ่อน อาจจะต้องตั้งระดับต่ำๆ

### บทที่ 3

#### การคำนวณและการสร้าง

จากการศึกษาการทดลองของนักวิทยาศาสตร์หลายท่านเกี่ยวกับทฤษฎีการแยกแยะเสียง ได้ข้อสรุปที่น่าสนใจคือ

1. สัญญาณที่ถูกคลิป(clipping) หรือจำกัดระดับให้เหลือเพียงสองระดับคือ 0 และ 1 โดยวิธีหาจุดตัดศูนย์กลางนั้น จะยังคงคุณสมบัติเด่นของเสียงเพียงพอที่จะใช้ในการแยกแยะและรู้จำเสียงได้
2. การกระจายของอัตราตัดผ่านศูนย์กลาง เกี่ยวข้องโดยตรงกับหน่วยเสียงหนึ่งถึงสองหน่วยเสียงแรก โดยเฉพาะหน่วยเสียงก้อง
3. ในการวิเคราะห์คุณสมบัติของเสียง โดยใช้วิธีหาจุดตัดผ่านศูนย์กลางนั้น การวิเคราะห์เสียงช่วงสั้นๆ(Window or Frame Analysis) ประมาณ 10-20 มิลลิวินาที เป็นเวลาที่แคบจนไม่สามารถให้คุณสมบัติที่คงที่สำหรับการวิเคราะห์การกระจายของจุดตัดศูนย์กลางได้ ทำให้ต้องวิเคราะห์ทั้งคำโดยการให้การหาจุดสิ้นสุดมาช่วย ซึ่งได้นำหลักการมาใช้ในทางปฏิบัติ ดังต่อไปนี้
  1. ใช้การ์ดเสียง(Sound Blaster Compatible) ทำการเปลี่ยนเสียงจากไมโครโฟนเป็นสัญญาณดิจิทัลแบบไม่มีเครื่องหมายขนาด 8 บิต (0-255) ที่ความถี่ 11 กิโลเฮิร์ต โดยทำการหาจุดสิ้นสุดของสัญญาณไปพร้อมกันเพื่อหาขอบเขตของการวิเคราะห์ในขั้นต่อไป
  2. การหาจุดสิ้นสุดของสัญญาณทำได้โดยพิจารณาความถี่ของการสุ่มสัญญาณร่วมกับระดับของสัญญาณด้วย โดยโครงานนี้ได้กำหนดให้เริ่มทำการวิเคราะห์เมื่อระดับสัญญาณสูงกว่าระดับปกติตั้งแต่ 3 ขึ้นไปติดต่อกัน 16 ครั้งการสุ่ม ตัวเลขทั้งสองได้มาจากการทดลองหลายๆครั้งทั้งนี้หากเปลี่ยนความถี่สุ่ม อาจต้องทดลองเปลี่ยนตัวเลขนี้ใหม่
  3. จากนั้น นำสัญญาณที่ได้ไปลดเหลือสองระดับโดยวิธีหาจุดตัดศูนย์กลาง
  4. แบ่งความถี่ของการกระจายของการตัดศูนย์กลางเป็น 16 อันตรภาคชั้น โดยชั้นที่สั้นที่สุดคือหนึ่งครั้งการสุ่ม หรือ 90 ไมโครวินาที ที่อัตราสุ่ม 11 กิโลเฮิร์ต และยาวที่สุด(ความถี่ต่ำสุดที่พอใจ) คือ 64 ครั้งการสุ่ม หรือ 5.8 มิลลิวินาที หรือเท่ากับความถี่ 86 เฮิร์ต เพื่อลดการคำนวณลง
  5. นำค่าที่ได้ไปหาความคล้ายโดยใช้ทฤษฎีเมตริกซ์ความคล้าย(Contingency matrix method) ซึ่งเมตริกซ์ความคล้ายจะมีขนาด 3 แถว 17 หลัก ดังนี้

$$\begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & \cdots & a_{1,16} \\ a_{2,1} & a_{2,2} & a_{2,3} & \cdots & a_{2,16} \\ a_{3,1} & a_{3,2} & a_{3,3} & \cdots & a_{3,16} \end{bmatrix} \begin{bmatrix} \sum a_{1,j} \\ \sum a_{2,j} \\ \sum a_{3,j} \end{bmatrix}$$

เมื่อ  $j$  มีค่าตั้งแต่ 1 ถึง 16

โดยที่  $a_{1,1}$  ถึง  $a_{1,16}$  เป็นแถวเมตริกซ์ของค่าที่ทดสอบ

$a_{2,1}$  ถึง  $a_{2,16}$  เป็นแถวของค่าต้นแบบ

หรือเขียนเป็นสมการได้ดังนี้

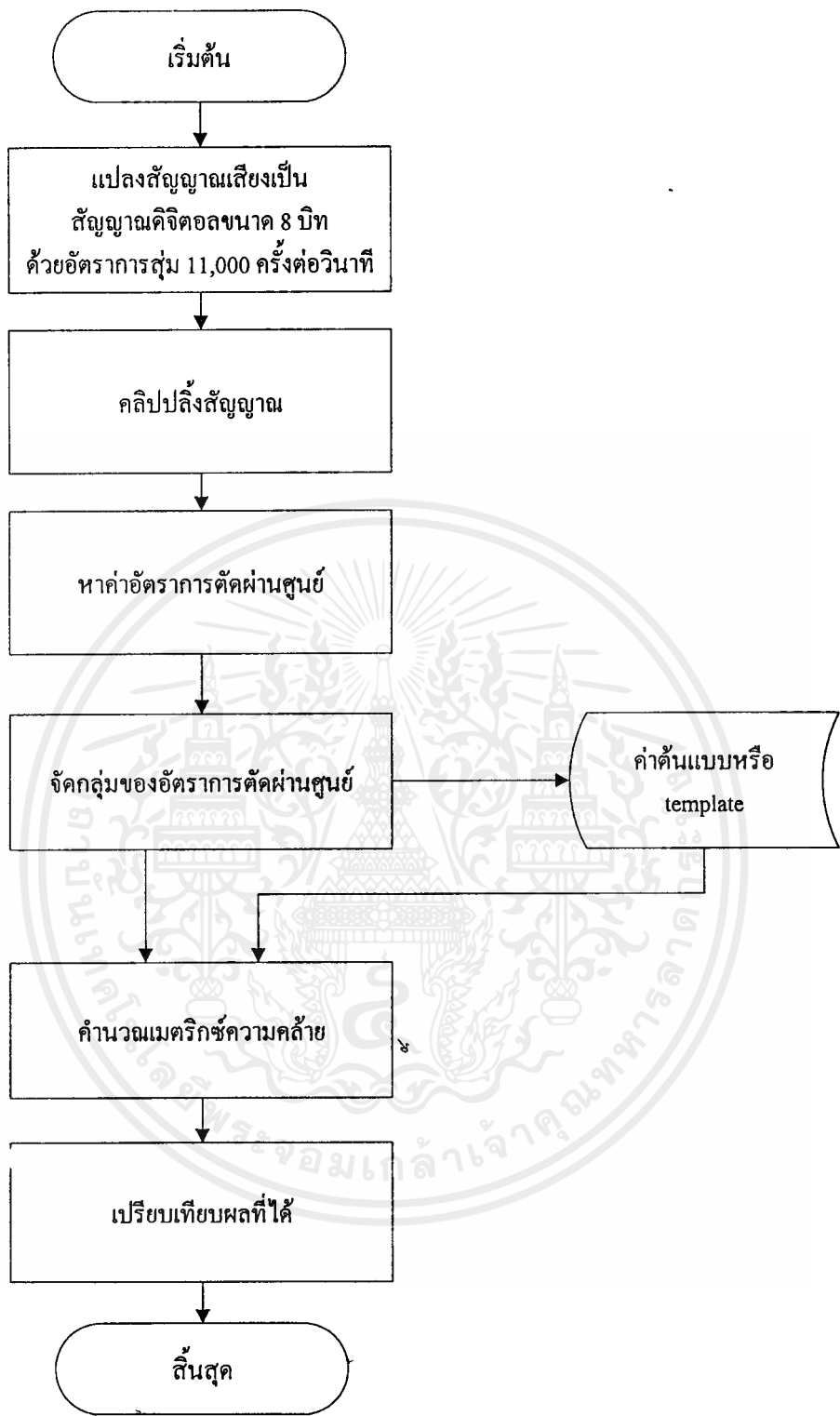
$$C = \sum_{i=1}^2 \sum_{j=1}^{16} a_{i,j} * \log_2 \frac{a_{i,j}}{\sum a_i \sum a_j}$$

เมื่อ C คือค่าเอนโทรปี (Entropy)

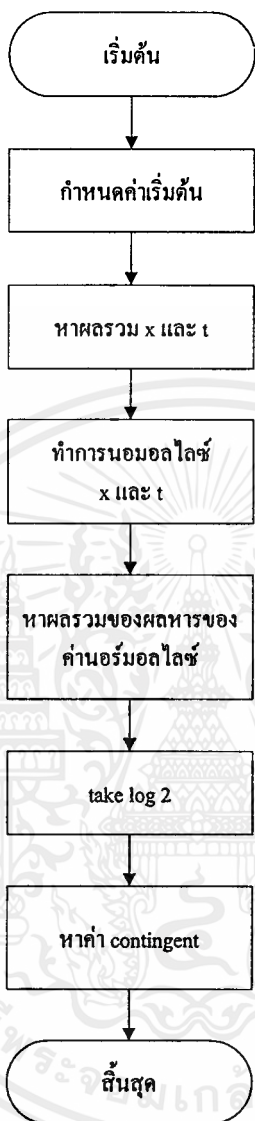
6. การพิจารณาว่าสัญญาณใดๆคล้ายกัน เอนโทรปีจะมีค่าเข้าใกล้ศูนย์ โดยโครงการนี้เลือกพิจารณาที่น้อยกว่าหรือเท่ากับ 0.001

7. ผลลัพธ์จากการเปรียบเทียบที่ได้นำไปควบคุมการปิด-เปิดอุปกรณ์ไฟฟ้าเพื่อเสนอแนวทางการนำไปใช้ในทางปฏิบัติ





รูปที่ 3.1 แผนภูมิแสดงการทำงานของโปรแกรม



รูปที่ 3.2 แผนภูมิแสดงขั้นตอนการหาเมตริกซ์ความคล้าย

### 3.3 การสร้างฮาร์ดแวร์

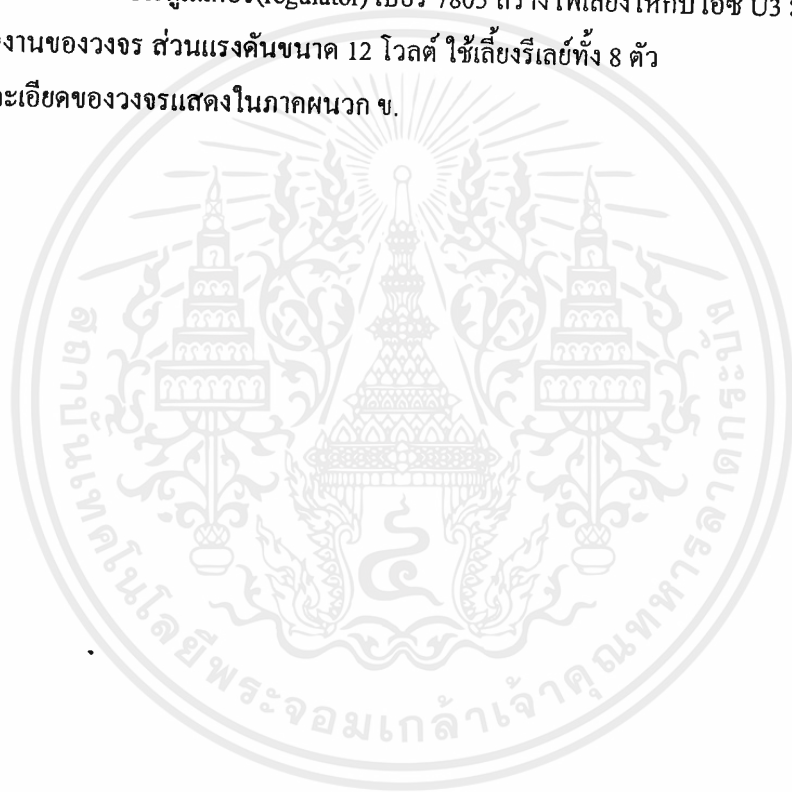
ผลจากการทำงานของโปรแกรม นำไปควบคุมอุปกรณ์ไฟฟ้าโดยผ่านทางวงจรรีเลย์เฟส ทางพอร์ทขนาน(parallel port) เนื่องจากสะดวกต่อการต่อเข้ากับคอมพิวเตอร์และใช้อุปกรณ์น้อยชิ้น

#### หลักการทำงาน

ไอซี U3 เบอร์ 74LS374 เป็น D-FlipFlop ขนาด 8 บิต ทำหน้าที่แลตช์(latch) ข้อมูลจากพอร์ทขนานที่ขอบขาขึ้นของสัญญาณนาฬิกา(clock) ที่ต่อกับขาสโตรบ(strobe)ของพอร์ทขนาน เอาท์พุทจาก U3 จะต่อเข้ากับ U2 ซึ่งเป็น ไอซีไดรเวอร์(driver) ขนาด 8 บิต สามารถขับรีเลย์ได้โดยตรง โดยมี LED D3, D4....ทำหน้าที่แสดงสถานะของรีเลย์(relay)แต่ละตัว

ภาคจ่ายไฟใช้ไอซีเรกูเลเตอร์(regulator) เบอร์ 7805 สร้างไฟเลี้ยงให้กับไอซี U3 มี LED D2 แสดงสถานะการทำงานของวงจร ส่วนแรงดันขนาด 12 โวลต์ ใช้เลี้ยงรีเลย์ทั้ง 8 ตัว

รายละเอียดของวงจรแสดงในภาคผนวก ข.



## บทที่ 4

### การทดลองและผลการทดลอง

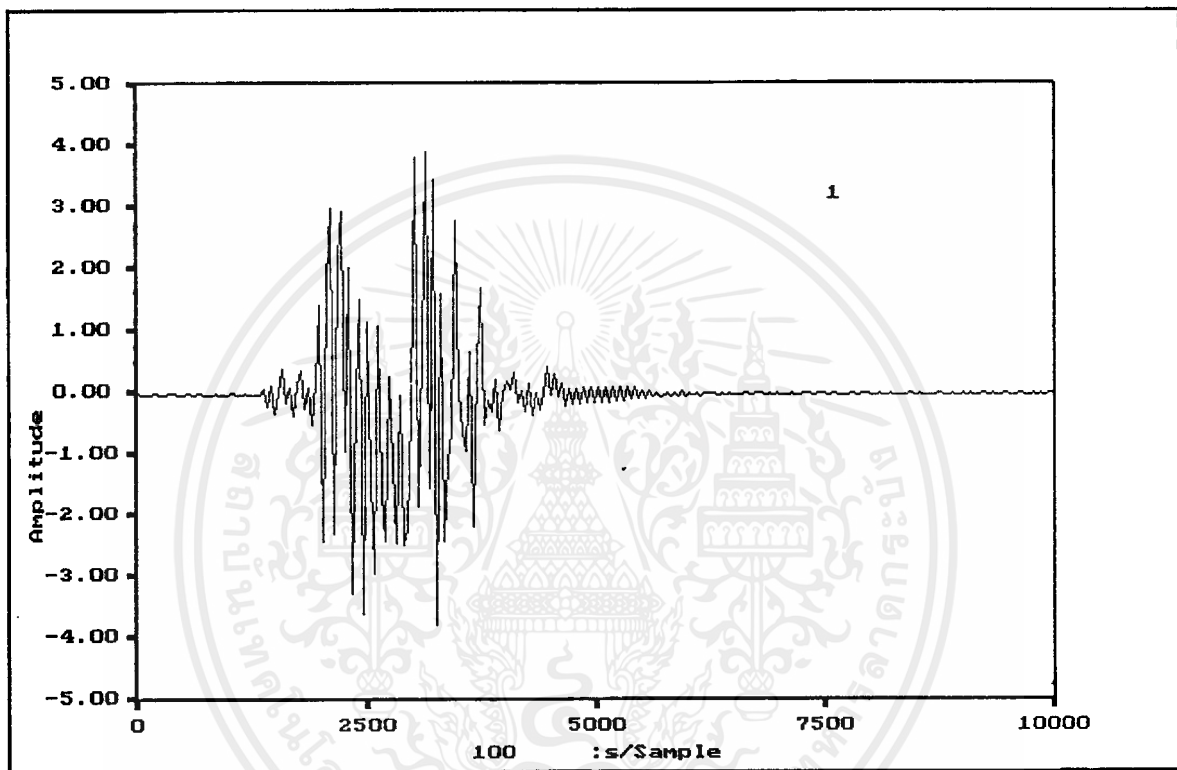
เนื่องจากระบบการรู้จำในการทดลองนี้เป็นการรู้จำแบบแยกคำและผู้พูดเฉพาะเจาะจง เพราะฉะนั้นคำที่ใช้จึงต้องเป็นคำโดด โดยสามารถใช้คำพูดใดก็ได้ แต่ต้องเก็บคำๆ นั้นไว้ในคำค้นแบบก่อน และผู้พูดกับผู้เก็บเสียงในคำค้นแบบจะต้องเป็นคนๆ เดียวกัน

#### ขั้นตอนการทดลอง

1. พูดคำว่า หนึ่ง สอง สาม สี่ ห้า หก เจ็ด แปด เก็บไว้เป็นคำค้นแบบ
2. ทดลองพูดคำใดๆ ( หนึ่ง,สอง,สาม,สี่,ห้า,หก,เจ็ด,แปด) เพื่อให้โปรแกรมตัดสินใจว่าตรงกับคำใด โดยพูดคำดังกล่าวคำละ 10 ครั้ง แล้ววัดจำนวนครั้งที่ถูกต้อง



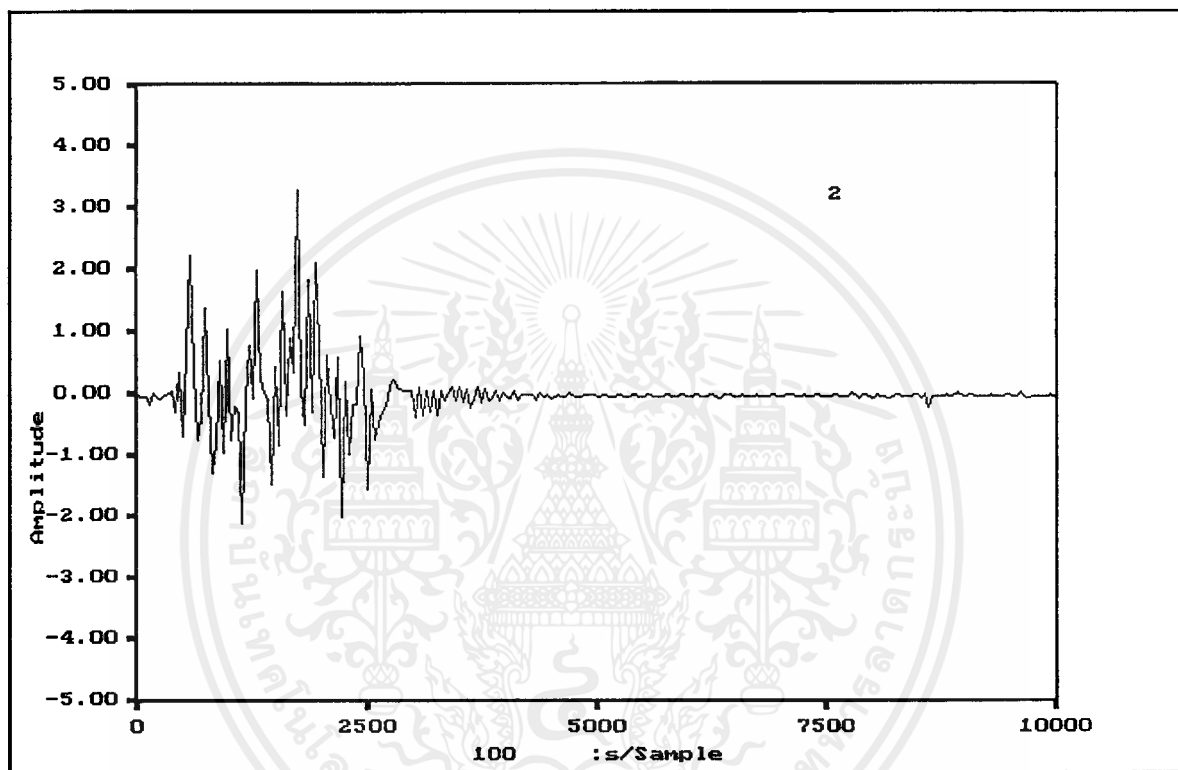
คำว่า “หนึ่ง”



ถูกต้อง 6 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

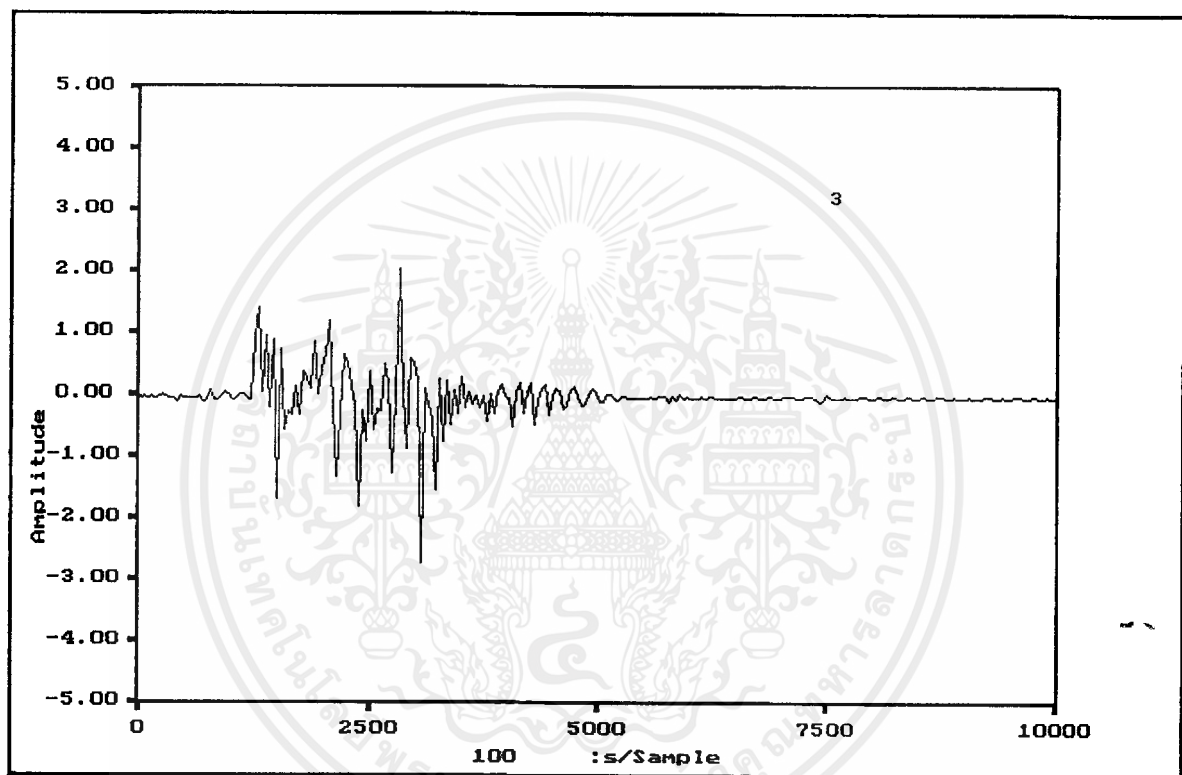
คำว่า"สอง"



ถูกต้อง 9 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

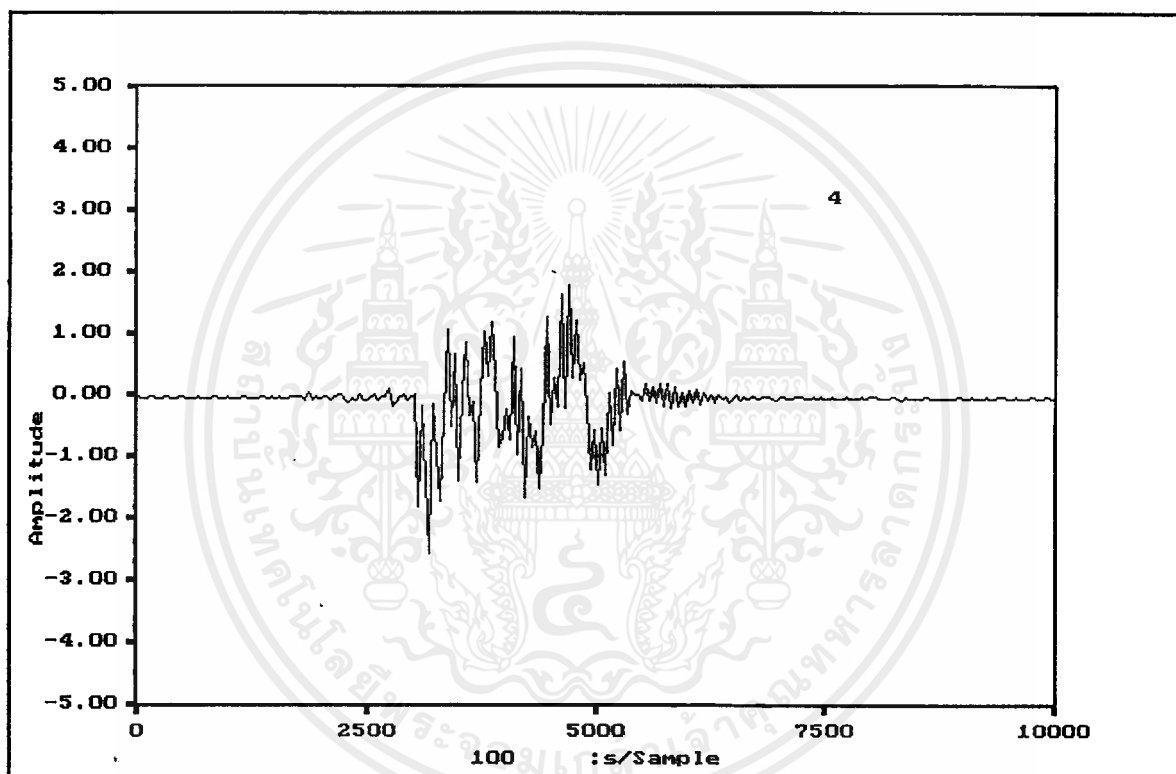
คำว่า “สาม”



ถูกต้อง 9 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

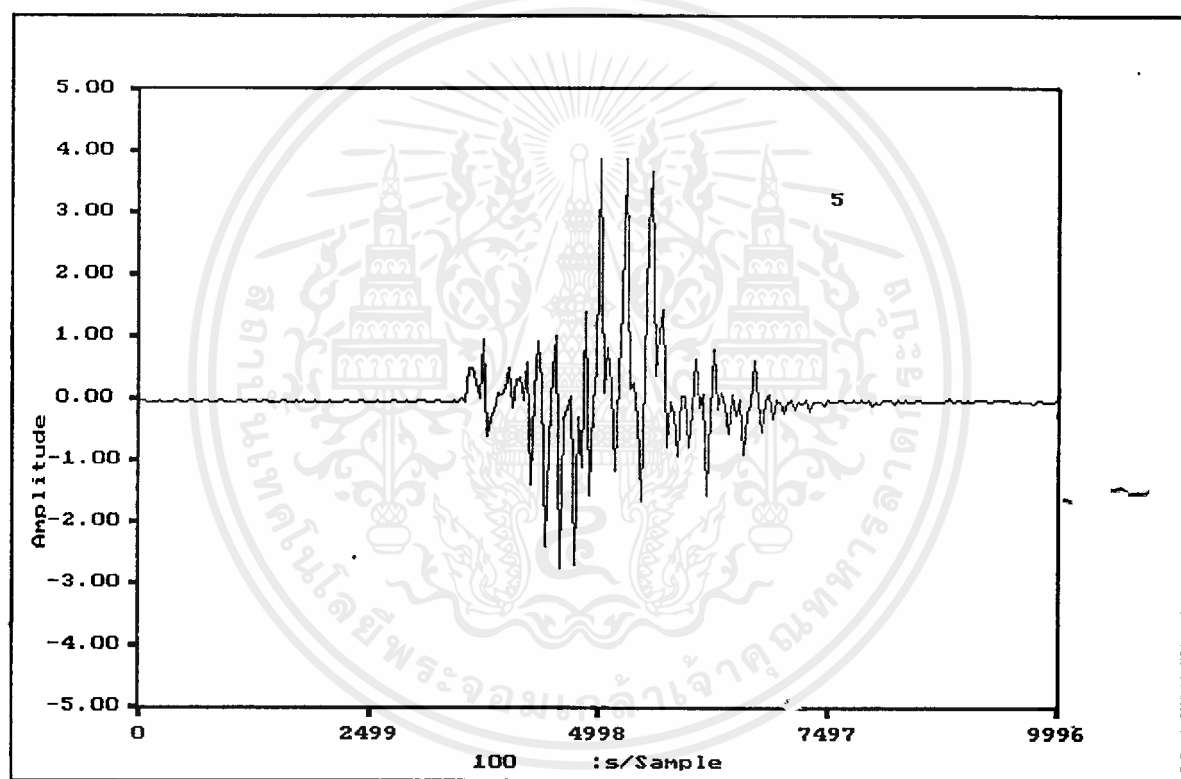
คำว่า “สี่”



ถูกต้อง 5 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

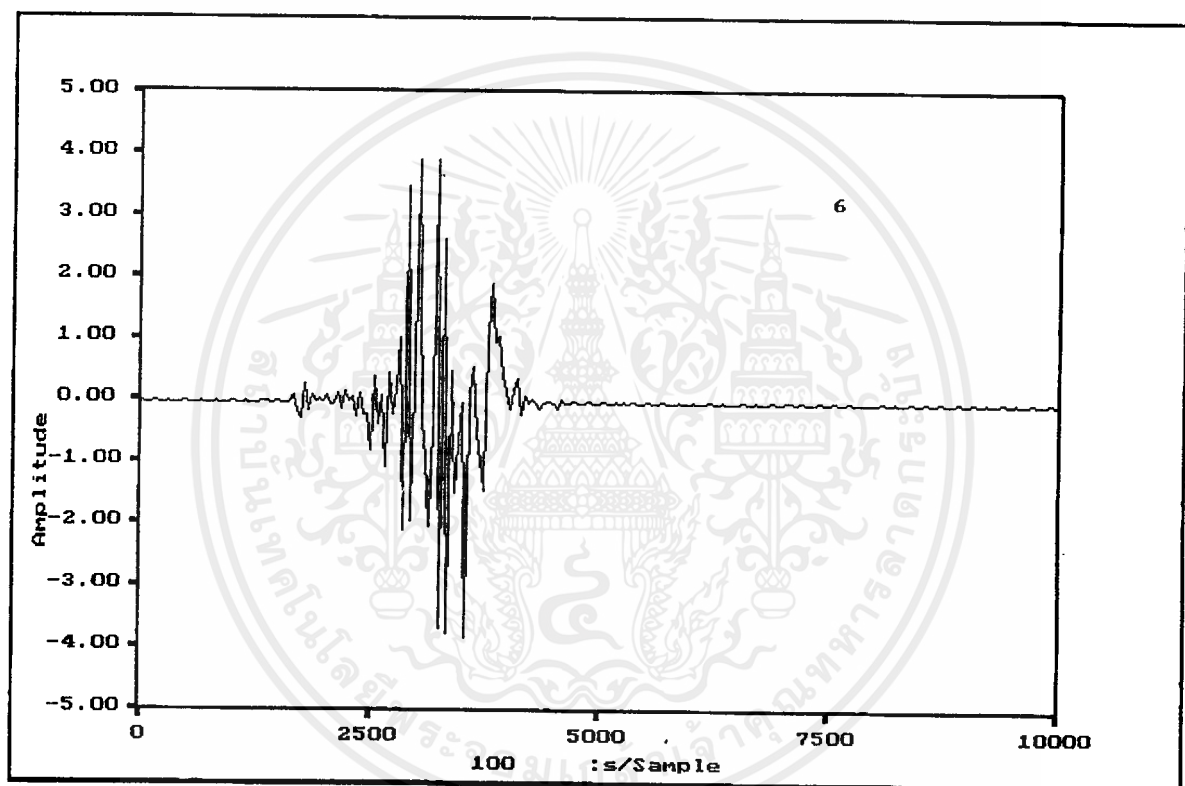
คำว่า “ห้า”



ถูกต้อง 6 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

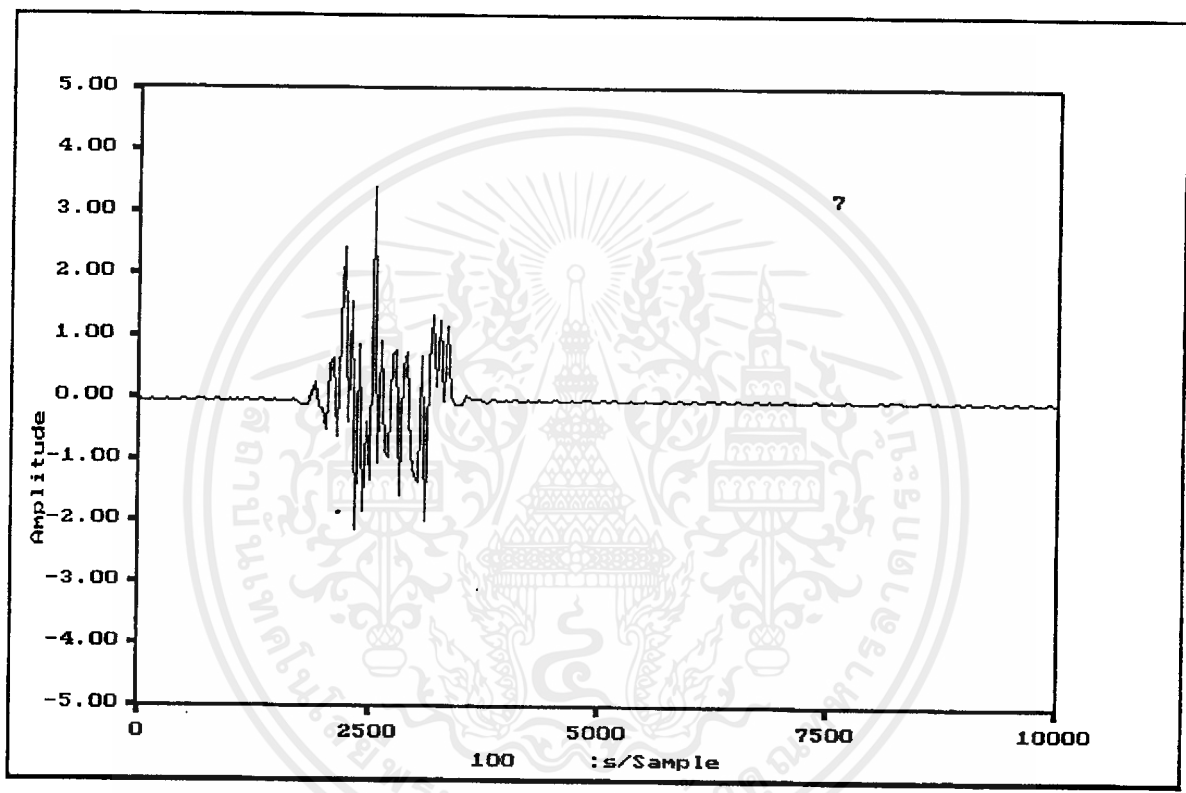
คำว่า “หก”



ถูกต้อง 8 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

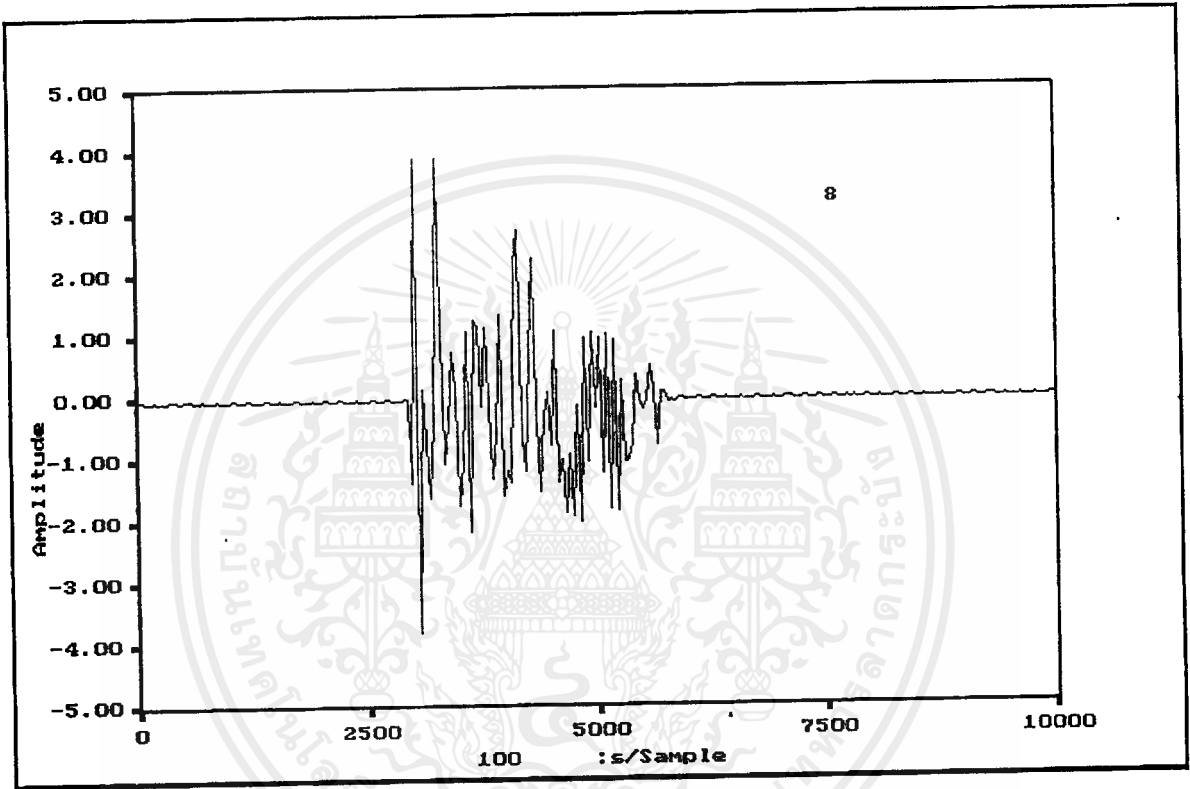
คำว่า "เจ็ด"



ถูกต้อง 8 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำว่า “เปิด”



ถูกต้อง 9 ครั้ง

## บทที่ 5

### สรุปผลการทดลอง

จากผลการทดลอง เมื่อคิดเป็นเปอร์เซ็นต์จะได้ผลตามตาราง

คำ	เปอร์เซ็นต์ความถูกต้อง
หนึ่ง	60
สอง	90
สาม	90
สี่	50
ห้า	60
หก	80
เจ็ด	80
แปด	90

จากผลการทดลองที่ได้ตั้งตาราง ถือว่าอยู่ในขั้นที่น่าพอใจ มีบางคำถูกต้องถึง 90 เปอร์เซ็นต์ แม้ว่าจะมีบางคำมีเปอร์เซ็นต์ความถูกต้องน้อย ซึ่งปัจจัยที่มีผลกระทบต่อผลการทดลองนี้ มีอยู่หลายปัจจัย แต่ปัจจัยที่สำคัญที่สุดก็คือเสียงรบกวนจากภายนอกนั่นเอง

จากการทดลองที่ผ่านมาแต่ละครั้งนั้นจะได้ค่าเปอร์เซ็นต์ความถูกต้องแตกต่างกันไปขึ้นอยู่กับสิ่งแวดล้อม ซึ่งถ้าสิ่งแวดล้อมใดมีเสียงรบกวนมากก็อาจทำให้มีความผิดพลาดเกิดขึ้นได้สูง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/*
  Compile with Turbo C 2.0 in a small-model then link with DIRECT.OBJ
  modified to work well with Pentium 133 with Sound Blaster 16
*/
#define CPUSPEED 133
#include <stdio.h>
#include <stdlib.h>
#include <alloc.h>
#include <math.h>
#include <io.h>
#include <fcntl.h>
#include <string.h>
#include <dos.h>
#define SBOK 0
/* SB direct access functions, driver independent, catch from the internet
:) */
extern int read_data(void);
extern void write_data(int __dat);
extern void speaker_on(void);
extern void speaker_off(void);
extern int reset_dsp(void);
extern void asmdelay(int __delay);
#define TRUE 1
#define FALSE 0
#define ERROR(x) {printf("\nAn error occured: "); printf(x);
printf("\n");}
#define MAXDICTIONARY 64
#define MAX_LABEL_LEN 8
#define PDELAY CPUSPEED*6
#define RDELAY CPUSPEED*2.3
typedef int boole;
typedef unsigned char byte;
typedef float soundvect[16];

unsigned zerotable[64];
soundvect dictionary[MAXDICTIONARY], parvect;
char identifiers[MAXDICTIONARY][MAX_LABEL_LEN];
int dictsize=0;
unsigned zerolength=0;
/*-----hardware interface routines-----
--*/
/* thanks Norton Guide for parallel-port informations */
int relay_stat=0;
unsigned int datap, sttsp, ctrlp;

relay_init()
{
    int far *lpt;
    int port;

    /* get base address of LPTn: */
    for(port=1;port<=4;port++){
        lpt=MK_FP(0x0040,0x0008+port*2-2);
        if(*lpt)break;
    }
    datap=*lpt;
    sttsp=datap+1;
    ctrlp=datap+2;
    outportb(ctrlp,0xCC);
    relay_clear();
}
relay_toggle(int rl)
{
    if(rl<1 || rl>8){
        printf("function relay_toggle : Warning, relay number out of

```

range!\n");

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        return;
    }
    rl--;
    relay_stat^=(1<<rl);          /* toggle bit */
    outportb(datap,relay_stat);
    outportb(ctrllp,0xCC|0x01);  /* /STB=1 */
    outportb(ctrllp,0xCC&0xfe); /* /STB=0 */
    relay_status();
    sound(4000);
    delay(50);
    nosound();
}
relay_clear()
{
    outportb(datap,relay_stat=0x00);
    outportb(ctrllp,0xCC|0x01);  /* /STB=1 */
    outportb(ctrllp,0xCC&0xfe); /* /STB=0 */
}
relay_status()
{
    int i;

    printf("Relay  : ");
    for(i=0;i<8;i++)
        printf(" %d  ",i+1);
    printf("\nStatus : ");
    for(i=0;i<8;i++)
        if(relay_stat&(1<<i))
            printf(" ON ");
        else
            printf("OFF ");
    printf("\n");
}
/*-----end hardware routines-----
--*/

void play_sample(byte *snd,long size)
{
    long i;
    byte *wpl;

    wpl=snd;
    speaker_on();
    for(wpl=snd,i=0;i<size;i++,wpl++){
        write_data(*wpl); asmdelay(PDELAY);
    }
    speaker_off();
}

void clip(byte *signal,unsigned size)
{
    unsigned i;

    for (i=0;i<size;i++)
        if ((byte)signal[i]>128) (byte)signal[i]=255;
        else (byte)signal[i]=0;
}

void classify(unsigned length)
{
    zerotable[(length>64)?63:length-1]++;
}

void analyze(byte *signal,unsigned size)
{
    unsigned i;

    clip(signal,size);
    for (i=1;i<size;i++){
        zerolength++;

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if ((byte)signal[i]!=(byte)signal[i-1]) {
            classify(zeroLength); zeroLength=0;
        }
    }
}

byte limits[17]={0,1,2,3,4,5,6,7,8,10,12,15,19,25,34,48,64};

void addtovector(byte i,float a)
/*
table 1 | 2 | 3 | 4 | 5 | 6 | 7 8 | 9 | 11 | 13 | 16 | 20 | 26 | 35 |
49
      | | | | | | | | -10|-12 |-15 |-19 |-25 |-34 |-48
|-64
class 1 | 2 | 3 | 4 | 5 | 6 | 7 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
16
*/
{
    byte j,size;

    j=i/4; while (i>limits[j]) j++;
    size=limits[j]-limits[j-1];
    parvect[j-1]+=(float)a/(float)size;
}

void analyze_table(void)
{
    unsigned max=1;
    byte i;

    for(i=0;i<64;i++) if (zerotable[i]>max) max=zerotable[i];
    for(i=0;i<64;i++) addtovector(i+1,(float)zerotable[i]/(float)max);
}

void initvar(void)
{
    byte i;

    for(i=0;i<64;i++) {
        parvect[i/4]=0.0;
        zerotable[i]=0;
    }
}

boole record(boole quiet)
{
    byte threshold=3;
    byte *snd;
    int ctr,w;
    long i,maxsize=32767;
    boole result=FALSE;

    initvar();
    if(reset_dsp()!=SBOK) {
        printf("\nError resetting Sound Blaster.\n"); exit(1);
    }
    snd=(byte*)malloc(maxsize); ctr=0;
    if (snd==(byte*)NULL) {
        ERROR("Memory allocation (fct record)"); exit(0);
    }
    printf("Waiting...");
    do {
        w=read_data(); asmdelay(RDELAY);
        if (abs(w-128)>threshold) ctr++;
        else ctr=0;
    } while (ctr<16);
    printf(" recording...");
    ctr=0; i=0;
    do {
        w=read_data(); asmdelay(RDELAY);

```

```

        if (abs(w-128)<threshold) ctr++;
        else ctr=0;
        snd[i++]=w;
    } while ((ctr<512) && (i<maxsize));
    printf(" Done (%u)\n",i);
    i-=1024;
    if (i>512) {
        result=TRUE;
        if (!quiet) play_sample(snd,i);
        analyze(snd+16,i);
        analyze_table();
    }
    free(snd);
    return(result);
}

void training(void)
{
    char wk[64],wk2[8];
    boole done,inok;
    byte j;

    printf("You can now train up to %d words. You will be prompted
to\n",
        MAXDICTIONARY);
    printf(" enter label for each word, then pronounce it.\n");
    do {
        printf("\nPlease enter label # %d, or Q to quit training: ",
            dictsize);
        scanf("%s",wk);
        strupr(wk);
        done=!strcmp(wk,"Q");
        if (!done) {
            strcpy(identifiers[dictsize],wk);
            do {
                inok=record(FALSE);
                if (!inok) {
                    printf("No word identified. ");
                    printf("Try again (Y/N)? ");
                    scanf("%s",wk2);
                    strupr(wk2);
                    done=!strcmp(wk2,"N");
                    if (done) inok=TRUE;
                }
            } while (!inok);
            if (!done) {
                printf("Insert %s into dictionary (Y/N)? ",
                    wk);
                scanf("%s",wk2);
                strupr(wk2);
                if (!!strcmp(wk2,"N")){
                    printf("Parvect : ");
                    for(j=0;j<16;j++){
                        dictionary[dictsize][j]=parvect[j];
                        printf("%1.3f\t",parvect[j]);
                    }
                    printf("\n");
                    dictsize++;
                }
            }
        }
    } while (!done);
}

float contingency(byte n)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* compares the parameter vector (parvect) to the nth vector stored
in the dictionary. Returns a 0<=value<=1.
for the similarity of the two vectors. 0.0 means very similar.
Error status: -1.0 -- All matrix fields zero, cannot divide
*/

```

```

{
    float s,t=0.0;
    byte i,j;
    float pij;
    float cmatrix[3][17];
    float result;

    /* copy parameter vectors into matrix, calculate line and
    overall sums
    */
    cmatrix[2][16]=0.0;
    for (i=0;i<2;i++) {
        cmatrix[i][16]=0.0;
        for (j=0;j<16;j++) {
            if (i==0) cmatrix[i][j]=parvect[j];
            else cmatrix[i][j]=dictionary[n][j];
            cmatrix[i][16]+=cmatrix[i][j];
        }
        cmatrix[2][16]+=cmatrix[i][16];
    }
    if (cmatrix[2][16]==0.0) result=-1.0;
    else {
        /* normalize matrix to overall sum=1.0 */
        for (i=0;i<2;i++)
            for (j=0;j<17;j++)
                cmatrix[i][j]/=cmatrix[2][16];
        cmatrix[2][16]=1.0;
        /* calculate column sums */
        for (j=0;j<16;j++) {
            cmatrix[2][j]=0.0;
            for (i=0;i<2;i++)
                cmatrix[2][j]+=cmatrix[i][j];
        }
        /* calculate rate of transmission */
        for (i=0;i<2;i++)
            for (j=0;j<16;j++) {
                s=cmatrix[i][16]*cmatrix[2][j];
                if (s>0.0) {
                    pij=cmatrix[i][j];
                    s=pij/s;
                    if (s>0.0) t=t+pij*(log(s)/log(2));
                }
            }
        result=t;
    }
    return(result);
}

```

```

typedef struct
{
    byte first[256],second[256];
} matchstrc;

```

```

matchstrc *match(void)
/* Compares the parvect to each one stored in the dictionary;
Error status: NULL returned -- dictionary empty
*/

```

```

{
    byte i,cf,cs;
    int rs,minval=2000;
    int rsarr[MAXDICTIONARY];
    matchstrc *result=NULL;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for (i=0;i<dictsize;i++) {
    rs=(int)(contingency(i)*1000);
    if ((rs>-1000) && (rs<minval)) minval=rs;
    rsarr[i]=rs;
}
if (minval<60) {
    result=(matchstrc*)malloc(sizeof(matchstrc));
    if (result!=NULL) {
        cf=cs=0;
        for (i=0;i<dictsize;i++) {
            if (rsarr[i]==minval)
                result->first[cf++]=i+1;
            if (rsarr[i]==minval+1)
                result->second[cs++]=i+1;
        }
        result->first[cf]=0;
        result->second[cs]=0;
    }
}
return(result);
}

void ppmatches(matchstrc m)
{
    byte cf=0,cs=0,i;
    int relay;

    relay=-1;
    while (m.first[cf]>0) cf++;
    while (m.second[cs]>0) cs++;
    if (cf==1) {
        printf("%s (%d) matched best. ",identifiers[m.first[0]-
1],m.first[0]-1);
        relay=atoi(identifiers[m.first[0]-1]);
    }
    else {
        for (i=0;i<cf;i++) {
            printf("%s (%d)",identifiers[m.first[i]-1],m.first[i]-
1);
            relay=atoi(identifiers[m.first[i]-1]);
            if (i<cf-2) printf(", ");
            else if (i<cf-1) printf(" or ");
            else printf(" match best. ");
        }
    }
    if (cs==1) printf("%s comes close",identifiers[m.second[0]-1]);
    else {
        for (i=0;i<cs;i++) {
            printf("%s",identifiers[m.second[i]-1]);
            if (i<cs-2) printf(", ");
            else if (i<cs-1) printf(" and ");
            else printf(" come close.");
        }
    }
    printf("\n");
    relay_toggle(relay);
}

void recogniser(void)
{
    boole done=FALSE;
    matchstrc *matches;

    printf("\nNow you can speak words, the program will match them
with\n");
    printf(" the trained dictionary and give the closest matches.\n");
    printf(" Press <ESC> to abort...\n");

```

```

delay(2000);
do {
    if(kbhit() && getch()==27) done=TRUE;
    if (!done) {
        if (record(TRUE)) {
            matches=match();
            if (matches!=NULL) {
                ppmatches(*matches);
                free(matches);
            }
            else printf("No matches.\n");
        }
        else printf("No word identified during recording\n");
    }
} while (!done);
}

void initidents(void)
{
    int i;

    for(i=0;i<MAXDICTIONARY;i++)
        identifiers[i][0]='\0';
}

/* Save sound dictionary to a file */
int save_to_file(char* filename)
{
    unsigned int i,j;
    FILE *out;

    if((out = fopen(filename,"wt")) == NULL)
    {
        fprintf(stderr, "Cannot open output file.\n");
        return -1;
    }

    fprintf(out, "%d\n",dictsize);
    for (i=0;i<dictsize;i++){
        fprintf(out, "%s\n", identifiers[i]);
        for (j=0;j<16;j++)
            fprintf(out, "%f ", dictionary[i][j]);
        fprintf(out, "\n");
    }
    printf("%d word(s) template saved successfully.\n",dictsize);
    fclose(out);
    return 0;
}

/* Read sound dictionary from a file */
int read_from_file(char* filename)
{
    unsigned int i,j;
    FILE *in;

    if((in = fopen(filename, "rt")) == NULL)
    {
        fprintf(stderr, "Cannot open input file %s.\n", filename);
        return -1;
    }

    fscanf(in, "%d", &dictsize);
    for (i=0;i<dictsize;i++){
        fscanf(in, "%s", identifiers[i]);
        for (j=0;j<16;j++)
            fscanf(in, "%f", &dictionary[i][j]);
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

printf("%d word(s) template loaded successfully.\n",dictsize);
fclose(in);
return 0;
}

main()
{
int i;
char fname[80];

relay_init();

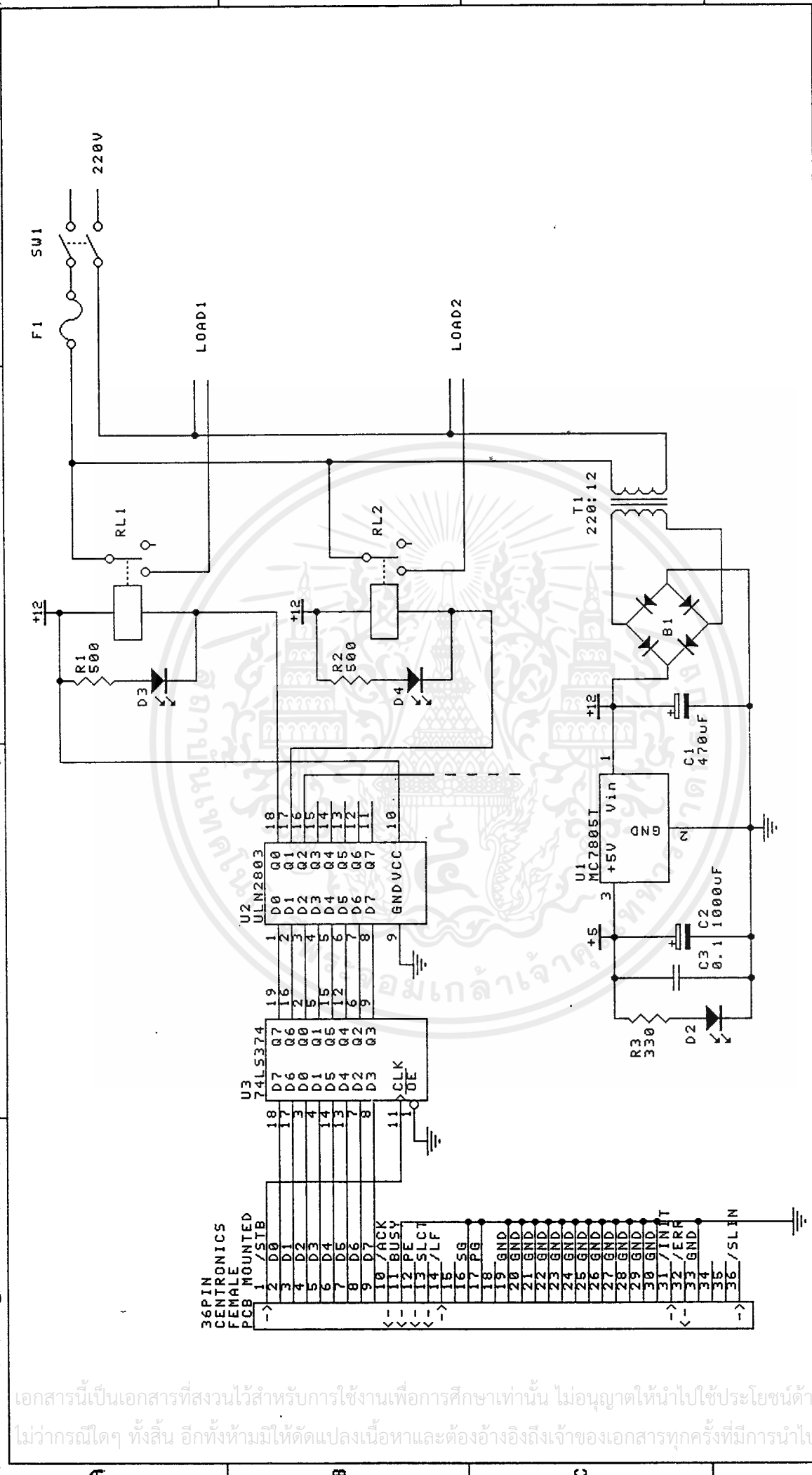
printf("*** Speech Recognition Program ***\n");
do {
printf("Please select : \n");
printf(" 1. Training\n");
printf(" 2. Recognize\n");
printf(" 3. Save template to file\n");
printf(" 4. Load template from file\n");
printf(" 5. Quit to dos\n");
i=getch();
switch (i) {
case '1':
dictsize=0;
initidents();
training();
printf("%d word(s) trained.\n",dictsize);
break;
case '2':
if(!dictsize)
printf("No template available,try training
or load from a file\n");
else
recogniser();
break;
case '3':
if(!dictsize)
printf("Blank template,file not saved\n");
else{
printf("File name :");
scanf("%s",fname);
save_to_file(fname);
}
break;
case '4':
printf("File name :");
scanf("%s",fname);
if(read_from_file(fname)==-1)
dictsize=0;
break;
case '5':
break;
default:
sound(1500);
delay(400);
nosound();
break;
}
}while(i!='5');
relay_clear();
return(0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

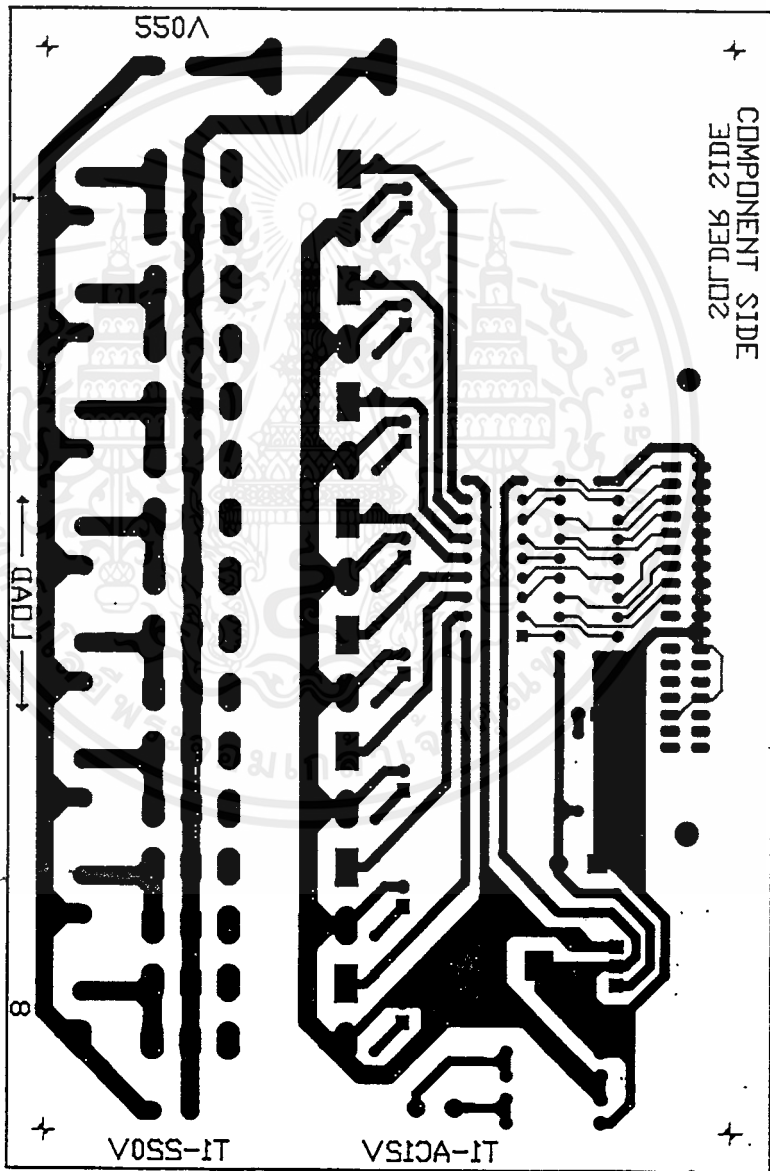


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Title		LPT PORT RELAY CARD	
Size	Number	Revision	
A4	1	A	
Date:	12-FEB-1998	Sheet	1 of 1
File:	PIN/I	Drawn By:	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# OCTAL TRANSPARENT LATCH WITH 3-STATE OUTPUTS; OCTAL D-TYPE FLIP-FLOP WITH 3-STATE OUTPUT

The SN54/74LS373 consists of eight latches with 3-state outputs for bus organized system applications. The flip-flops appear transparent to the data (data changes asynchronously) when Latch Enable (LE) is HIGH. When LE is LOW, the data that meets the setup times is latched. Data appears on the bus when the Output Enable (OE) is LOW. When OE is HIGH the bus output is in the high impedance state.

The SN54/74LS374 is a high-speed, low-power Octal D-type Flip-Flop featuring separate D-type inputs for each flip-flop and 3-state outputs for bus oriented applications. A buffered Clock (CP) and Output Enable (OE) is common to all flip-flops. The SN54/74LS374 is manufactured using advanced Low Power Schottky technology and is compatible with all Motorola TTL families.

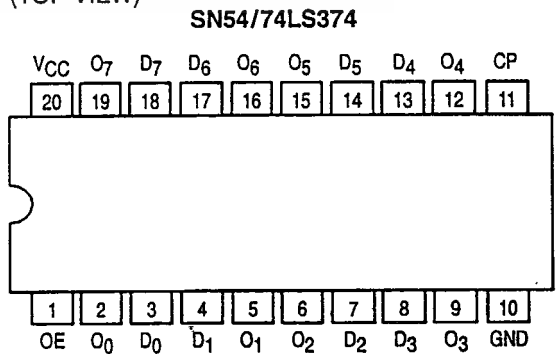
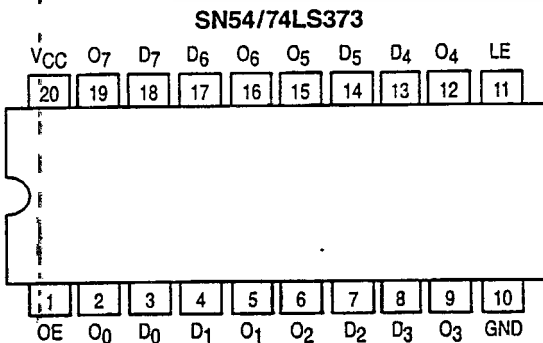
- Eight Latches in a Single Package
- 3-State Outputs for Bus Interfacing
- Hysteresis on Latch Enable
- Edge-Triggered D-Type Inputs
- Buffered Positive Edge-Triggered Clock
- Hysteresis on Clock Input to Improve Noise Margin
- Input Clamp Diodes Limit High Speed Termination Effects

**PIN NAMES**

		LOADING (Note a)	
		HIGH	LOW
D <sub>0</sub> -D <sub>7</sub>	Data Inputs	0.5 U.L.	0.25 U.L.
LE	Latch Enable (Active HIGH) Input	0.5 U.L.	0.25 U.L.
CP	Clock (Active HIGH going edge) Input	0.5 U.L.	0.25 U.L.
OE	Output Enable (Active LOW) Input	0.5 U.L.	0.25 U.L.
O <sub>0</sub> -O <sub>7</sub>	Outputs (Note b)	65 (25) U.L.	15 (7.5) U.L.

**NOTES:**

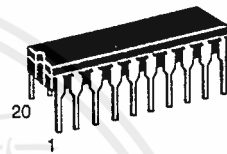
- a) 1 TTL Units Load (U.L.) = 40  $\mu$ A HIGH/1.6 mA LOW.  
 b) The Output LOW drive factor is 7.5 U.L. for Military (54) and 25 U.L. for Commercial (74) Temperature Ranges. The Output HIGH drive factor is 25 U.L. for Military (54) and 65 U.L. for Commercial (74) Temperature Ranges.

**CONNECTION DIAGRAM DIP (TOP VIEW)**


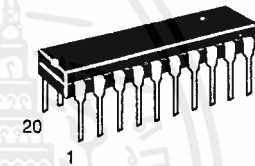
**NOTE:**  
 The Flatpak version has the same pinouts (Connection Diagram) as the Dual In-Line Package.

**SN54/74LS373**  
**SN54/74LS374**

**OCTAL TRANSPARENT LATCH  
 WITH 3-STATE OUTPUTS;  
 OCTAL D-TYPE FLIP-FLOP  
 WITH 3-STATE OUTPUT**  
**LOW POWER SCHOTTKY**



**J SUFFIX  
 CERAMIC  
 CASE 732-03**



**N SUFFIX  
 PLASTIC  
 CASE 738-03**



**DW SUFFIX  
 SOIC  
 CASE 751D-03**

**ORDERING INFORMATION**

SN54LSXXXJ Ceramic  
 SN74LSXXXN Plastic  
 SN74LSXXXDW SOIC

# SN54/74LS373 • SN54/74LS374

## TRUTH TABLE

LS373

D <sub>n</sub>	LE	OE	O <sub>n</sub>
H	H	L	H
L	H	L	L
X	L	L	Q <sub>0</sub>
X	X	H	Z*

LS374

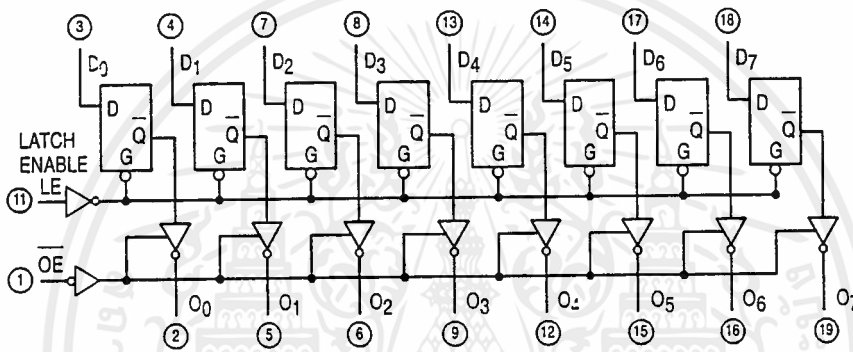
D <sub>n</sub>	LE	OE	O <sub>n</sub>
H		L	H
L		L	L
X	X	H	Z*

H = HIGH Voltage Level  
L = LOW Voltage Level  
X = Immaterial  
Z = High Impedance

\* Note: Contents of flip-flops unaffected by the state of the Output Enable input (OE).

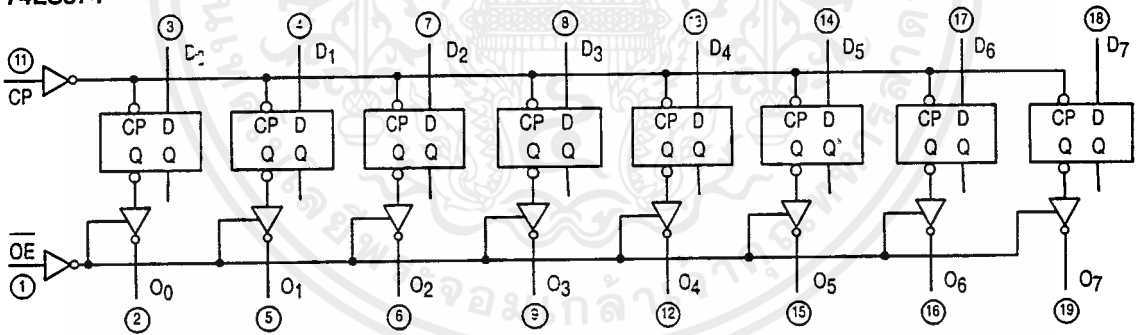
## LOGIC DIAGRAMS

SN54LS/74LS373



V<sub>CC</sub> = PIN 20  
GND = PIN 10  
○ = PIN NUMBERS

SN54LS/74LS374



## GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V <sub>CC</sub>	Supply Voltage	54	4.5	5.0	5.5	V
		74	4.75	5.0	5.25	
T <sub>A</sub>	Operating Ambient Temperature Range	54	-55	25	125	°C
		74	0	25	70	
I <sub>QH</sub>	Output Current — High	54			-1.0	mA
		74			-2.6	
I <sub>OL</sub>	Output Current — Low	54			12	mA
		74			24	

# SN54/74LS373 • SN54/74LS374

## DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter	Limits			Unit	Test Conditions	
		Min	Typ	Max			
$V_{IH}$	Input HIGH Voltage	2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
$V_{IL}$	Input LOW Voltage	54		0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74		0.8			
$V_{IK}$	Input Clamp Diode Voltage		-0.65	-1.5	V	$V_{CC} = \text{MIN}$ , $I_{IN} = -18 \text{ mA}$	
$V_{OH}$	Output HIGH Voltage	54	2.4	3.4	V	$V_{CC} = \text{MIN}$ , $I_{OH} = \text{MAX}$ , $V_{IN} = V_{IH}$ or $V_{IL}$ per Truth Table	
		74	2.4	3.1	V		
$V_{OL}$	Output LOW Voltage	54, 74		0.25	0.4	V	$I_{OL} = 12 \text{ mA}$ $V_{CC} = V_{CC} \text{ MIN}$ , $V_{IN} = V_{IL}$ or $V_{IH}$ per Truth Table
		74		0.35	0.5	V	
$I_{OZH}$	Output Off Current HIGH			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{OUT} = 2.7 \text{ V}$	
$I_{OZL}$	Output Off Current LOW			-20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{OUT} = 0.4 \text{ V}$	
$I_{IH}$	Input HIGH Current			20	$\mu\text{A}$	$V_{CC} = \text{MAX}$ , $V_{IN} = 2.7 \text{ V}$	
				0.1	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 7.0 \text{ V}$	
$I_{IL}$	Input LOW Current			-0.4	mA	$V_{CC} = \text{MAX}$ , $V_{IN} = 0.4 \text{ V}$	
$I_{OS}$	Short Circuit Current (Note 1)	-30		-130	mA	$V_{CC} = \text{MAX}$	
$I_{CC}$	Power Supply Current			40	mA	$V_{CC} = \text{MAX}$	

Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

## AC CHARACTERISTICS ( $T_A = 25^\circ\text{C}$ , $V_{CC} = 5.0 \text{ V}$ )

Symbol	Parameter	Limits						Unit	Test Conditions
		LS373			LS374				
		Min	Typ	Max	Min	Typ	Max		
$f_{MAX}$	Maximum Clock Frequency				35	50		MHz	$C_L = 45 \text{ pF}$ , $R_L = 667 \Omega$
$t_{PLH}$ $t_{PHL}$	Propagation Delay, Data to Output		12 12	18 18				ns	
$t_{PLH}$ $t_{PHL}$	Clock or Enable to Output		20 18	30 30		15 19	28 28	ns	
$t_{PZH}$ $t_{PZL}$	Output Enable Time		15 25	28 36		20 21	28 28	ns	
$t_{PHZ}$ $t_{PLZ}$	Output Disable Time		12 15	20 25		12 15	20 25	ns	$C_L = 5.0 \text{ pF}$

## AC SETUP REQUIREMENTS ( $T_A = 25^\circ\text{C}$ , $V_{CC} = 5.0 \text{ V}$ )

Symbol	Parameter	Limits				Unit
		LS373		LS374		
		Min	Max	Min	Max	
$t_W$	Clock Pulse Width	15		15		ns
$t_s$	Setup Time	5.0		20		ns
$t_h$	Hold Time	20		0		ns

## DEFINITION OF TERMS

**SETUP TIME ( $t_s$ )** — is defined as the minimum time required for the correct logic level to be present at the logic input prior to LE transition from HIGH-to-LOW in order to be recognized and transferred to the outputs.

**HOLD TIME ( $t_h$ )** — is defined as the minimum time following the LE transition from HIGH-to-LOW that the logic level must be maintained at the input in order to ensure continued recognition.

# SN54/74LS373

## AC WAVEFORMS

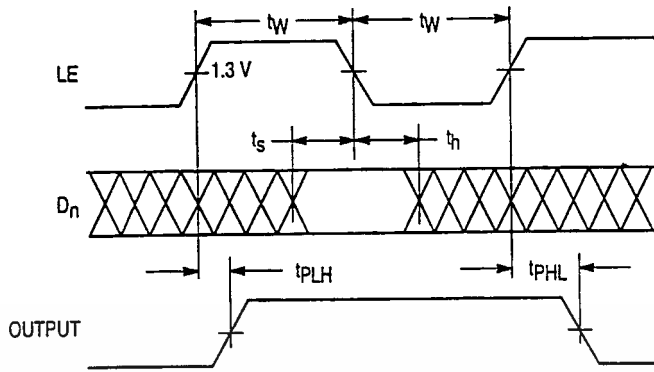


Figure 1

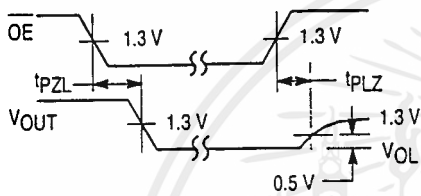


Figure 2

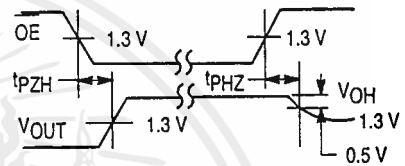
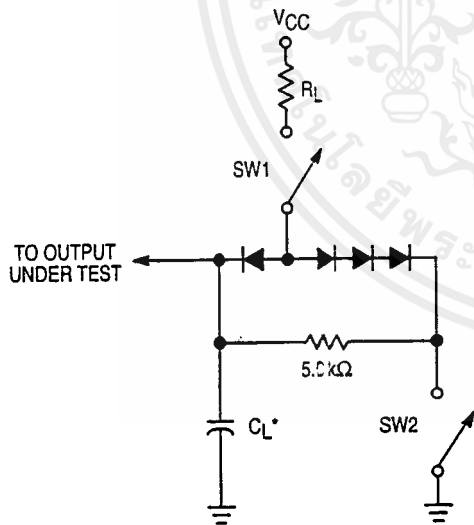


Figure 3

## AC LOAD CIRCUIT



\* Includes Jig and Probe Capacitance.

Figure 4

## SWITCH POSITIONS

SYMBOL	SW1	SW2
$t_{pZH}$	Open	Closed
$t_{pZL}$	Closed	Open
$t_{pLZ}$	Closed	Closed
$t_{pHZ}$	Closed	Closed

# SN54/74LS374

## AC WAVEFORMS

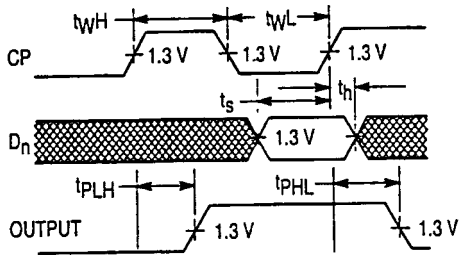


Figure 5

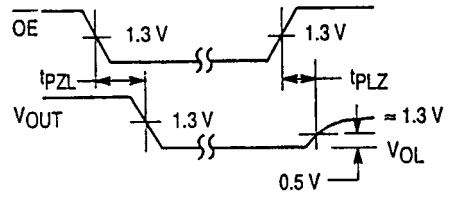


Figure 6

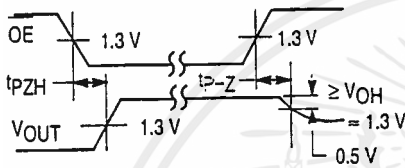
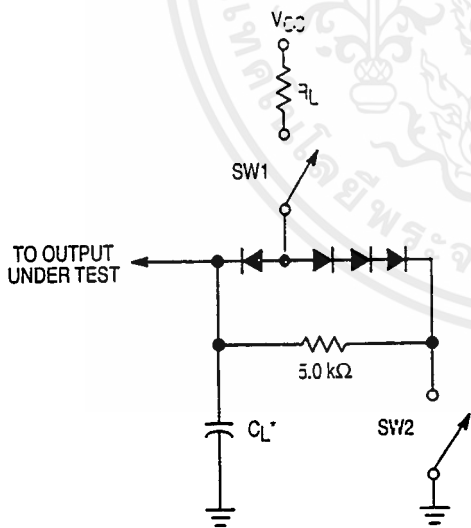


Figure 7

## AC LOAD CIRCUIT



\* Includes Jig and Probe Capacitance.

Figure 8

## SWITCH POSITIONS

SYMBOL	SW1	SW2
tpZH	Open	Closed
tpZL	Closed	Open
tpLZ	Closed	Closed
tpHZ	Closed	Closed



# Octal High Voltage, High Current Darlington Transistor Arrays

The eight NPN Darlington connected transistors in this family of arrays are ideally suited for interfacing between low logic level digital circuitry (such as TTL, CMOS or PMOS/NMOS) and the higher current/voltage requirements of lamps, relays, printer hammers or other similar loads for a broad range of computer, industrial, and consumer applications. All devices feature open-collector outputs and free wheeling clamp diodes for transient suppression.

The ULN2803 is designed to be compatible with standard TTL families while the ULN2804 is optimized for 6 to 15 volt high level CMOS or PMOS.

**MAXIMUM RATINGS** ( $T_A = 25^\circ\text{C}$  and rating apply to any one device in the package, unless otherwise noted.)

Rating	Symbol	Value	Unit
Output Voltage	$V_O$	50	V
Input Voltage (Except ULN2801)	$V_I$	30	V
Collector Current – Continuous	$I_C$	500	mA
Base Current – Continuous	$I_B$	25	mA
Operating Ambient Temperature Range	$T_A$	0 to +70	$^\circ\text{C}$
Storage Temperature Range	$T_{stg}$	-55 to +150	$^\circ\text{C}$
Junction Temperature	$T_J$	125	$^\circ\text{C}$

$R_{\theta JA} = 55^\circ\text{C/W}$

Do not exceed maximum current limit per driver.

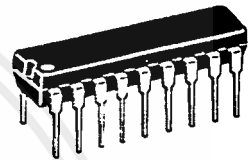
## ORDERING INFORMATION

Device	Characteristics		
	Input Compatibility	$V_{CE}(\text{Max})/I_C(\text{Max})$	Operating Temperature Range
ULN2803A	TTL, 5.0 V CMOS	50 V/500 mA	$T_A = 0$ to +70 $^\circ\text{C}$
ULN2804A	6 to 15 V CMOS, PMOS		

# ULN2803 ULN2804

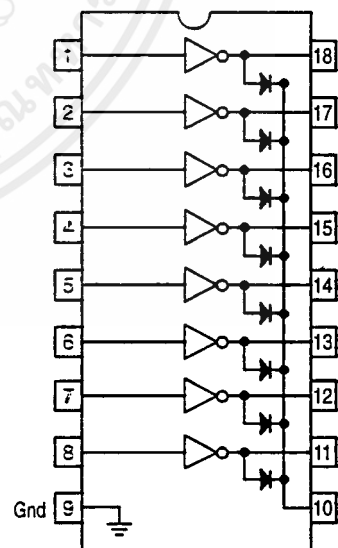
## OCTAL PERIPHERAL DRIVER ARRAYS

### SEMICONDUCTOR TECHNICAL DATA



A SUFFIX  
PLASTIC PACKAGE  
CASE 707

## PIN CONNECTIONS



# ULN2803 ULN2804

## ELECTRICAL CHARACTERISTICS (T<sub>A</sub> = 25°C, unless otherwise noted)

Characteristic		Symbol	Min	Typ	Max	Unit
Output Leakage Current (Figure 1)		I <sub>CEX</sub>				μA
(V <sub>O</sub> = 50 V, T <sub>A</sub> = +70°C)	All Types		–	–	100	
(V <sub>O</sub> = 50 V, T <sub>A</sub> = +25°C)	All Types		–	–	50	
(V <sub>O</sub> = 50 V, T <sub>A</sub> = +70°C, V <sub>I</sub> = 6.0 V)	ULN2802		–	–	500	
(V <sub>O</sub> = 50 V, T <sub>A</sub> = +70°C, V <sub>I</sub> = 1.0 V)	ULN2804		–	–	500	
Collector–Emitter Saturation Voltage (Figure 2)		V <sub>CE(sat)</sub>				V
(I <sub>C</sub> = 350 mA, I <sub>B</sub> = 500 μA)	All Types		–	1.1	1.6	
(I <sub>C</sub> = 200 mA, I <sub>B</sub> = 350 μA)	All Types		–	0.95	1.3	
(I <sub>C</sub> = 100 mA, I <sub>B</sub> = 250 μA)	All Types		–	0.85	1.1	
Input Current – On Condition (Figure 4)		I <sub>I(on)</sub>				mA
(V <sub>I</sub> = 17 V)	ULN2802		–	0.82	1.25	
(V <sub>I</sub> = 3.85 V)	ULN2803		–	0.93	1.35	
(V <sub>I</sub> = 5.0 V)	ULN2804		–	0.35	0.5	
(V <sub>I</sub> = 12 V)	ULN2804		–	1.0	1.45	
Input Voltage – On Condition (Figure 5)		V <sub>I(on)</sub>				V
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 300 mA)	ULN2802		–	–	13	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 200 mA)	ULN2803		–	–	2.4	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 250 mA)	ULN2803		–	–	2.7	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 300 mA)	ULN2803		–	–	3.0	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 125 mA)	ULN2804		–	–	5.0	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 200 mA)	ULN2804		–	–	6.0	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 275 mA)	ULN2804		–	–	7.0	
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 350 mA)	ULN2804		–	–	8.0	
Input Current – Off Condition (Figure 3)	All Types	I <sub>I(off)</sub>	50	100	–	μA
(I <sub>C</sub> = 500 μA, T <sub>A</sub> = +70°C)						
DC Current Gain (Figure 2)	ULN2801	h <sub>FE</sub>	1000	–	–	–
(V <sub>CE</sub> = 2.0 V, I <sub>C</sub> = 350 mA)						
Input Capacitance		C <sub>I</sub>	–	15	25	pF
Turn–On Delay Time (50% E <sub>I</sub> to 50% E <sub>O</sub> )		t <sub>on</sub>	–	0.25	1.0	μs
Turn–Off Delay Time (50% E <sub>I</sub> to 50% E <sub>O</sub> )		t <sub>off</sub>	–	0.25	1.0	μs
Clamp Diode Leakage Current (Figure 6)	T <sub>A</sub> = +25°C	I <sub>R</sub>	–	–	50	μA
(V <sub>R</sub> = 50 V)	T <sub>A</sub> = +70°C				100	
Clamp Diode Forward Voltage (Figure 7)		V <sub>F</sub>	–	1.5	2.0	V
(I <sub>F</sub> = 350 mA)						

# ULN2803 ULN2804

## TEST FIGURES

(See Figure Numbers in Electrical Characteristics Table)

Figure 1.

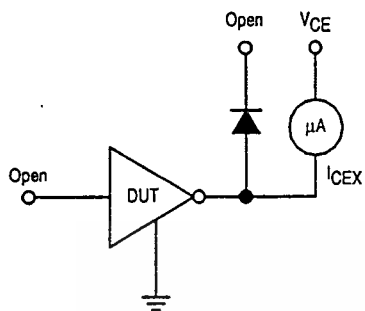


Figure 2.

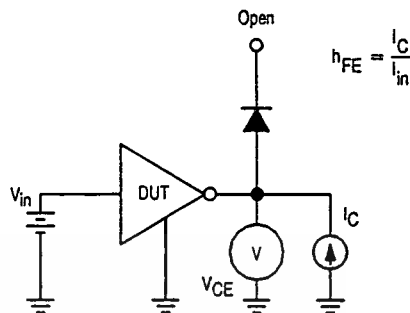


Figure 3.

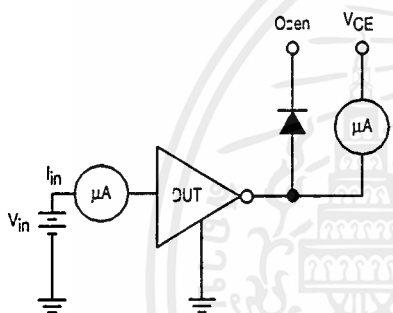


Figure 4.

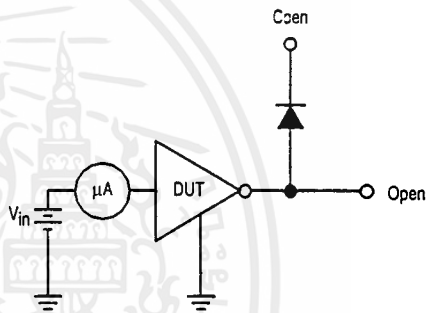


Figure 5.

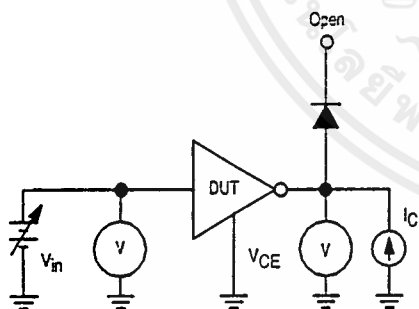


Figure 6.

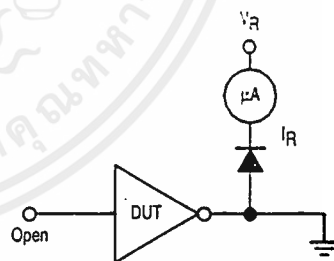
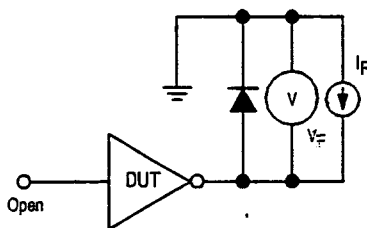


Figure 7.



# ULN2803 ULN2804

TYPICAL CHARACTERISTIC CURVES –  $T_A = 25^\circ\text{C}$ , unless otherwise noted

## Output Characteristics

Figure 8. Output Current versus Saturation Voltage

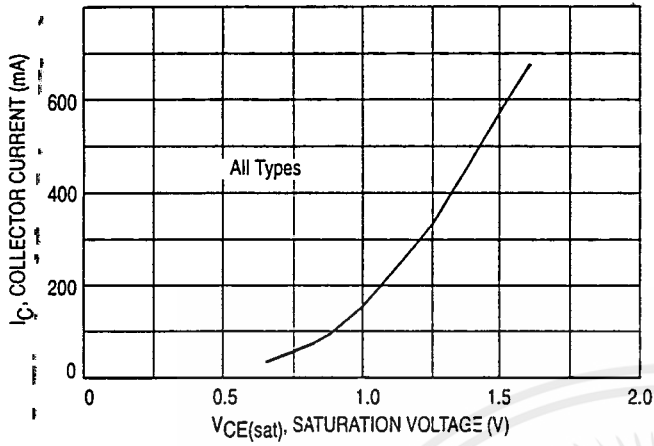
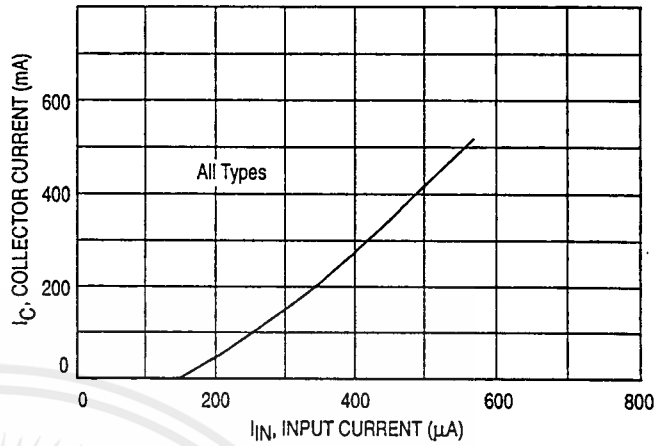


Figure 9. Output Current versus Input Current



## Input Characteristics

Figure 10. ULN2803 Input Current versus Input Voltage

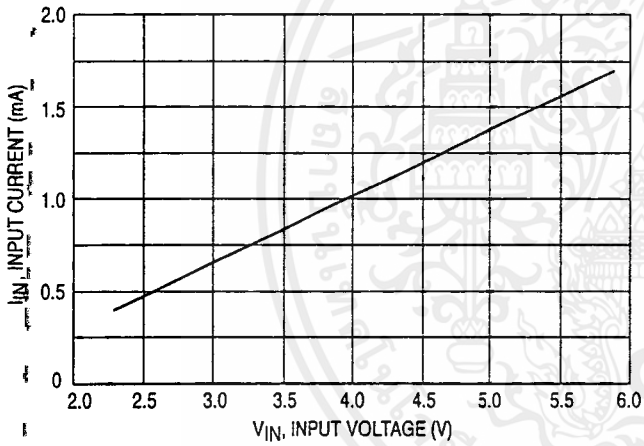


Figure 11. ULN2804 Input Current versus Input Voltage

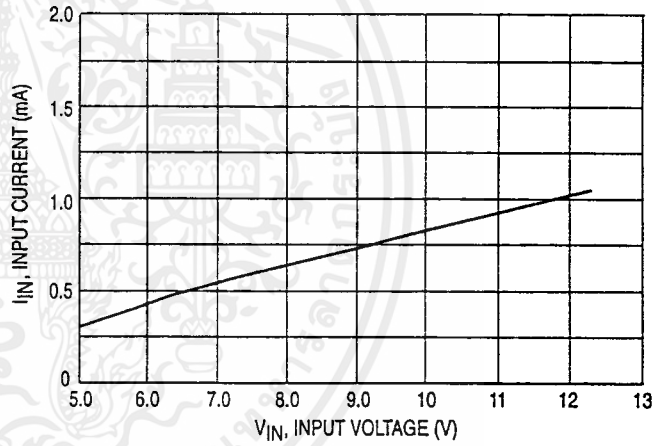
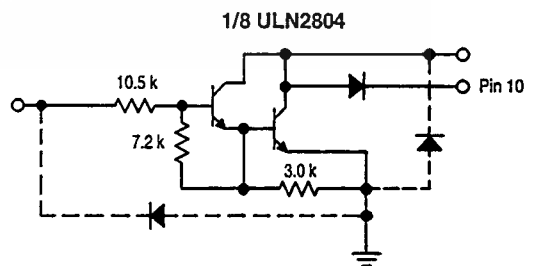
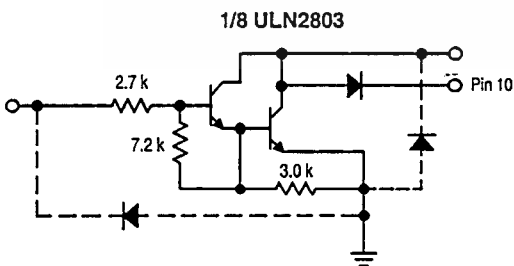


Figure 12. Representative Schematic Diagrams



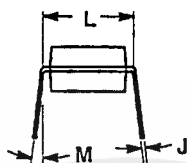
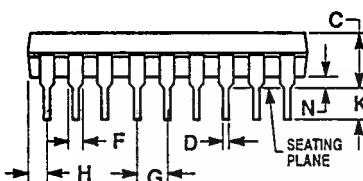
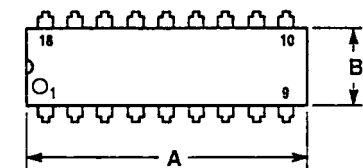
# ULN2803 ULN2804

## OUTLINE DIMENSIONS

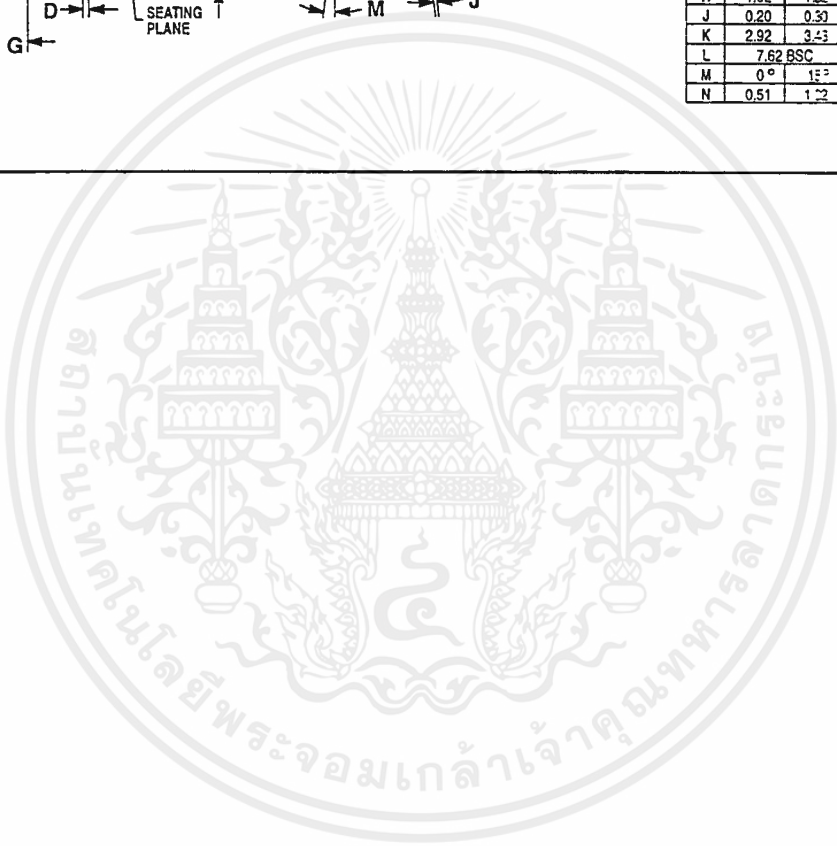
A SUFFIX  
PLASTIC PACKAGE  
CASE 707-02.  
ISSUE C

NOTES:

1. POSITIONAL TOLERANCE OF LEADS (D), SHALL BE WITHIN 0.25 (0.010) AT MAXIMUM MATERIAL CONDITION, IN RELATION TO SEATING PLANE AND EACH OTHER.
2. DIMENSION L TO CENTER OF LEADS WHEN FORMED PARALLEL.
3. DIMENSION B DOES NOT INCLUDE MOLD FLASH.



DIM	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A	22.22	23.24	0.875	0.915
B	6.10	6.50	0.240	0.260
C	3.56	4.57	0.140	0.180
D	0.36	0.56	0.014	0.022
F	1.27	1.73	0.050	0.070
G	2.54 BSC		0.100 BSC	
H	1.02	1.52	0.040	0.060
J	0.20	0.30	0.008	0.012
K	2.92	3.43	0.115	0.135
L	7.62 BSC		0.300 BSC	
M	0°	12°	0°	15°
N	0.51	1.72	0.020	0.040





1. J.N.Holmes "Speech Synthesis and Recognition" Van Nostrand Reinhold Co.Ltd, UK,1998.
2. L.R.Rabiner "Digital Processing of Speech Signals" Prentice-Hall, New Jersey.
3. James L. Flanagan "Speech Analysis Synthesis and Perception" Head-Speech and Auditory Research Department Bell Telephone Laboratories Inc, New Jersey, 1964.
4. Ronald N. Brancewell "The Fourier Transform and its Applications" McGraw-Hill Book Company, USA

