เครื่องอำนวยความสะดวกในการใช้โทรศัพท์โดยใช้เสียง

FACILITIES VIA TELEPHONE BY SPEECH

โดย

นางสาวณัฐชยา ขาวละออ

นายทีฆวัฒน์ ทรงเหรียญชัย

นายเทพฤทธิ์ ยงเกียรติกานต์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

เครื่องอำนวยความสะดวกในการใช้โทรศัพท์โดยใช้เสียง

FACILITIES VIA TELEPHONE BY SPEECH

โดย

นางสาวณัฐชยา	ขาวละออ	39014158
นายทีฆวัฒน์	ทรงเหรียญชัย	39014201
นายเทพฤทธิ์	ยงเกียรติกานต์	39014202

อาจารย์ที่ปรึกษา

รศ.ดร. กอบชัย เดชหาญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2542

ปริญญานิพนธ์ปีการศึกษา 2542

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **เครื่องอำนวยความสะดวกในการใช้โทรศัพท์โดยใช้เสียง**

Facilities Via Telephone By Speech

ผู้จัดทำ

1. นางสาวณัฐชยา	ขาวละออ	39014158
2. นายทีมวัฒน์	ทรงเหรียญชัย	39014201
3. นายเทพฤทธิ์	ยงเกียรติกานต์	39014202

..... อาจารย์ที่ปรึกษา

(รศ.ดร. กอบชัย เดชหาญ)

เครื่องอำนวยความสะดวกในการใช้โทรศัพท์โดยใช้เสียง
FACILITIES VIA TELEPHONE BY SPEECH

โดย นางสาวณัฐชยา ขาวละออ 39014158
นายทีฆวัฒน์ ทรงเหรียญชัย 39014201
นายเทพฤทธิ์ ยงเกียรติกานต์ 39014202

อาจารย์ที่ปรึกษา รศ.ดร. กอบชัย เดชหาญ

บทคัดย่อ

โครงงานนี้มีจุดประสงค์เพื่อที่จะเพิ่มประสิทธิภาพในการเชื่อมต่อระหว่างคอมพิวเตอร์ส่วนบุคคลกับโทรศัพท์ เพื่ออำนวยความสะดวกในชีวิตประจำวัน โดยเป็นการเขียนโปรแกรมเพื่อให้สามารถใช้เสียงในการต่อหมายเลขโทรศัพท์ไปยังหมายเลขปลายทางที่ต้องการได้ โดยจะแบ่งออกเป็น 2 แบบ คือ

1. พูดหมายเลขโทรศัพท์ ที่ต้องการติดต่อโดยโปรแกรมจะทำการรับเสียงจากไมโครโฟนและทำการตัดเสียงก่อนที่จะนำไปวิเคราะห์ว่าเป็นโมเดลเสียงหมายเลขใด แล้วจึงทำการสั่งโมเด็มให้ติดต่อไปยังหมายเลขปลายทางที่ต้องการ

2. พูดชื่อ ที่มีอยู่ในฐานข้อมูลโดยโปรแกรมจะทำเหมือนที่พูดหมายเลขแต่เมื่อวิเคราะห์ได้แล้วว่าเป็นโมเดลใด ก็จะนำไปเปรียบเทียบกับฐานข้อมูลหมายเลขโทรศัพท์ที่มีอยู่ว่าเป็นหมายเลขใด แล้วจึงทำการสั่งให้โมเด็มให้ติดต่อไปยังหมายเลขปลายทางที่ต้องการ

Abstract

The propose of this project is to optimize the ability of telephone and computer. This project is about programming for using voice to dial telephone via external modem to desired destination. All of programs are built from Visual C++. There are 2 methods for using this project ,including

1. By speaking telephone number. After speaking, the program will get the voices from microphone and record them then the program will separate the voices into individual words. Each individual word will be proceeded and transferred to a model of voice that will be compared with databases. After the program can recognize every individual word , it will have modem dial the recognized number to desired destination

2. By speaking the name of receiver . This method is similar to the former. After analized and transferred into the model of voice, the word will be compared with every database. The program will recognize the word with the database that has the nearest model . Finally the program will have the modem dial telephone with the number that matches with the recognized name of the receiver.

สารบัญ

สารบัญรูปภาพ

สารบัญตาราง

บทที่ 1
บทนำ

เนื่องจากความเจริญก้าวหน้าทางเทคโนโลยีในปัจจุบัน ทำให้การติดต่อสื่อสารเป็นไปได้อย่างสะดวกรวดเร็วและประหยัด ทำให้อุปกรณ์อำนวยความสะดวกมีราคาถูก ดังนั้นเราจึงสามารถนำเทคโนโลยีใหม่ ๆ เข้ามาช่วยอำนวยความสะดวกในชีวิตประจำวันของเราได้

ในอดีตมีการวิเคราะห์ว่าการสอนให้คอมพิวเตอร์สามารถรู้จำเสียงพูดของมนุษย์นั้นทำไม่ได้ เนื่องจากการพูดของมนุษย์มีความซับซ้อนและมีความแตกต่างกันในแต่ละบุคคล ความเร็วในการพูด รวมถึงเสียงสูงต่ำของคนแต่ละคน แต่ก็มีการวิจัยเพื่อให้คอมพิวเตอร์สามารถรู้จำเสียงพูดเรื่อยมา จนในปัจจุบันการรู้จำเสียงพูดสามารถจะใช้งานได้ดี และมีการนำไปใช้กับอุปกรณ์ต่าง ๆ เช่น ATM (Automatic Teller Machine) และเครื่องตอบรับโทรศัพท์อัตโนมัติ ถึงแม้คอมพิวเตอร์จะไม่สามรถรู้จำเสียงได้เท่ากับมนุษย์ แต่เราสามารถสอนให้คอมพิวเตอร์รู้จำได้ในขีดจำกัดระดับหนึ่งโดยมีการกำหนดขอบเขตของการรู้จำ เช่น จำกัดคำที่จะสามารถรับรู้ได้

โครงงานนี้ได้มีการศึกษาและทดลองนำเอาส่วนของการรู้จำเสียงที่เป็นทฤษฎีมาใช้งานจริง โดยรูปแบบการใช้เสียงสั่งคอมพิวเตอร์ให้ต่อหมายเลขโทรศัพท์ไปยังหมายเลขที่ต้องการติดต่อได้ซึ่งจะได้กล่าวต่อไป

วัตถุประสงค์โครงการ

1. ศึกษารายละเอียดเกี่ยวกับสัญญาณต่าง ๆ ของคู่สายโทรศัพท์และโมเด็ม เพื่อทำการเชื่อมต่อคู่สายโทรศัพท์ผ่านโมเด็มเข้าเครื่องคอมพิวเตอร์
2. ศึกษาการวิเคราะห์เสียงพูดของมนุษย์
3. นำงานวิจัยเกี่ยวกับการรู้จำเสียงภาษาไทยที่ได้มีผู้พัฒนาไว้ในระดับหนึ่งแล้วมาทำการศึกษาและพัฒนาต่อโดยจะเน้นพิจารณาในด้านการนำไปใช้งานจริง
4. เพื่อพัฒนาโปรแกรมส่วนต่าง ๆ ที่ทำให้คอมพิวเตอร์สามารถรู้จำเสียงพูด และนำโปรแกรมมาใช้งานได้ เช่น แยกคำพูดของมนุษย์ออกเป็นคำเดี่ยว ๆ ได้
5. พัฒนาวิธีการทางทฤษฎีให้สามารถนำมาใช้ในทางปฏิบัติได้ เช่น หาวิธีในการที่จะช่วยเพิ่มความเร็วในการวิเคราะห์เสียง หรือปรับปรุงขั้นตอนบางอย่างเพื่อลดความผิดพลาดในการวิเคราะห์ เป็นต้น
6. นำโปรแกรมการรู้จำเสียงมาประยุกต์ใช้ในการต่อหมายเลขโทรศัพท์ไปยังปลายทางที่ต้องการ โดยใช้เสียงพูดในการต่อโทรศัพท์ผ่านทางเครื่องคอมพิวเตอร์

ขอบเขตของโครงการ

เพื่อให้สามารถ “ใช้เสียงสั่งคอมพิวเตอร์ให้โทรศัพท์ผ่านโมเด็มไปยังหมายเลขที่ต้องการได้” ซึ่งมีขอบเขตพอสังเขปดังนี้

1. รับเสียงพูดผ่านทางไมโครโฟนที่ต่อมาจาก Sound Card ของเครื่องคอมพิวเตอร์แล้วนำมาประมวลผล และส่งผลที่ได้ไปสั่งโมเด็มให้ทำการโทรศัพท์ไปยังหมายเลขปลายทางนั้น

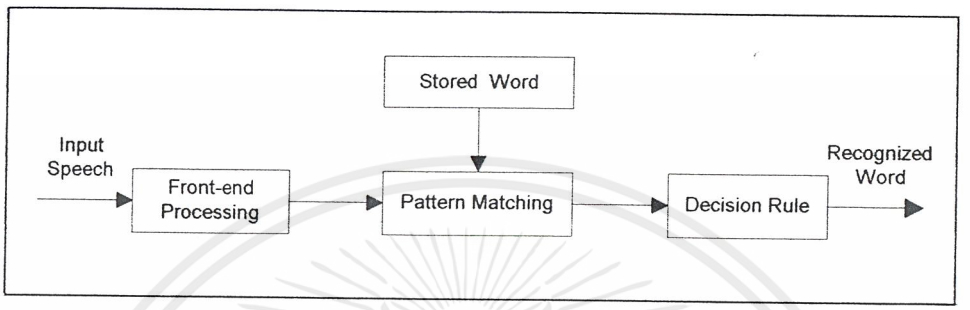
2. เสียงที่ใช้สั่งนั้นในขั้นต้นจะมีการเก็บตัวอย่างเสียงของคำนั้น ๆ ที่มากและหลากหลายพอไว้ก่อนเพื่อผลการรับรู้ที่ถูกต้อง

3. เป็นการวิเคราะห์เสียงพูดแบบต่อเนื่อง โดยที่เราสามารถแบ่งช่วงการวิเคราะห์สัญญาณออกเป็นช่วง ๆ เป็นคำเดี่ยวได้ ซึ่งในที่นี้จะแบ่งออกเป็น 2 กรณี คือพูดหมายเลขปลายทาง ซึ่งจะทำการตัดคำออกเป็นตัวเลขเดี่ยว ๆ และสั่งให้โมเด็มติดต่อไปยังหมายเลขนั้น ๆ และพูดชื่อที่ได้ทำการบันทึกเป็นฐานข้อมูลเอาไว้แล้ว โดยจะทำการวิเคราะห์ว่าเป็นชื่อใดในฐานข้อมูล แล้วทำการส่งหมายเลขโทรศัพท์ของชื่อนั้นให้โมเด็มติดต่อออกไป

บทที่ 2
ทฤษฎีและหลักการ

2.1 หลักการวิเคราะห์เสียง

จากการศึกษารูปแบบการรู้จำเสียงพูดแบบคำเดี่ยว (Isolated word recognition) มีรูปแบบและหลักการพื้นฐานดังรูปที่ 2.1



รูปที่ 2.1 แสดง Basic isolated-word recognition system

สัญญาณที่เข้าโมเดลเป็นคลื่นเสียง ส่วนผลลัพธ์เป็นเสียงวิเคราะห์ที่ได้จากสัญญาณเข้า โมเดลนี้มีการใช้กันอย่างแพร่หลาย แบ่งเป็นส่วนต่าง ๆ ดังนี้

- Front – end processing เป็นขั้นตอนการวิเคราะห์เสียงที่ได้รับมาให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ได้
- Rattern matching เป็นขั้นตอนการสร้างโมเดลของเสียง
- Decision rule เป็นขั้นตอนการตัดสินใจในการเลือกโมเดลเสียงที่ใกล้เคียงกับเสียงทดสอบมากที่สุด

2.1.1 Front – end processing

Front – end processing เป็นขั้นตอนที่ทำการแปลงข้อมูลที่มีอยู่มากมายเป็นส่วนเล็ก ๆ ที่สามารถแสดงคุณสมบัติของคลื่นเสียงนั้น ๆ โดยผ่านขั้นตอน ดังนี้

2.1.1.1 การประมาณเชิงเส้น (Linear predictive coding : LPC)

2.1.1.2 การจัดระดับเวคเตอร์ (Vector Quantization : VQ)

2.1.1.1 การประมาณเชิงเส้น (Linear predictive coding)

เป็นการวิเคราะห์เพื่อหาค่าพารามิเตอร์ที่เหมาะสมเป็นข้อมูลในการวิเคราะห์เสียง โดยแบ่งสัญญาณเสียงพูดที่จะวิเคราะห์ออกเป็นส่วน ๆ แต่ละส่วนใช้ระยะเวลาช่วงสั้น ๆ ประมาณ 15-20 มิลลิวินาที ซึ่งช่วงนี้สัญญาณเสียงพูดจะมีการเปลี่ยนคุณลักษณะอย่างช้า ๆ จนอาจถือว่าระบบกำเนิดเสียงมีคุณลักษณะที่คงที่ (stationary)

ก่อนที่จะนำข้อมูลเสียงพูดมาทำการประมาณเชิงเส้นนั้นจะต้องทำการปรับแต่งข้อมูลให้เหมาะสม โดยการตัดส่วนหัวและส่วนท้ายคำพูดที่เป็นส่วนเกินออก เนื่องจากเป็นส่วนของสัญญาณรบกวน จากนั้นจึงนำข้อมูลการผ่านขั้นตอนส่วนต่าง ๆ ดังนี้

2.1.1.1.1 การพรีเอมฟาซิส (pre – emphsis)

2.1.1.1.2 การแบ่งช่วงสัญญาณ (frame blocking)

2.1.1.1.3 การวินโดว์ (windowing)

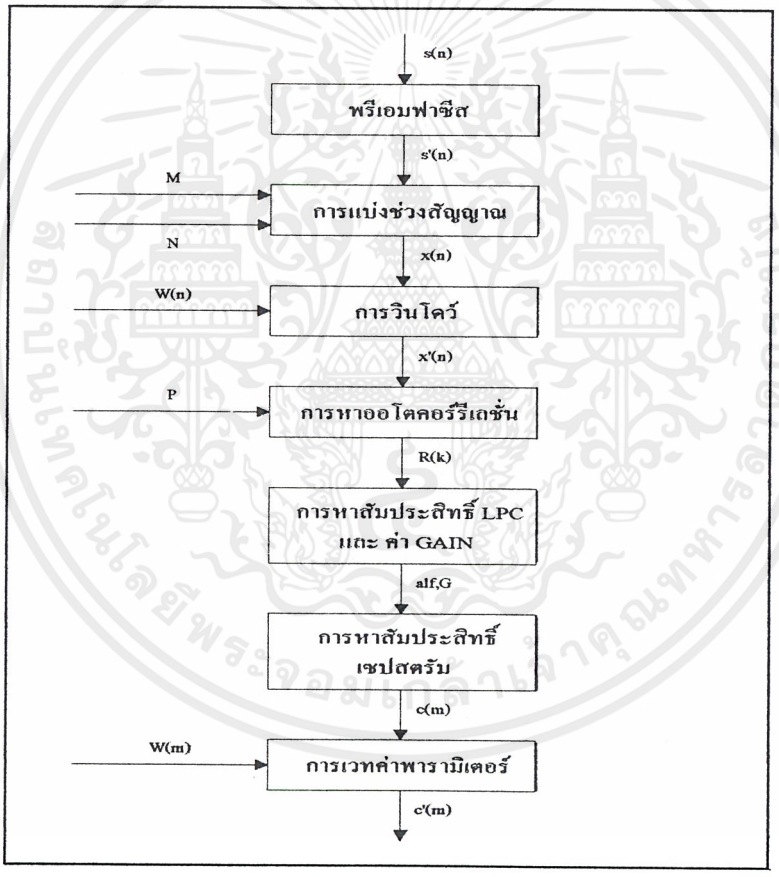
2.1.1.1.4 การวิเคราะห์ออโตคอร์รีเลชัน (autocorrelation analysis)

2.1.1.1.5 การหาอัตราการขยาย G

2.1.1.1.6 การสัมประสิทธิ์เซปสตรัม (cepstrum)

2.1.1.1.7 การเวทค่าพารามิเตอร์ (paramiter weighting)

ดังแสดงในรูปที่ 2.2



รูปที่ 2.2 แสดงขั้นตอนการเตรียมสัญญาณในการวิเคราะห์

การประมาณเชิงเส้น จะทำการประมาณค่าสัญญาณจากผลรวมเชิงเส้นของสัญญาณก่อนหน้า โดยใช้หลักผลรวมกำลังสองของความคลาดเคลื่อนที่มีค่าต่ำสุด ซึ่งการประมาณเชิงเส้นมีอยู่หลายวิธีได้แก่

- วิธีโควาเรียนซ์ (Co-variance Method)

- วิธีออโตคอร์รีเลชั่น (Auto-correlation Method)
- วิธีแลตทิส (Lattice Method)

และอื่น ๆ อีกหลายวิธี แต่วิธีที่นิยมใช้คือวิธีออโตคอร์รีเลชั่น หรือ วิธีอัตตสัมพันธ์ (Auto-correlation) หลังจากผ่านขั้นตอนดังกล่าวจะได้ค่าสัมประสิทธิ์ LPC (α) และอัตราการขยาย (G) ซึ่งในแต่ละขั้นตอนอธิบายได้ดังนี้

2.1.1.1.1 การพรีเอมฟาซิส (Preemphasis)

เนื่องจากสัญญาณเสียงพูดของมนุษย์ มีองค์ประกอบส่วนใหญ่อยู่ในช่วงความถี่ต่ำ เมื่อเทียบกับแถบความถี่ที่ปฏิบัติงาน (bandwidth) ไม่เกิน 5 กิโลเฮิรตซ์ ดังนั้นเพื่อให้อัตราส่วนเสียงต่อสัญญาณรบกวน (siglnal to noise : SNR) มีค่าค่อนข้างคงที่ ตลอดช่วงความถี่ที่ปฏิบัติงานนี้ เราจึงต้องมีการพรเอมฟาซิส เพื่อเน้นความพี่สูงให้มีขนาดสูงขึ้น โดยการกรองสัญญาณด้วยวงจรกรองความถี่สูง (high pass filter) ซึ่งจะใช้วงจรกรองดิจิตอลแบบ first order มีรูปแบบสมการดังนี้

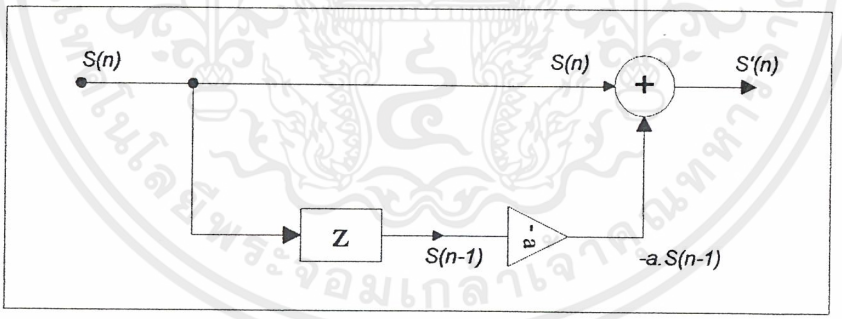
$$y(n) = a_{\sigma}x(n) - b_1 y(n-1) \qquad (2.1)$$

มีฟังก์ชันถ่ายโอนเป็น

$$H(z) = 1 - a.z^{-1} \; ; \; 0.9 < a < 1.0 \qquad (2.2)$$

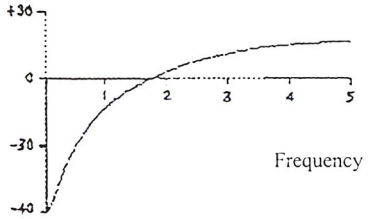
สมมติว่าสัญญาณเดิมเป็น S(n) เมื่อประมาณค่าสัญญาณแล้วจะเป็น S'(n)

จะได้ว่า $S'(n) = S(n) - a.S(n-1) \qquad (2.3)$



รูปที่ 2.3 แสดงวงจรกรองความถี่สูงผ่าน

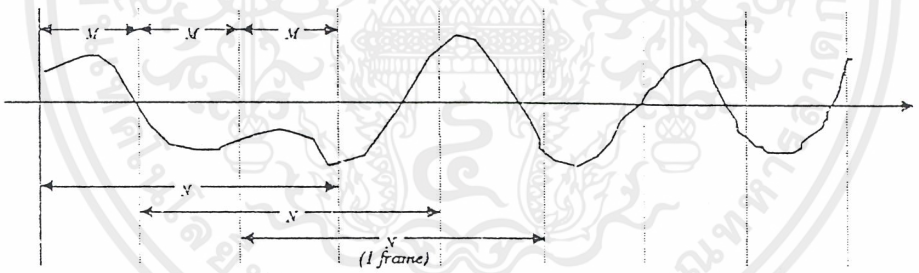
ยิ่งค่า α ใกล้ 1 เท่าใดความถี่สูงก็จะขยายมากขึ้นเท่านั้น ค่า α ที่ควรใช้การพรีเอมฟาซิสคือ 0.9375 ถ้านำฟังก์ชันการพรีเอมฟาซิสพล็อตกราฟของ ขนาด กับ ความถี่จะได้กราฟดังรูป



รูปที่ 2.4 แสดงกราฟแอมปลิจูดกับความถี่ เมื่อ $\alpha = 0.9375$

2.1.1.1.2 การแบ่งช่วงสัญญาณ (block into frames)

สัญญาณที่ผ่านการพรีเอมฟาซิสแล้วจะถูกตัดแบ่งออกเป็นช่วง ๆ หรือ เฟรมช่วงละ N ตัวอย่างสัญญาณ การวิเคราะห์จะวิเคราะห์ทีละช่วงของแต่ละ N ตัวอย่างสัญญาณ โดยช่วงในการวิเคราะห์แต่ละช่วงจะถูกเลื่อนไปเป็นระยะ M ช่วงสัญญาณ จะเห็นว่าถ้าค่า M โตกว่าค่า N ในการเลื่อนของช่วงในการวิเคราะห์ ก็จะเป็นการสูญเสียส่วนหนึ่งทำให้ผลที่ได้ไม่ถูกต้องเท่าที่ควร ถ้าค่า M เล็กกว่า N ก็จะทำให้ตัวอย่างสัญญาณทุกตัวถูกนำมาวิเคราะห์ ยิ่งค่า M เล็กเท่าใด ความแม่นยำในการวิเคราะห์ก็จะสูงยิ่งขึ้นเท่านั้น แต่ก็จะทำให้การคำนวณช้าลง

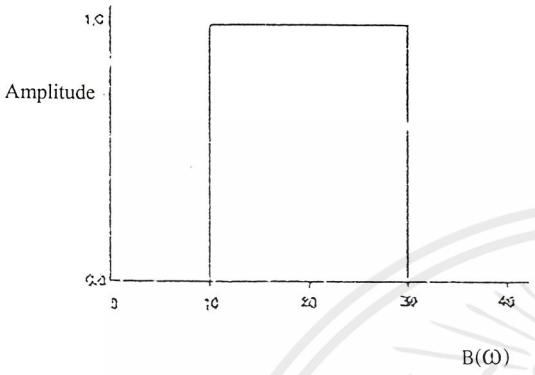


รูปที่ 2.5 แสดงการแบ่งช่วงสัญญาณ

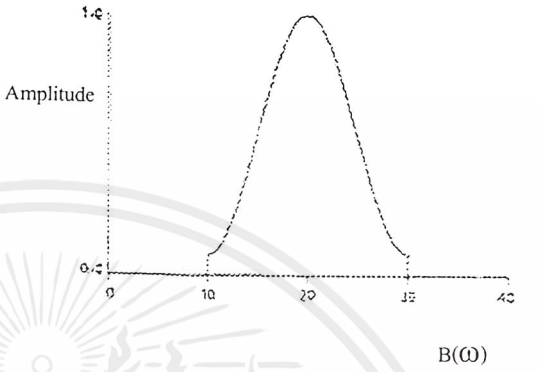
2.1.1.1.3 การวินโดว์ (Windowing)

พิจารณาช่วงสัญญาณ N ตัวอย่างสัญญาณของช่วงใด ๆ ที่ตัดมาวิเคราะห์จะเห็นว่าที่ขอบของเฟรมที่ตัดมานี้มีความไม่ต่อเนื่องของสัญญาณ ถ้ามองในโดเมนความถี่ที่สูงเหล่านี้ เราจะคูณด้วยฟังก์ชันวินโดว์ เพื่อลดความไม่ต่อเนื่องของสัญญาณที่ขอบ และไม่ทำให้สเปคตรัมของสัญญาณในช่วงความถี่ต่ำเปลี่ยนไปมากนัก

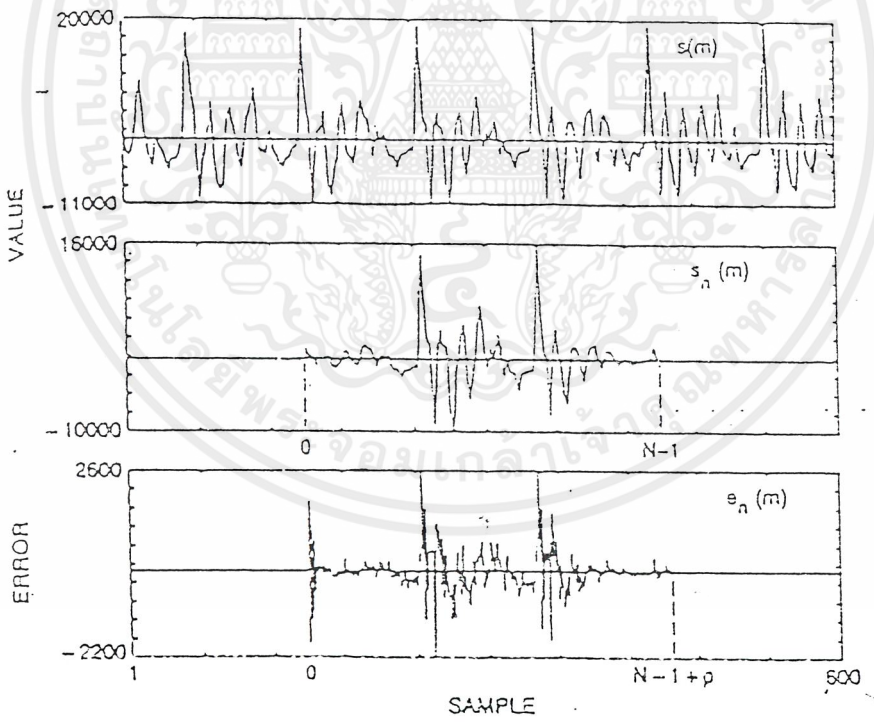
หลักการกำหนดขนาดของวินโดว์ ฟังก์ชันวินโดว์ที่ใช้ในการรวมกับสัญญาณมีหลายชนิด เช่น วินโดว์แบบสี่เหลี่ยม (rectangular window) มีลักษณะดังรูปที่ 2.6 สำหรับฟังก์ชันวินโดว์ที่เหมาะสมที่ใช้กันคือวินโดว์แบบแฮมมิ่ง (Hamming Window) ซึ่งมีลักษณะดังรูปที่ 2.7



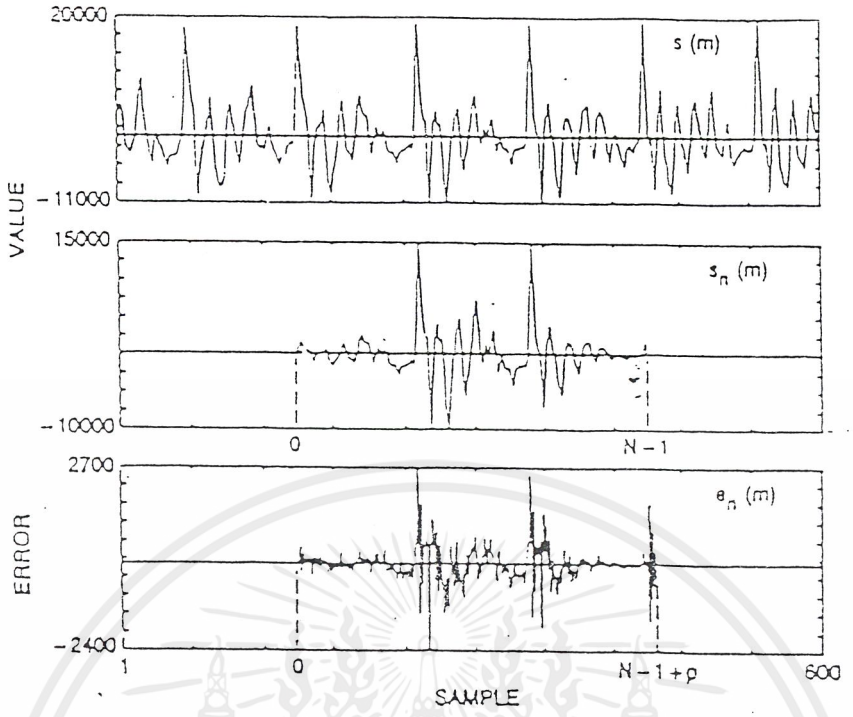
รูปที่ 2.6 แสดง Rectangular window function



รูปที่ 2.7 แสดง Hamming window function



รูปที่ 2.8 แสดงการเกิดความไม่ต่อเนื่องของสัญญาณที่ขอบส่วนต้นของเฟรมที่ตัดมาวิเคราะห์



รูปที่ 2.9 แสดงการเกิดความไม่ต่อเนื่องของสัญญาณที่ขอบส่วนท้ายของเฟรมที่ตัดมาวิเคราะห์

ฟังก์ชันวินโดว์แฮมมิ่งมีสมการดังนี้

$$w(n) = 0.54 - 0.4\cos(2\pi n / (N-1) \quad \text{เมื่อ } n = 0,1,2....N-1 \qquad (2.4)$$

ในการวิเคราะห์เสียงโดยใช้ฟังก์ชันวินโดว์ จะพบว่าสัญญาณเสียง ที่ผ่านการกรองโดยวินโดว์นั้นจะมีการแกว่งขึ้นลงมากน้อยขึ้นกับช่วงเวลาของวินโดว์ (ความกว้างของวินโดว์) คือถ้าช่วงเวลาของการวินโดว์สั้น จะมีการแกว่งขึ้นลงอย่างรวดเร็ว และถ้าช่วงเวลาของการวินโดว์มากจะมีการแกว่งขึ้นลงช้า ๆ ดังนั้นในการเลือกช่วงเวลาของการวินโดว์ ต้องให้อยู่ในช่วงที่เหมาะสม คือไม่ให้เอาท์พุทของสัญญาณแกว่งช้าหรือเร็วเกินไป อยู่ในช่วงระหว่าง 10 – 30 ms เมื่อคูณกับฟังก์ชันวินโดว์แล้ว ก็จะได้

$$x'(n) = w(n)x(n) \qquad (2.5)$$

2.1.1.1.4 การวิเคราะห์ออโตคอร์รีเลชัน (auto – correlation)

สมมติว่าสัญญาณเดิมเป็น S(n) การประมาณ ค่าสัญญาณเป็น S'(n) ดังนั้นสามารถอธิบายการประมาณเชิงเส้นด้วยสมการต่อไปนี้

$$s'(n) = \sum_{k=1}^{p} a_k S(n-k) \qquad (2.6)$$

เมื่อ α_k เป็นค่าคงที่ เรียกวิธีการนี้ว่าการประมาณเชิงเส้นอันดับ p โดยมีเงื่อนไขว่า ค่า α_k ที่ใช้ในการประมาณจะต้องทำให้ ผลรวมของกำลังสองของความคลาดเคลื่อน $\{s(n) - s'(n)\}^2$ มีค่าน้อยที่สุด นั่นคือ $\sum e^2(n) = \sum\{s(n)-s'(n)\}^2$ มีค่าต่ำที่สุด ซึ่งจะเป็นการประมาณเชิงเส้นวิธีออโตคอร์รีเลชั่น (Autocorrelation Method) หรือวิธีอัตตสัมพันธ์

การคำนวณออโตคอร์รีเลชั่นเป็นวิธีการหาสัมประสิทธิ์ LPC โดยฟังก์ชั่นออโตคอร์รีเลชั่น ซึ่งเป็นการเปรียบเทียบสัญญาณกับสัญญาณของตัวมันเองที่ถูกเลื่อนไปตามแกนเวลา ที่ใช้วิธีนี้เนื่องจากเป็นการคำนวณที่มีการแก้สมการที่น้อยกว่าวิธีอื่น ๆ และมีความปน่นอนในด้านเสถียรภาพ อีกทั้งมีการเก็บข้อมูลที่น้อยกว่า

การประมาณเชิงเส้นอันดับ p ของอันดับสัญญาณ s(n) และผลรวมเชิงเส้นของสัญญาณนี้ที่ถูกกำหนดด้วยการถ่วงน้ำหนัก α_1 ถึง α_p จะใช้ในการประมาณค่าของสัญญาณถัดไป $s'(n)$ เขียนได้ในรูปสมการ

$$S'(n) = \sum_{k=1}^{p} \alpha_k S(n-k) \qquad ;10 \le p \le 26 \tag{2.7}$$

ผลต่างของค่าสัญญาณประมาณ S'(n) กับ ค่าของสัญญาณปัจจุบันเรียกว่าค่าความคลาดเคลื่อน e(n) (residual signal)

$$e(n) = s(n) - s'(n) \tag{2.8}$$

การวิเคราะห์การประมาณเชิงเส้นคือการหาค่า α_k ที่ทำให้ค่ากำลังสองของความคลาดเคลื่อนมีค่าน้อยที่สุด ซึ่งสามารถหาได้จากสมการ

$$R(k) = \sum_{n=0}^{N-1-k} S(n)S(n+k) \quad ; \quad 0 \le k \le p \tag{2.9}$$

และแทนค่าในเมตริกซ์เพื่อหา α_k

$$\begin{bmatrix} R_n(0) & R_n(1) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & \cdots & R_n(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_{n(p-2)} & \cdots & R_n(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix} \tag{2.10}$$

หรือ $R_n \alpha = r_n$ (2.11)

เมื่อ $$R_n = \begin{bmatrix} R_n(0) & R_n(1) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & \cdots & R_n(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_{n(p-2)} & \cdots & R_n(0) \end{bmatrix}; \quad \alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix}, r_n = \begin{bmatrix} R_n(1) \\ R_n(2) \\ \vdots \\ R_n(p) \end{bmatrix}$$

2.1.1.1.5 การหาอัตราการขยาย G

อัตราการขยายสามารถหาได้โดยตรงจากสมการ

$$G^2 = \frac{R_n(0) - \sum_{k=1}^{p} \alpha_k R_n(k)}{\sum_{m=0}^{N-1} u^2(m)} \tag{2.12}$$

2.1.1.1.6 การสัมประสิทธิ์เซปสตรัม (Cepsturm)

หลังจากที่หาสัมประสิทธิ์ LPC และ Gain ใน 1 เฟรมแล้ว จะเปลี่ยนให้เป็นสัมประสิทธิ์เซปสตรัมเนื่องจากการรู้จำเสียงพูดนั้น สัมประสิทธิ์เซปสตรัมนี้เป็นพารามิเตอร์ที่มีลักษณะน่าเชื่อถือได้ดีกว่าสัมประสิทธิ์ LPC ทั้งยังมีความสัมพันธ์ใกล้ชิดกับการรับรู้เสียง ตามความรู้สึกของมนุษย์โดยแท้จริง สัมประสิทธิ์เซปสตรัมสามารถหาได้โดยตรงจากสัมประสิทธิ์ LPC ดังนี้

$C_0 = \ln G$ เป็นสัมประสิทธิ์ตัวแรก ซึ่งเป็นแกน (2.13)

$Q \approx 3p/2$ โดย $p = 12$ ดังนั้น $Q = 18$ รวมกับเกน (C_0)

จะได้ว่า สัมประสิทธิ์เซปสตรัมใน 1 เฟรม = 19 ตัว

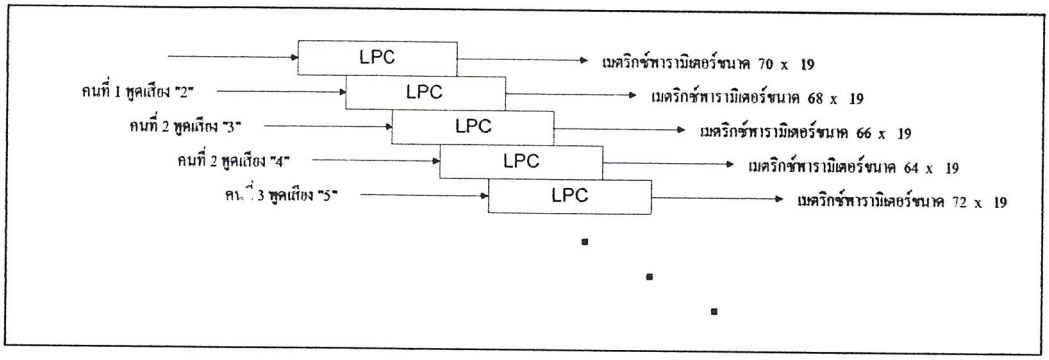
$$C_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \quad , 1 \le m \le p \tag{2.14}$$

$$C_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \quad , m \ge p \tag{2.15}$$

2.1.1.1.7 การเวทค่าพารามิเตอร์ (parameter weighting)

เนื่องจากสัมประสิทธิ์เซปสตรัมที่ได้นั้น ช่วงลำดับต้น ๆ และลำดับท้าย ๆ ของเฟรมที่นำมาวิเคราะห์จะเกิดความคลาดเคลื่อนมากกว่าบริเวณส่วนอื่น เพราะฉะนั้น จึงทำการถ่วงน้ำหนัก เพื่อลดค่าความคลาดเคลื่อนดังกล่างนี้ ด้วยฟังก์ชั่นเวทติ้ง ดังนี้คือ

$$W_m = \left[1 + \frac{Q}{2} \sin\left(\frac{\pi m}{Q}\right)\right] \quad , 1 \le m \le Q \tag{2.16}$$



รูปที่ 2.10 แสดงลักษณะการหาค่าพารามิเตอร์

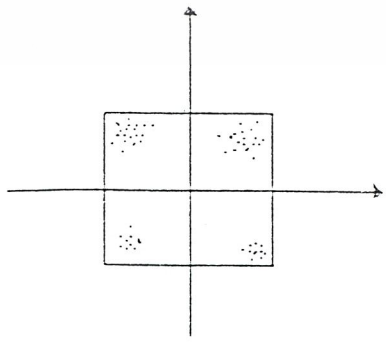
จะได้พารามิเตอร์สุดท้าย คือ

$$C_m = C_m * W_m \tag{2.17}$$

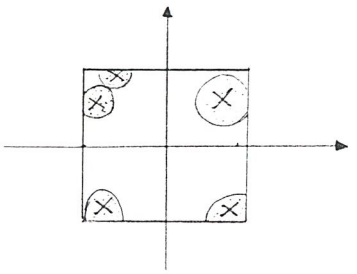
จากนั้นก็พิจารณาให้ครบทุกเฟรมของข้อมูล เมื่อพิจารณาเรียบร้อยแล้วก็จะนำไปจัดกลุ่มเสียงและสร้างแบบจำลองเสียงเพื่อใช้ในการเปรียบเทียบต่อไป

2.1.1.2 การจัดระดับเวคเตอร์ (Vector Quantization)

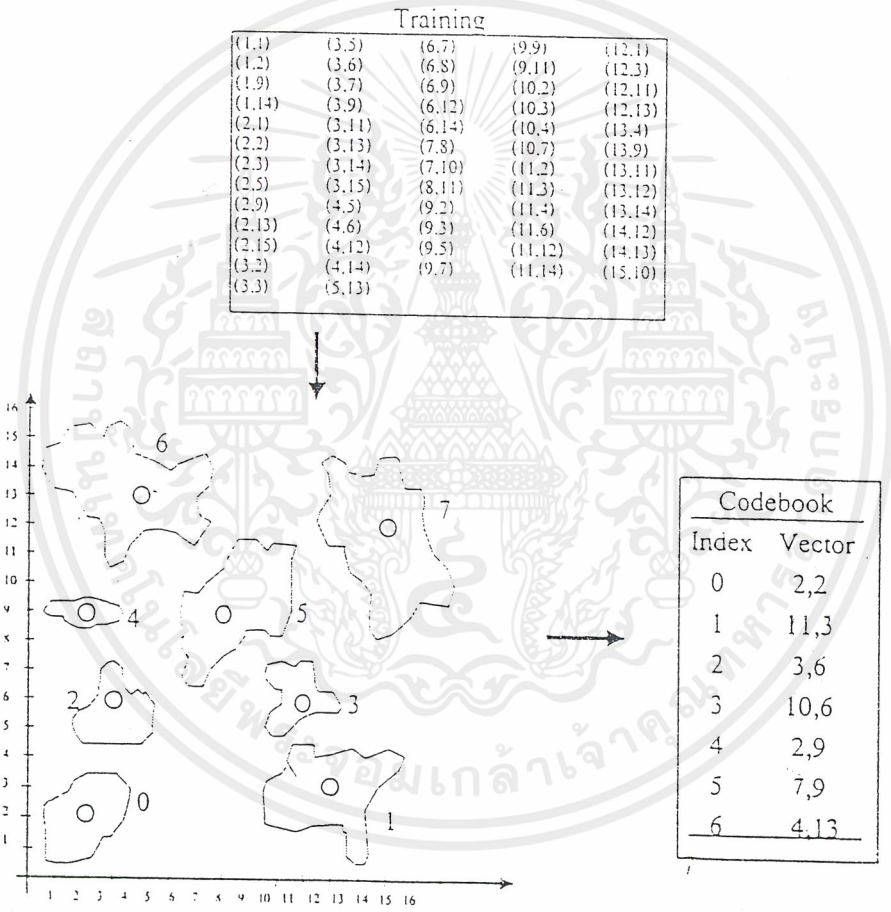
เวคเตอร์ ควอนไตเซชั่น เป็นวิธีการลดไดเมนชั่น (dimension) หรือจำนวนของข้อมูล เวคเตอร์อินพุทหรือเซตเทรนนิ่ง หรือพารามิเตอร์ที่ได้จากขั้น LPC จะถูกเลือกมากลุ่มหนึ่งซึ่งใช้เป็นตัวแทนของข้อมูลนำนวนหนึ่งหรือเรียกว่าการหา Codebook อินพุทที่เข้ามาจะถูกทำการเปรียบเทียบกับ codebook ที่มีอยู่ โดยจะพิจารณาว่าอินพุทที่เข้ามานั้นห่างจาก codebook ใดน้อยที่สุด อินพุทดังกล่าวจะถูกนำมาหาจุดศูนย์กลางร่วมใหม่ และนำจุดศูนย์กลางที่ได้นี้ไปทำการหาความคลาดเคลื่อนกับสมาชิกทุกตัว ถ้าค่าความคลาดเคลื่อนที่ได้มีค่ามากกว่าค่าที่กำหนดไว้ค่าหนึ่งหรือค่าที่ยอมรับได้ ก็จะนำศูนย์กลางใหม่นั้นไปเป็น codebook แทน และจะทำการจัดกลุ่มอินพุทเข้ากับ codebook ใหม่ที่ได้และหาความคลาดเคลื่อนอีกครั้งทำอย่างนี้ซ้ำๆ จนกระทั่งค่าความคลาดเคลื่อนมีค่าน้อยถึงค่าที่ยอมรับได้ก็จะถือว่าได้ codebook ที่ดีที่สุดที่จะเป็นตัวแทนของอินพุททั้งหมด จะสังเกตุได้ว่าทุกครั้งที่มีการหา codebook ใหม่นั้น ค่าความคลาดเคลื่อนที่ได้จะมีค่าลดลงทุกครั้งด้วย



รูปที่ 2.11 แสดงการกระจายเฟรมของเสียงพูดแต่ละจุดแทนเฟรมของเสียง

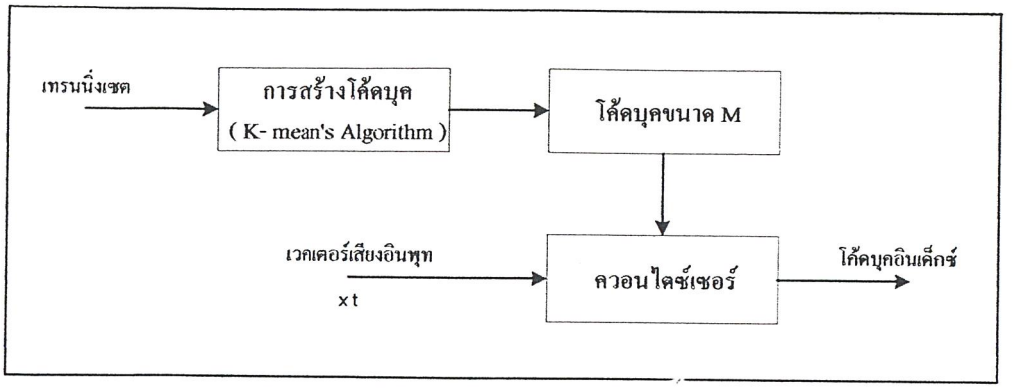


รูปที่ 2.12 แสดงการรวมกลุ่มของเฟรมเสียงเพื่อสร้าง codebook X
แทนเวคเตอร์ศูนย์กลางเพื่อแบ่งแยกเวคเตอร์



รูปที่ 2.13 แสดงตัวอย่างการหา Codebook

ตัวอย่างเวคเกอร์ควอนไตเซชั่น สมมติให้เวกเตอร์แต่ละตัวมี 2 มิติ แบะทำการหา Codebook ขนาด 8 เวกเตอร์



รูปที่ 2.14 แสดงบล็อกไดอะแกรมของเวกเตอร์ควอนไตซ์

เวกเตอร์ทั้งหมดจะถูกจัดเข้าไปในกลุ่มของ Codebook ต่าง ๆ แล้วทำการหาจุดศูนย์ใหม่โดยการเฉลี่ยค่าเวกเตอร์สมาชิกทุกตัวที่อยู่ในกลุ่มเดียวกัน ผลที่ได้คือ Codebook 8 ตัวที่เป็นตัวแทนของเวกเตอร์ทั้งหมด

การทำงานของการควอนไตซ์แบบเวกไตซ์ แบ่งเป็น 2 ขั้นตอนดังนี้

2.1.1.2.1 การสร้างโค้ดบุค (codebook) โดยใช้วิธี K-means

จากขั้นตอนการประมาณเชิงเส้นของเสียงตัวอย่างจำนวนมากจะได้เทรนนิ่งเซตซึ่งประกอบด้วยเวกเตอร์สเปคตรัมจำนวน L เฟรม ; $\{ x_I , 1 \leq i \leq L\}$ เฟรมละ P มิติ ; $x = [x_1, x_2,..., x_p]$ แล้วนำข้อมูลที่ได้มาทำการสร้างเป็นกลุ่มของแบบอ้างอิง

ในระบบการรับรู้เสียงพูดแบบต่างบุคคลจะใช้แบบอ้างอิงของคำหนึ่ง ๆ จากผู้พูดจำนวนมากเพื่อที่จะได้ครอบคลุมถึงความแปรปรวนต่าง ๆ ที่เกิดขึ้นระหว่างผู้พูดแต่ละคน เนื่องจากถ้าใช้แบบอ้างอิงจำนวนมาก เวลาที่ใช้ในการตอบสนองจะมาก เนื้อที่ในหน่วยความนำสำรองที่ใช้เก็บแบบอ้างอิงจะเพิ่มและเมื่อเพิ่มแบบอ้างอิงไปจนถึงระดับหนึ่ง ความถูกต้องในการรับรู้จะเริ่มคงที่ ดังนั้นจึงมีการจัดกลุ่มของแบบอ้างอิงใหม่เพื่อให้ได้แบบอ้างอิงที่เหมาะสม และสามารถใช้เป็นตัวแทนของแบบอ้างอิงที่มีอยู่ทั้งหมดได้ อัลกอริทึมที่ใช้ได้แก่ K-mean Algorithm ซึ่งขั้นตอนที่ใช้ในการสร้างโค้ดบุกมีดังนี้

- นำเทรนนิ่งเซตมาใช้ในการสร้างโค้ดบุค

ขนาดโค้ดบุคของการควอนไตซ์แบบเวกเตอร์ คือ $M\ 2^B$ เวกเตอร์ (B – bit codebook) และเพื่อที่จะหาเซตของ M โค้ดบุกที่ดีที่สุด จำนวนเวกเตอร์อินพุทจะต้องมากกว่าขนาดโค้ดบุคมาก (L>>M)

- การสุ่มค่าเริ่มต้น

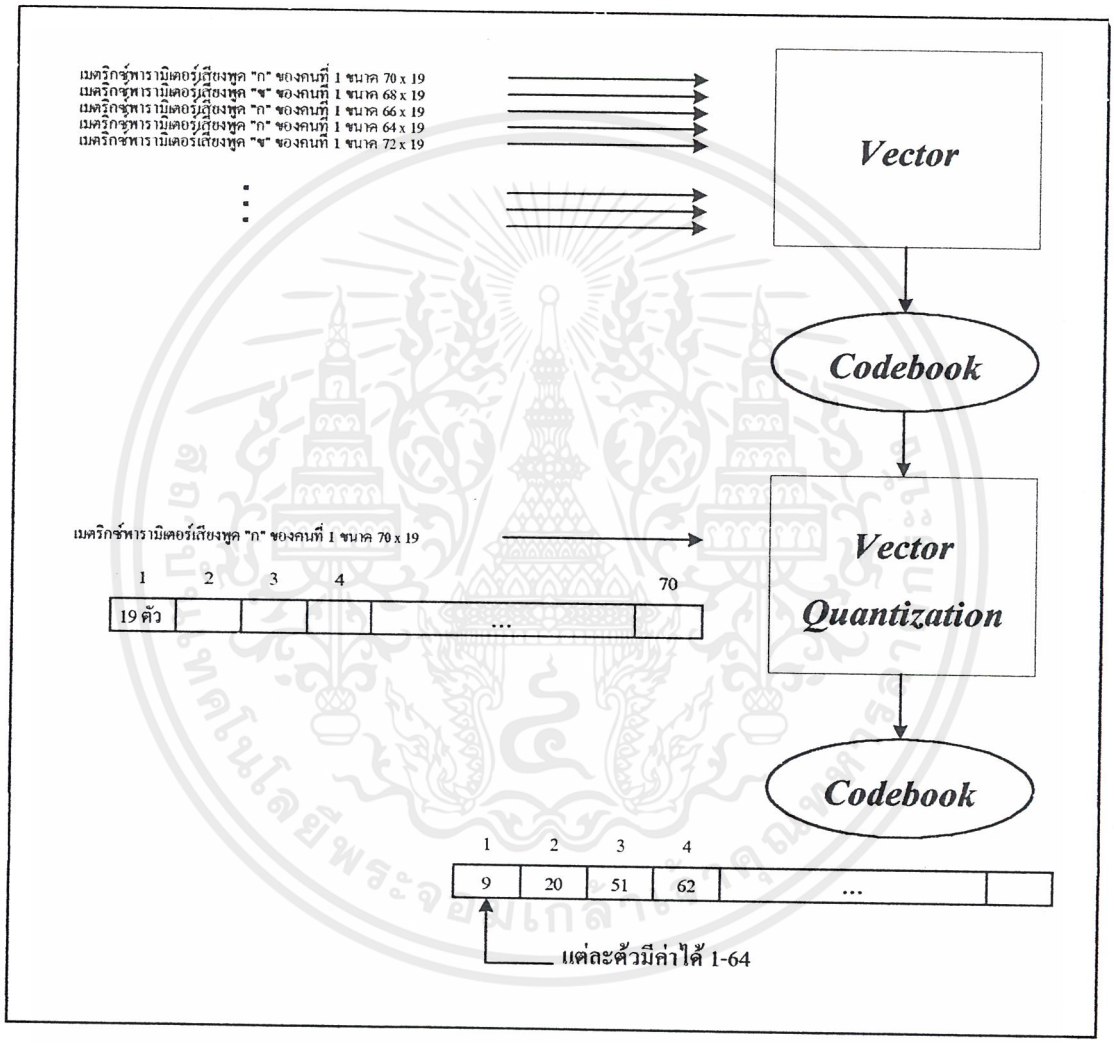
การสุ่มค่าเริ่มต้น เป็นวิธีหนึ่งในการออกแบบโค้ดบุคซึ่งก็คือ การเลือกค่าเริ่มต้นของโค้ดบุคเรียกโค้ดบุคที่ได้จากการสุ่มค่าเริ่มต้อนนี้ว่า แรนดอมโค้ดบุค (random codebook) ถึงแม้วิธีนี้จะไม่ใช่วิธีที่ดีนัก แต่โค้ดบุกที่ได้จากการสุ่มก็ให้ผลเป็นที่ยอมรับ

- การหาค่าความคลาดเคลื่อน

การวัดความคลาดเคลื่อน เป็นส่วนที่จำเป็นและเป็นประโยชน์ต่อการออกแบบโค้ดบุค สมการทางพีชคณิตที่ใช้ในการหาระยะทางมีหลายวิธี แต่วิธีที่นำมาใช้คือ การหาความคลาดเคลื่อนกำลังสองรวม (Total square error) ซึ่งเป็นวิธีการคำนวณที่ง่ายและรวดเร็ว

ถ้ามีสัญญาณมี P มิติ สามารถหาระยะห่างระหว่างสัญญาณอินพุท (x) กับเวกเตอร์โค้ด (y) โดยสมการ

$$d(v_1, v_2) = \|v_1 - v_2\|^2 = \sum_{i=0}^{k-1} (x_i - y_i)^2 \qquad (2.18)$$



รูปที่ 2.15 แสดงขั้นตอนของเวกเตอร์ควอนไตเซชั่น

- การจัดกลุ่ม (classification) และการหาจุดศูนย์กลางของกลุ่ม (center cluster)

การจัดกลุ่มเป็นการแบ่งเวกเตอร์อินพุทเข้าไปตามกลุ่มต่าง ๆ ของแรนดอมโค้ดบุค โดยพิจารณาระยะทาง หรือความคลาดเคลื่อนน้อยที่สุดของแต่ละเวกเตอร์อินพุท x กับเวกเตอร์โค้ดบุก y ซึ่งเป็นโค้ดบุค จากนั้นจะทำการหาค่าเฉลี่ยของแต่ละกลุ่ม เพื่อเป็นค่ากลางของกลุ่มนั้น ๆ จะได้

$$\overline{Y} = \frac{1}{L} \sum_{i=1}^{k} x_i \qquad (2.19)$$

Y เป็นจุดศูนย์กลางซึ่งเป็นเวกเตอร์ที่อยู่ตรงกลางของ $\{x_I\}^L_{I=1}$ ซึ่งแต่ละมิติจะไม่ขึ้นแก่กันหมายความว่า แต่ละ y_k เป็นค่ากลางของ $\{x_I\}^L_{I=1}$

ทำ 2 ขั้นตอนนี้ซ้ำจนกว่าจะเกิดการลู่เข้า (convergent) โดยความคลาดเคลื่อนรวมจะต่ำกว่าค่าหนึ่ง ๆ ซึ่งค่าความคลาดเคลื่อนรวมจะลดลงทุกครั้งที่มีการคำนวณซ้ำใหม่ จึงขึ้นกับค่าที่กำหนดว่าต้องการให้ความคลาดเคลื่อนรวมน้อยเท่าใด ค่ากลางดังกล่าวของแต่ละกลุ่มจะถูกเก็บเป็นเวกเตอร์โค้ดจะได้ว่า y เป็นควอนไตซ์ของค่า x

$$y = q(x) \tag{2.20}$$

โดย q(x) เป็นโอเปอร์เรเตอร์ของควอนไตซ์ y ถูกเรียกว่าเอาท์พุทเวกเตอร์ของค่า x โดย y เป็นค่าใดค่าหนึ่งใน $Y = \{ y_I , 1 \leq i \leq M \}$ โดย $y_i = [y_{i1}, y_{i2},..., y_{ip}]$ Y เป็นเซตของโค้ดบุค M เป็นขนาดของโค้ดบุค และ $\{ y_I \}$ เป็นเซตของเวกเตอร์โค้ด y_I อาจเรียกว่าเป็นโค้ดอ้างอิง และ M อาจเรียกว่าจำนวนระดับขั้น จะทำการแบ่งเวกเตอร์ x ไปใน M เซล $\{ C_I , 1 \leq i \leq M \}$ เมื่อ x อยู่ในเซล C_I

$$q(x) = Y_I \quad \text{ถ้า} \quad x \in C_I \tag{2.21}$$

2.1.1.2.2 การเปรียบเทียบ

เวกเตอร์ควอนไตเซชั่นที่ใช้เพื่อการออกแบบการรับรู้เสียงพูดนั้น มีจำนวนควอนไตเซอร์ M ตัว ซึ่งหมายถึง มี M ระดับเสียงเพื่อการรับรู้ แต่ละระดับเสียงพิจารณามาจากเซตของข้อมูลเทรนนิ่ง ซึ่ง M เป็นดัชนีระดับ แต่ละเซตของเทรนนิ่งในแต่ละระดับจะเก็บเสียงที่อยู่ในระดับเดียวกัน

เมื่อมีเสียงที่เราไม่ทราบ (unknown) ; x_t เข้ามา จะเป็นอินพุทเข้าไปยังทุก ๆ ควอนไตเซอร์ ค่าดัชนีระดับ (index) ที่ถูกเลือกจะเป็นระดับที่มีความคลาดเคลื่อนเฉลี่ยน้อยที่สุดของ D(c) เมื่อ I= 1,2,...,M ซึ่งความคลาดเคลื่อนเฉลี่ยนั้นหาได้จากการวัดระยะทาง โดยใช้วิธีการหาความคลาดเคลื่อนกำลังสองรวม

2.1.2 Pattern Matching

วิธีในการสร้างและหารูปแบบของคำพูดที่เหมือนมีอยู่ 2 แนวทาง

- Dynamic time warpimg (DTW)
- Hidden Markov Models (HMM)

ในที่นี้จะเลือกใช้แบบ HMM เนื่องจากเหมาะกับการวิเคราะห์คำพูดทั้งแบบต่อนเนื่องและไม่ต่อเนื่องได้ดี

Hidden Markov Models

แบบจำลองมาร์คอฟ เป็นแบบจำลอง (model) ทางสถิติซึ่งพัฒนาเพื่อแบ่งกลุ่มของอนุกรมทางเวลา หรือสัญญาณที่ไม่คงที่ นั่นคือ ใช้สำหรับจัดกลุ่มของสัญญาณที่ไม่รู้จัก (Unknown signal) ให้ไปอยู่ในกลุ่มใดกลุ่มหนึ่งของสัญญาณ ซึ่งแบบจำลองมาร์คอฟ ได้ถูกนำมาประยุกต์ใช้ในการรู้จำเสียงพูด

แบบจำลองมาร์คอฟ แบ่งออกเป็น 2 ประเภทคือ แบบต่อเนื่อง (Continuous) และแบบไม่ต่อเนื่อง (Discrete–time) ในที่นี้จะเลือกใช้แบบไม่ต่อเนื่องเพราะเป็นวิธีการที่ซับซ้อนน้อยกว่าและใช้ได้ดีกับคำพูดสั้น ๆ

2.1.2.1 ส่วนประกอบของแบบจำลอง HMM

-N คือจำนวนสเตทในแบบจำลอง โดยสามารถย้ายจากสเตทหนึ่งไปยังอีกสเตทหนึ่งได้ เราให้เซตของสเตทเป็น {1,2…,N} และสเตทที่เวลา t ใด ๆ เป็นq_t

-M คือจำนวนของค่าปรากฏต่อหนึ่งสเตทแทนด้วยสัญลักษณ์ $V=\{V_1,V_2,\ldots,V_M\}$

-A = $\{a_{ij}\}$ คือความน่าจะเป็นในการเปลี่ยนสเตทที่ $a_{ij} = P\{q_t=j \;\; q_{t-1}=i\}$ เมื่อ $1 \leq i,j \leq N$

-B = $\{b_j(k)\}$ คือความน่าจะเป็นของการเกิดค่าปรากฏ ที่ $b_j(k) = P\{O_t=V_k \;\; q_t=j\}$ เมื่อ j=1,2,…,N

-π_i คือความน่าจะเป็นที่แต่ละสเตทจะเป็นสเตทเริ่มต้น เมื่อ $\pi_i = P(q_i$ ที่เวลา t = 1)

2.1.2.2 โครงสร้างของแบบจำลอง HMM

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad (2.22)$$

- แบบ Egordic Model หรือ Fully Connected Model แบบจำลองนี้ทุกสเตทสามารถเปลี่ยนไปยังสเตทอื่น ๆ ได้ทุก ๆ สเตท

- แบบ Left – Right Model หรือ Bakis Model แบบจำลองนี้ การเปลี่ยนสเตทจะเปลี่ยนจากซ้ายไปขวา มีคุณสมบัติการเปลี่ยนสเตทดังนี้

$a_{ij} = 0, j < 1$ หมายความว่า เมื่อผ่านสเตทใดไปแล้วจะไม่มีการย้อนกลับไปยังสเตทนั้นอีก

$\pi_i = \{ 0$ เมื่อ $i \neq 1 ; 1$ เมื่อ $i=1\}$ หมายความว่า ลำดับของสเตทต้องเริ่มต้นที่สเตทที่ 1 สเตทที่เหลือจึงมีความน่าจะเป็นที่จะเป็นสเตทเริ่มต้นเท่ากับศูนย์

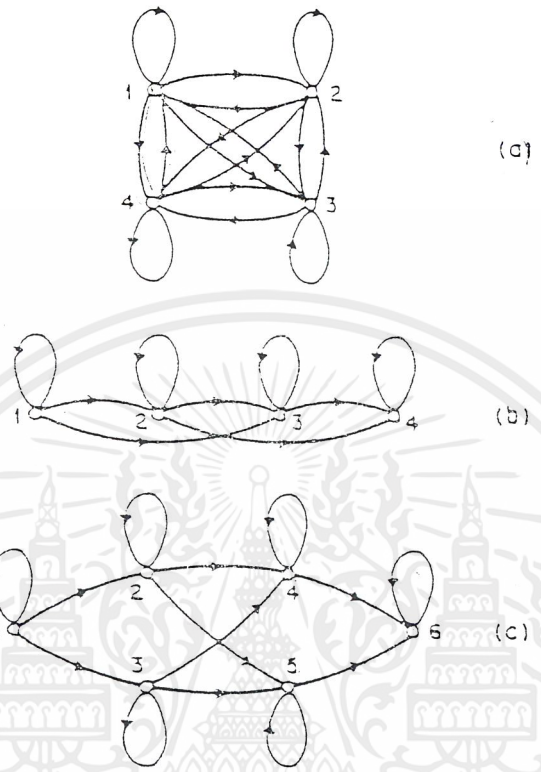
Left – Right Model นี้มีกฎข้อบังคับการเปลี่ยนสเตทดังนี้

$a_{ij} = 0$ เมื่อ $i > i+\Delta i$ โดยค่าของ $\Delta i = 2$ หมายความว่าการเปลี่ยนสเตทจะสามารถเปลี่ยนได้เกิน 2 สเตท จะได้เมตริกซ์การเปลี่ยนสเตทเป็น

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad (2.23)$$

จะเห็นได้ว่าสเตทสุดท้ายมีสัมประสิทธิ์การเปลี่ยนสเตทเป็น $a_{NN} = 1$, $a_{Ni} = 0$ เมื่อ $i < N$ แบบจำลองนี้จึงเหมาะกับสัญญาณที่มีการเปลี่ยนแปลงอย่างต่อเนื่องเช่น คำพูด

- แบบ Parallel Left-Right Model มีคุณสมบัติการเปลี่ยนสเตทคล้ายแบบที่ 2 แต่มีความยืดหยุ่นมากกว่า



รูปที่ 2.16 แสดงแบบจำลองต่าง ๆ ของ HMM

ปัญหาของ HMM

ปัญหาของ HMM มี 3 ข้อซึ่งต้องใช้อัลกอริทึมวิธีต่าง ๆ ในการคำนวณเพื่อแก้ปัญหานี้

ปัญหาที่ 1 เมื่อลำดับของค่าปรากฏ $O = \{O_1, O_2, \ldots, O_T\}$ และมีโมเดล $\lambda = (A,B,\pi)$ เราจะคำนวณหาค่า $P(O/\lambda)$ ของลำดับของค่าปรากฏได้อย่างไร

ปัญหาที่ 2 เมื่อมีลำดับของค่าปรากฏ $O = \{O_1, O_2, \ldots, O_T\}$ และแบบจำลอง $\lambda = (A,B,\pi)$ เราจะหาลำดับสเตท $q = q_1 q_2 \ldots q_T\}$ ที่เหมาะสมในการให้ค่าปรากฏนั้นได้อย่างไร

ปัญหาที่ 3 จะหาแบบจำลอง $\lambda = (A,B,\pi)$ ที่ให้ค่า $P(O/\lambda)$ มากที่สุดได้อย่างไร

ลำดับของค่าปรากฏที่ใช้ปรับค่าพารามิเตอร์ A,B และ π เพื่อให้ได้แบบจำลองที่ดีที่สุดนั้นเรียกว่าลำดับเทรนนิ่ง (training sequence)

2.1.2.3 การคำนวณเพื่อแก้ปัญหาของ HMM

การแก้ปัญหาที่ 1 เป็นการคำนวณว่าแบบจำลอง λ ให้ความน่าจะเป็นที่จะได้ลำดับค่าปรากฏมากน้อยเพียงใด มีวิธีการเพื่อช่วยแก้ปัญหาโดยใช้กระบวนการต่อไปนี้

1. กระบวนการไปข้างหน้า (Forward Procedure)

เมื่อกำหนดให้ตัวแปรไปข้างหน้า (forward variable) $\alpha_t(i)=P(O_1, O_2,..., O_T, q_t=i \quad)$หมายถึงความน่าจะเป็นของการเกิดลำดับค่าปรากฏ $O_1, O_2,..., O_T$ ที่จะอยู่ที่สเตท i ณ เวลา t โดยมีแบบจำลองเป็น λ โดยสามารถหา $\alpha_t(i)$ ได้ดังนี้

- <u>การเริ่มต้น (initialization)</u> เมื่อกำหนด $\alpha_t(i) = \pi_i b_i(O_1)$ ที่เวลาเริ่มต้น t=1 และเหตุการณ์เริ่มต้น O_1 เมื่อ $1 \le i \le N$

- <u>การเหนี่ยวนำ (induction)</u>

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(o_{t+1}) \qquad (2.25)$$

เมื่อ $1 \le t \le T-1$ และ $1 \le j \le N$ หมายถึง ความน่าจะเป็นของสเตท j ที่เวลา t+1 ได้มาจากสเตท i ที่เป็นไปได้ถึง N สเตท ที่เวลา t ดังรูปที่ 3.14

- <u>การสิ้นสุด (termination)</u>

$$P(o/\lambda) = \left[\sum_{i=1}^{N} \alpha_t(i) \right] \qquad (2.24)$$

ความน่าจะเป็นของลำดับค่าปรากฏ O ได้จากผลรวมของ $\alpha_t(i)$ จากทุก ๆ สเตท เมื่อ $1 \le j \le N$

2. กระบวนการย้อนกลับ (Backward Procedure)

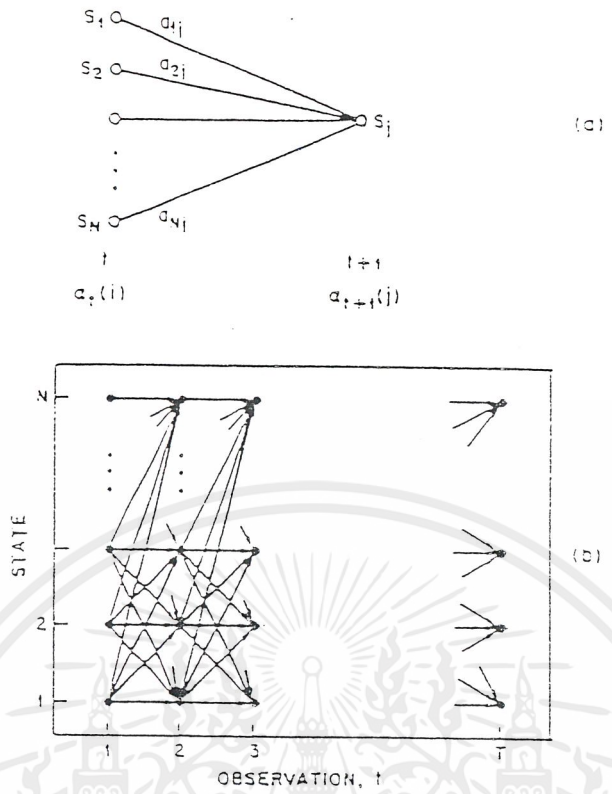
เมื่อกำหนดให้ตัวแปรย้อนกลับ (Backward Procedure) $\beta_T(i) = P(O_1, O_2,..., O_T, q_t=i \quad \lambda)$ หมายถึงความน่าจะเป็นของลำดับค่าปรากฏส่วนหลัง จากเวลา t + 1 ไปจนจบ โดยกำหนดว่าต้องอยู่ที่สเตท I ที่เวลา t และมีแบบจำลองเป็น λ เราจะคำนวณหา $\beta_T(i)$ ได้ดังนี้

- <u>การเริ่มต้น (initialization)</u>

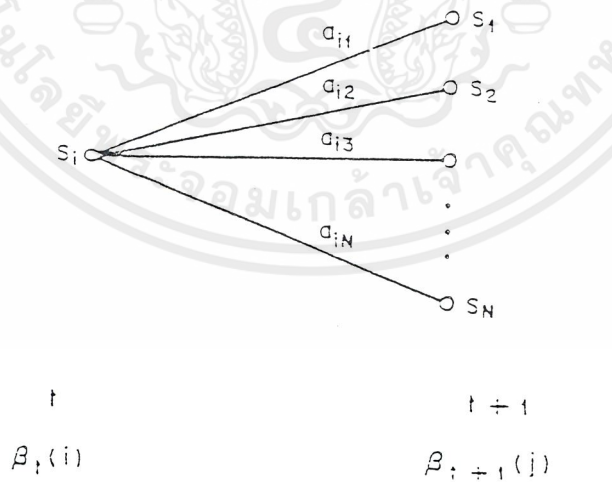
$$\beta_T(i) = 1 \text{ เมื่อ } 1 \le j \le N \qquad (2.26)$$

- <u>การเหนี่ยวนำ (induction)</u>

$$\beta_T(i) = \sum_{i=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(i) \qquad \text{เมื่อ } t = T-1, t-2,...,1,\ 1 \le i \le N \quad (2.27)$$



รูปที่ 2.17 แสดงกระบวนการไปข้างหน้า



รูปที่ 2.18 แสดงกระบวนการถอยหลัง

การแก้ปัญหาที่ 2 เพื่อหาลำดับสเตทที่เหมาะสม

เราจะใช้วิธีวิทเทอร์ บี อัลกอริทึม (Viterbi Algorithm) เพื่อหาลำดับสเตทที่ดีที่สุด ณ เวลา t หนึ่ง เมื่อกำหนดลำดับเหตุการณ์ $O = (O_1, O_2, \ldots, O_t)$ โดยนิยามให้

$$\delta_t(i) = \max_{q1,q2,\ldots,qt} P\left[q_1 q_2 \ldots q_{t-1}, q_t = i, o_1 o_2 \ldots o_t \middle| \lambda\right] \qquad (2.28)$$

หมายถึง ความน่าจะเป็นสูงสุดของเส้นทาง (path) ณ เวลา t ซึ่งเริ่มนับจากเหตุการณ์ที่เวลาเริ่มต้นจนถึงเวลา t ที่สเตท i และโดยการอาศัยคุณสมบัติการเหนี่ยวนำ (induction) เราจะได้

$$\delta_{t-1}(i) = \left[\max_t \delta_t(i) a_{ij}\right] b_j(o_{t+1}) \qquad (2.29)$$

เราสามารถหาลำดับสเตทที่ดีที่สุดได้โดยใช้กระบวนการต่อไปนี้ เมื่อกำหนดให้ $\psi_t(i)$ เป็น อาเรย์ (array)

- <u>การเริ่มต้น (initialization)</u>

$$\delta_t(i) = \pi_i b_i(o_1) \qquad \text{เมื่อ } 1 \le j \le N \qquad (2.30)$$

$$\psi_t(i) = 0 \qquad (2.31)$$

- <u>การย้อนกลับ (recursion)</u>

$$\delta_t(j) = \pi_i b_i(o_1) \qquad \text{เมื่อ } 2 \le t \le T,\ 1 \le j \le N \qquad (2.32)$$

$$\psi_i(j) = \arg \max_{1 \le i \le N} \left[\delta_{t-1} a_{ij}\right] \qquad \text{เมื่อ } 2 \le t \le T,\ 1 \le j \le N \qquad (2.33)$$

- <u>การสิ้นสุด (termination)</u>

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \qquad (2.34)$$

$$q_T^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \qquad (2.35)$$

- <u>เส้นทางเดินย้อนกลับ (Path backtracking)</u>

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \qquad \text{เมื่อ } t = T-1, T-2, \ldots, 1 \qquad (2.36)$$

<u>การแก้ปัญหาที่ 3</u> เพื่อหาโมเดลที่จะให้ผลตามลำดับค่าปรากฏหนึ่ง ๆ โดยเลือกค่าพารามิเตอร์ A,B,π ที่ดีที่สุด โดยใช้กระบวนการทำซ้ำ (Iterative) วิธีที่เราเลือกใช้ คือวิธี บาม – เวลช์ (Baum – Welch) หรือ EM (Expectation – Maximization)

เมื่อนิยามให้

1. $\gamma_t(i) = P(q_t=i \mid O,\lambda)$

หมายถึง ความน่าจะเป็นที่จะอยู่ที่สเตท i ณ เวลา t โดยกำหนดลำดับเหตุการณ์ O และแบบจำลอง λ ให้ สามารถแสดงค่า $\gamma_t(i)$ ได้ดังนี้

$$\gamma_t(i) = \frac{p(O, q_t = i \setminus \lambda)}{p(O \setminus \lambda)}$$

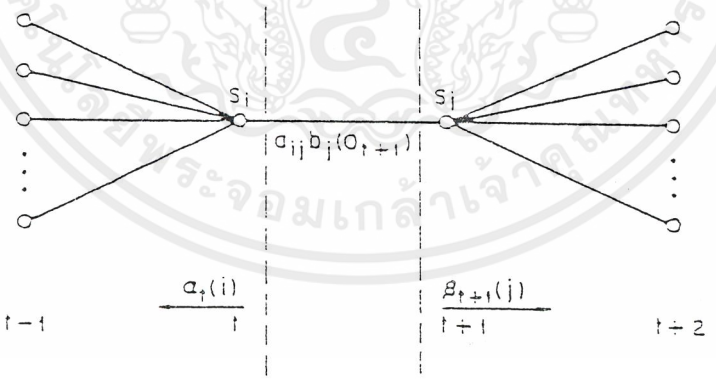
$$= \frac{p(O, q_t = i \setminus \lambda)}{\sum_{i=1}^{N} p(O, q_t = i \setminus \lambda)} \qquad (2.37)$$

เนื่องจาก $P(O,q_t=i \mid \lambda)$ มีค่าเท่ากับ $\alpha_t(i)\beta(i)$ จึงได้

$$\gamma_t(i) = \frac{\alpha(i)\beta(i)}{\sum_{i=1}^{N} \alpha(i)\beta(i)} \qquad (2.38)$$

2. $\varepsilon_t(i,j) = P(q_t=i, q_{t+1}=j \mid O,\lambda)$

หมายถึง ความน่าจะเป็นที่จะอยู่ที่สเตท i ที่เวลา t และสเตท j ที่เวลา t+1 เมื่อกำหนดแบบจำลองและลำดับค่าปรากฏให้



รูปที่ 2.19 แสดงค่าปรากฏที่จะอยู่ที่สเตท I ที่เวลา t โดยคำนึงถึงลำดับค่าปรากฏจากเวลา t+1 ซึ่งต้องพิจารณาสเตท j ที่จะเป็นไปได้ทั้งหมด ณ เวลา t+1 โดยจะขึ้นอยู่กับค่า a_{ij} และ $b_j(o_{t+1})$

ซึ่งจากนิยามของตัวแปรไปข้างหน้าและตัวแปรย้อนกลับ สามารถนำมาสัมพันธ์กับ $\varepsilon_t(i,j)$ ได้ดังนี้

$$\varepsilon(i,j) = \frac{p(q_t = i, q_{t+1} = j, O \setminus \lambda)}{p(O \setminus \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_{ij}(o_{t+1}) \beta_{t+1}(j)}{p(O \setminus \lambda)}$$

$$= \frac{\alpha_t(i) a_{ij} b_{ij}(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_{ij}(o_{t+1}) \beta_{t+1}(j)} \quad (2.39)$$

และจะได้ความสัมพันธ์ของ $\gamma_t(i)$ กับ $\varepsilon_t(i,j)$ ดังนี้

$$\gamma_t(i) = \sum_{j=1}^{N} \varepsilon_t(i,j) \quad (2.40)$$

และ

$\sum_{i=1}^{T-1} \varepsilon_t(i)$ = จำนวนของการเปลี่ยนสเตทออกจากสเตท I ลำดับค่าปรากฏ O

$\sum_{i=1}^{T-1} \gamma_t(i)$ =จำนวนของการเปลี่ยนสเตทจากสเตท i ไป j ในลำดับค่าปรากฏ O

ดังนั้นสามารถหาค่าพารามิเตอร์ได้ดังนี้

$$\pi_i' = \gamma_t(i) \text{ เมื่อ } 1 \le i \le N \quad (2.41)$$

$$a_{ij}' = \frac{\sum_{t=1, o_t = v_k} \wp(j)}{\sum_{t=1}^{T} \wp(i)} \quad (2.42)$$

$$b_i(k) = \frac{\sum_{t=1, o_t = v_k} \wp(j)}{\sum_{t=1}^{T} \wp(j)} \quad (2.43)$$

จากกระบวนการข้างต้น ถ้าเราจะคำนวณซ้ำ ๆ โดยให้ $\lambda' = (A', B', \pi')$ แทน $\lambda = (A, B, \pi)$ ซึ่งเป็นแบบจำลองเริ่มต้นแล้ว จะทำให้ความน่าจะเป็นของการเกิดลำดับค่าปรากฏ O ดีขึ้น จนกระทั่งถึงจุดวิกฤตจึงหยุด ซึ่งเราจะได้จุดวิกฤตของฟังก์ชันความน่าจะเป็นในกรณีที่ $\lambda' = \lambda$ หรือถ้า λ' มีความน่าจะเป็นมากกว่าแบบจำลอง λ' ในลักษณะที่ $P(O|\lambda') > P(O|\lambda)$ นั่นก็คือ เราก็จะได้แบบจำลอง λ' ใหม่ที่น่าจะทำให้เกิดลำดับค่าปรากฏ 0 ได้ดีกว่า

2.1.2.4 การปรับค่าพารามิเตอร์ของ HMM

1. การสเกลลิ่ง (Scaling) เนื่องจาก $\alpha_t(i)$ จะประกอบด้วยผลรวมของเทอมจำนวนมาก ซึ่งก็คือ

$$\left(\prod_{s=1}^{t-1} a_{q_s q_{s+1}} = \prod_{s=1}^{t} b_{q_s}(O_s)\right) \tag{2.44}$$

และเนื่องจากแต่ละเทอมของ a และ b มีค่าน้อยกว่า 1 อยู่แล้ว เมื่อพิจารณาผลรวมของการคูณค่ายิ่งน้อยลงเรื่อย ๆ แสดงว่าเมื่อ t มากขึ้น แต่ละเทอมของ $\alpha_t(i)$ จะเข้าสู่ศูนย์ ทำให้ Dynamic Range ของการคำนวณ $\alpha_t(i)$ เกิน Range ของคอมพิวเตอร์ ทำให้ค่าที่ได้ไม่ถูกต้อง จึงได้มีการสเกลลิ่งขึ้นเพื่อทำให้ $\alpha_t(i)$ อยู่ภายใน Dynamic Range ของคอมพิวเตอร์ การสเกลลิ่งทำได้โดยการคูณ $\alpha_t(i)$ โดยสัมประสิทธิ์การสเกลลิ่ง ซึ่งสัมประสิทธิ์นี้ไม่ขึ้นกับ i การสเกลลิ่ง $B_t(i)$ ก็เช่นเดียวกัน หลังการคำนวณค่าการสเกลลิ่งก็จะตัดกันหมดไปเอง พิจารณาจากสมการ

$$\overline{a_{ij}} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{T} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{2.45}$$

เมื่อเราให้ $\alpha_t(i)$ แทน α ที่ยังไม่ได้สเกลลิ่ง

$\hat{\alpha}_t(i)$ เป็น α ที่สเกลลิ่งแล้ว

$\hat{\hat{\alpha}}_t(i)$ แทนเวอร์ชันของ α ก่อนการสเกลลิ่ง

เมื่อ $t = 1$ จะได้ $\alpha_t(i) = c_1(i)\,\alpha_1(i)$

เมื่อ $$c_1 = \frac{1}{\sum_{i=1}^{N} \alpha_i(i)} \tag{2.46}$$

เมื่อ $2 \leq t \leq T$ คำนวณ $\alpha_t(i)$ จากสมการ (2.17) ในเทอมของ $\alpha_t(i)$ ค่าก่อน

$$\hat{\hat{\alpha}}_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ij} b_i(O_t) \tag{2.47}$$

เมื่อสัมประสิทธิ์ การสเกลลิ่ง เป็น

$$c_1 = \frac{1}{\sum_{i=1}^{N} \hat{\hat{\alpha}}_i(i)} \tag{2.48}$$

จากสมการ 2.17 จะเขียนได้ว่า

$$\hat{\alpha}_{t-1}(j) = (\prod_{t=1}^{t-1} c_t)\alpha_{t-1}(i) \tag{2.49}$$

และโดยการเหนี่ยวนำ จะได้

เมื่อให้ $\hat{\alpha}_t(i) = c_t \hat{\hat{\alpha}}_t(i)$

$$\alpha_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ij} b_j(O_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ij} b_j(O_t)} \tag{2.50}$$

จะได้ว่า

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j)(\prod_{\tau=1}^{t-1} c_t) a_{ij} b_j(O_t)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_{t-1}(j)(\prod_{\tau=1}^{t-1} c_t) a_{ij} b_j(O_t)} = \frac{\alpha_t(i)}{\sum_{t=1}^{\tau} \alpha_t(j)} \tag{2.51}$$

นั่นคือ จะสเกล $\alpha_t(i)$ ได้โดยหารด้วยผลรวมของ $\alpha_t(i)$ ทั้งหมดและสเกล $B_t(i)$ ด้วยค่าเดียวกันนี้ในเทอมของการสเกลนี้สมการ จะเป็น

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{i=1}^{T-1}\sum_{j=1}^{N} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{2.52}$$

โดยแต่ละ $\alpha_t(i).\beta_{t+1}(j)$ จะได้เป็น

$$\alpha_t(i) = \left[\prod_{s=1}^{T} c_s\right] \alpha_t(i) = c_t \alpha_t(i) \tag{2.53}$$

$$\beta_{t+1}(j) = \left[\prod_{s=1}^{T} c_s\right] \beta_{t+1}(j) = D_{t+1} \beta_{t+1}(j) \tag{2.54}$$

ดังนั้นสมการ จะเขียนได้เป็น

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \hat{\beta}_{t+1}(j)}{\sum_{i=1}^{T-1}\sum_{j=1}^{N} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)} \tag{2.55}$$

ซึ่งเทอม $C_t.D_{t+1}$ จะเขียนได้ในเทอม

$$C_t D_{t+1} = \prod_{s=1}^{t} c_s \prod_{s=t+1}^{T} c_s = \prod_{s=1}^{T} c_s = C_T \tag{2.56}$$

ซึ่งไม่ขึ้นกับเวลา t ดังนั้น $C_t.D_{t+1}$ จะถูกตัดทิ้ง ทั้งเศษและส่วนของสมการ ซึ่งทำให้ได้สูตรการคำนวณซ้ำ ๆ (re-estimate) เดิมกลับคืนมา

กระบวนการสเกลลิ่ง ดังกล่าวนี้สามารถใช้ได้กับสัมประสิทธิ์ β และ π ในการสเกลลิ่งนี้จะทำให้การคำนวณค่า $P(O|\lambda)$ เปลี่ยนไป เราจะไม่สามารถหาได้จากการรวมกับเทอม $\alpha_t(i)$ แต่จะหาจากคุณสมบัติ

$$\prod_{t=1}^{T} c_t \sum_{t=1}^{N} \alpha_T(i) = C_T \sum_{i=1}^{N} \alpha_T(i) = 1 \tag{2.57}$$

ดังนั้นจะได้

$$\prod_{t=1}^{T} c_t P(O \setminus \lambda) = 1 \tag{2.58}$$

$$P(O \setminus \lambda) = \frac{1}{\prod_{t=1}^{T} c_t} \tag{2.59}$$

ทำให้อยู่ในรูป log ของ P เพื่อไม่ให้เกิน Dynamic Range ของคอมพิวเตอร์

$$\log\left[P(O \setminus \lambda)\right] = \sum_{t=1}^{T} c_t \tag{2.60}$$

2. ลำดับของค่าปรากฏหลายเหตุการณ์ (Multiple Observation Sequence)

ในการใช้แบบจำลองแบบ Left-Right นั้น การเทรนแบบจำลองต้องใช้หลาย ๆ เหตุการณ์ของลำดับค่าปรากฏเข้ามาเทรน เพื่อให้ได้ค่าพารามิเตอร์ที่ถูกต้องมากขึ้น

ถ้าให้เซตของ v ลำดับค่าปรากฏเป็น

$$O = [O^{(1)}, O^{(2)}, \ldots, O^{(k)}] \tag{2.61}$$

เมื่อ $O^{(v)} = O_1^{(v)}, O_2^{(v)}, \ldots, O_{Tv}^{(v)}$ เป็นลำดับค่าปรากฏของเหตุการณ์ที่ v โดยให้แต่ละเหตุการณ์ เป็นอิสระต่อกันจะได้

$$P(O \setminus \lambda) = \prod_{v=1}^{V} P = \prod P_v \tag{2.62}$$

นำเอาจำนวนเหตุการณ์ของการเกิดค่าปรากฏแต่ละเหตุการณ์มารวมกันจะได้สูตรหา $a_{ij}, b_j(k)$ เป็น

$$\overline{a}_{ij} = \frac{\sum_{v=1}^{V} \frac{1}{P_v} \sum_{t=1}^{T_v - 1} \alpha_t^v(i) a_i b_j(O_{t+1}^v) \beta_{t+1}^v(j)}{\sum_{v=1}^{T-1} \frac{1}{P_v} \sum_{t=1}^{T_v - 1} \alpha_t^v(i) \beta_t^v(j)} \tag{2.63}$$

ส่วน π ไม่ต้องคำนวณเนื่องจาก $\pi_1 = 1, \pi_i = 0, i \neq 1$
จะได้การสเกลลิ่ง ที่เหมาะสมของสมการ

$$\overline{a}_1 = \frac{\sum_{v=1}^{V} \sum_{t=1}^{T_v - 1} \alpha^v a_i b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)}{\sum_{v=1}^{V} \sum_{t=1}^{T_v - 1} \sum_{j=1}^{N} \alpha_t^v(i) a_i b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)} \tag{2.64}$$

$$\overline{b}_1(k) = \frac{\sum_{v=1}^{V} \sum_{t=1}^{T_v - 1} \alpha^v a_i b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)}{\sum_{v=1}^{V} \sum_{t=1}^{T_v - 1} \sum_{j=1}^{N} \alpha_t^v(i) a_i b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)} \tag{2.65}$$

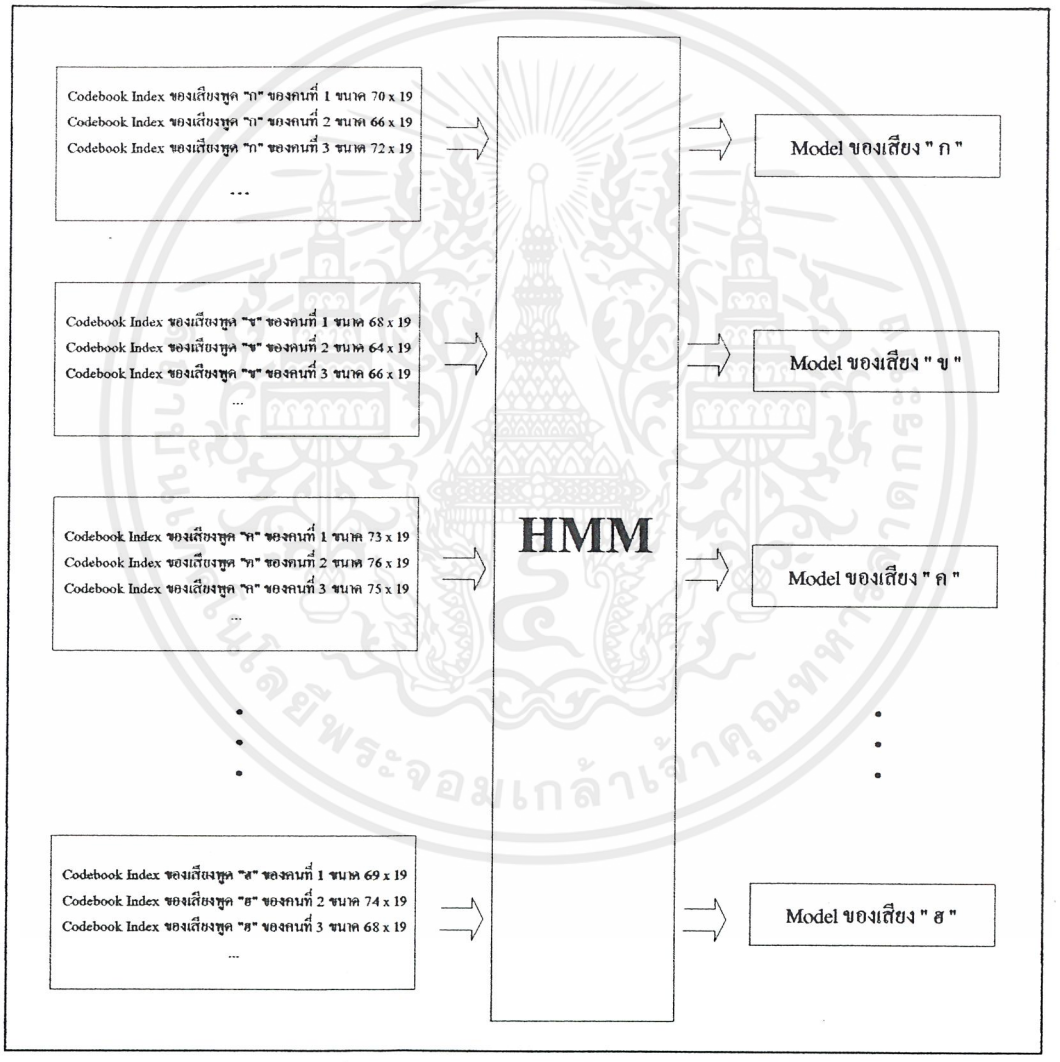
2.1.2.5 ขั้นตอนการสร้างโมเดล

การสร้างโมเดลโดยวิธี HMM นั้นเป็นการนำทฤษฎีความน่าจะเป็นมาอธิบายการเกิดของตัวแปร 2 ตัว คือ **สถานะ** หรือ **state** ซึ่งจะเก็บข้อมูลการเปล่งเสียงส่วนหนึ่งของพยางค์ที่เวลาต่าง ๆ ไว้และ **ค่าปรากฏ** คือ **sequence** ของค่า index ของเสียงที่ได้จากขั้นตอน VQที่เวลาต่าง ๆ ผู้สังเกตจะเห็นเพียงค่าปรากฏของแต่ละ state แต่จะไม่ทราบแน่ชัดว่าอยู่ที่ state ใดจึงเรียกว่า Hiddan Markov Models อาจเปรียบเทียบกับการโดยเหรียญ สมมุติว่าโยนเหรียญ 3 เหรียญ แล้วออก "ก้อย" 1 เหรียญ ในที่นี้เหรียญ 3 เหรียญ

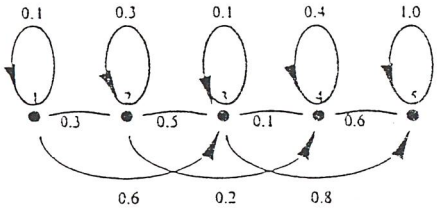
ก็คือ จำนวน state 3 state และค่าปรากฏก็คือ “ก้อย” เราทราบเพียงว่าออก “ก้อย” แต่ไม่ทราบว่าเหรียญใดแน่ที่ออก “ก้อย”

การใช้วิธี HMM สามารถบอกได้ว่าค่าปรากฏนั้นอยู่ที่ state ใด โดยการใช้เทคนิคความน่าจะเป็น
เข้ามาช่วย โมเดลที่สร้างจาก HMMจะประกอบด้วยข้อมูลของสถานะต่าง ๆ ที่เชื่อมเข้าหากัน โดยใช้เส้นที่หมายถึงการเปลี่ยนแปลงสถานะของข้อมูลความน่าจะเป็น เพื่อใช้บอกว่าสถานะใดจะเกิดต่อจากสถานะปัจจุบัน

ชนิดของโมเดลที่ใช้เป็นแบบ Left-Right Model ซึ่งมีลักษณะการย้าย state จากซ้ายไปขวา และที่state ใด ๆ จะสามารถย้ายไป state ถัดไปได้มากสุด 2 state ดังแสดงในรูปที่ 2.21



รูปที่ 2.20 แสดงขั้นตอนการสร้างโมเดล



รูปที่ 2.21 แสดงโมเดลของเสียง ๆ หนึ่ง

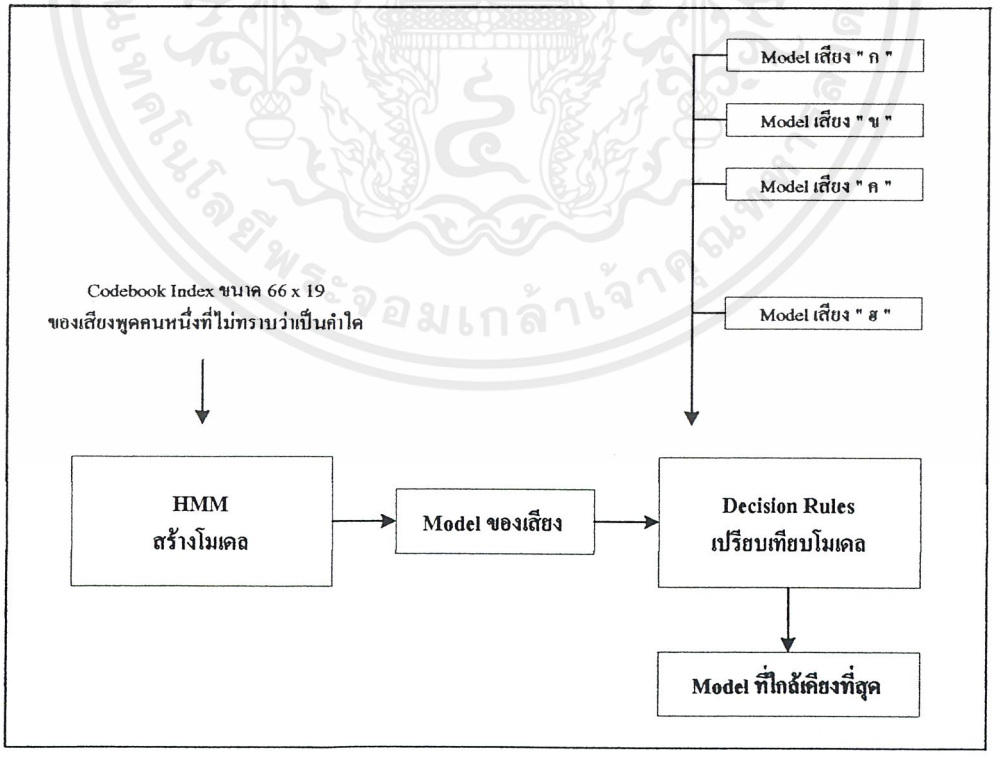
จากรูปตัวอย่างโมเดลของเสียงนี้ จะประกอบด้วยสเตท 5 สเตท แต่ละเส้นของการย้าย state จะกำหนดด้วยค่าความน่าจะเป็นค่าหนึ่ง ซึ่งแต่ละโมเดลจะมีค่าต่างกัน ที่สเตทใด ๆ ค่าความน่าจะเป็นรวมของการย้ายสเตทจะเท่ากับ 1

2.1.3 Decision rule

ดังที่ได้กล่าวไว้ในส่วนการแก้ปัญหาข้อที่ 2 ของ HMM โดยใช้วิธีตาม Viterbi algorithm

Viterbi Algorithm ใช้ในการระยะทางที่สั้นที่สุดจากระยะทางที่เป็นไปได้ ของโมเดลที่มีอยู่กับโมเดลของเสียงที่เข้ามา โดยเลือกโมเดลที่มีความเป็นไปได้ในการเกิดเหตุการณ์ ถ้าโมเดลของคำใดมีค่าความน่าจะเป็นสูงกว่าโมเดลของคำอื่น ๆ จะแสดงว่าคำที่ไม่ทราบว่าเป็นเสียงใด ก็คือคำนั้นนั่นเอง

ขั้นการเปรียบเทียบโมเดลเป็นขั้นที่นำเสียง Input ที่ไม่ทราบว่าเป็นคำใด เข้ามาผ่านกระบวนการสร้างโมเดลเฉพาะของเสียงนั้น แล้วนำโมเดลที่ได้นี้ไปเปรียบเทียบกับโมเดลของเสียงทุกเสียงที่ได้เก็บไว้แล้ว



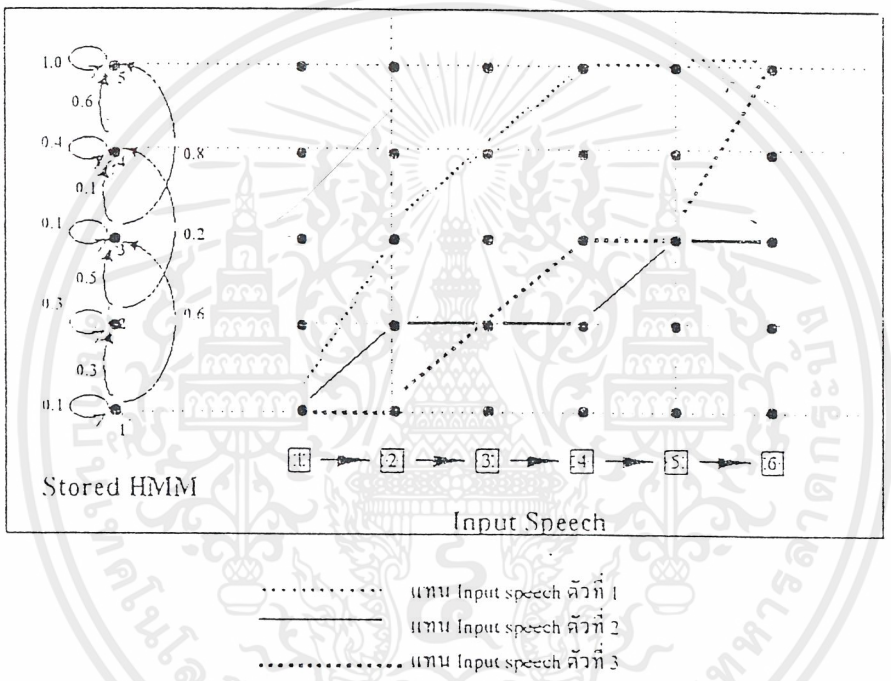
รูปที่ 2.22 แสดงขั้นตอนการตัดสินใจ

การเปรียบเทียบโมเดลนั้นจะใช้วิธี Viterbi Algorithm (VA) ซึ่งจะเลือกเส้นทางที่สั้นที่สุดของโมเดลเสียงที่ไม่ทราบกับโมเดลที่เก็บไว้ทุกตัว ผลของการเลือกจะทำให้ทราบว่าเสียงนั้นเป็นเสียงใด

ความสั้นยาวของเส้นทางที่กล่าวถึงก็คือความน่าจะเป็นของการเปลี่ยนสเตทนั่นเอง ค่าปรากฏที่เกิดจากโมเดลของเสี่ยงที่ไม่ทราบค่าจะเป็นตัวกำหนดสเตทที่เกิดขึ้นในแต่ละช่วง

การตัดสินใจว่าเสียงนั้นตรงกับโมเดลใดได้มาจากการเปรียบเทียบค่าที่น้อยที่สุดของ ความยาวรวมของเส้นทางที่ได้จากสเตททั้งหมด ที่เปรียบเทียบกับแต่ละโมเดล

ตัวอย่างนี้มี Input Speech 3 ตัวเทียบกับแบบจำลอง HMM1 ตัว Input ตัวที่มีระยะทางห่างจากแบบจำลองนี้น้อยที่สุด ก็จะถือว่าเป็นคำนี้



รูปที่ 2.23 แสดงการเปรียบเทียบโมเดล

2.2 หลักการตัดเสียงให้เป็นคำเดี่ยว

หลักการตัดเสียงให้เป็นคำเดี่ยวเพื่อใช้ในการวิเคราะห์เสียง จากหลักการของการวิเคราะห์จำเป็นที่จะต้องเป็นคำเดี่ยวเท่านั้นจึงจะทำให้การจำเสียงได้ดี ดังนั้นเราจึงจำเป็นที่จะต้องแยกเสียงที่เข้ามาให้เป็นคำเดี่ยวเสียก่อน จากหลักการประมวลผลในเชิงเวลานั้นคือ การวิเคราะห์สัญญาณของเสียงพูดโดยตรงจากรูปคลื่นสัญญาณ (wave form) ซึ่งแตกต่างจากการประมวลผลเชิงความถี่ (Frequency-Domain) ตัวอย่างของการประมวลผลของสัญญาณเสียงพูดเชิงเวลา เช่น การหาค่าอัตราการตัดศูนย์เฉลี่ย (Zero-

crossing Rate) การหาค่าพลังงาน (Energy) และค่าออโตคอร์รีเลชั่น (Autocorrelation) โดยการประมวลผลหาค่าเหล่านี้เป็นไปโดยง่ายมีขั้นตอนที่ไม่ซับซ้อน แต่ให้องค์ประกอบสำคัญของสัญญาณเสียง แต่จากการศึกษาในการคำนวณหาคุณลักษณะของเสียงที่ใช้ในการตัดคำแล้ว เราจะเป็นได้ว่าวิธีการใช้การหาค่าพลังงานของเสียง นั่นคือการหาขนาดกำลังสองของสัญญาณเฉลี่ยในช่วงเวลาสั้น ๆ เป็นวิธีการที่สามารถทำการตัดคำได้ดี และยังใช้เวลาในการคำนวณน้อยเมื่อเปรียบเทียบกับการหาค่าอื่น ๆ นั่นจึงทำให้การหาค่าพลังงานสามารถที่จะนำมาใช้งานจริงได้ โดยจะมีหลักการที่ใช้ศึกษาดังนี้

2.2.1 การประมวลผลเสียงพูดโดยขึ้นกับเวลา

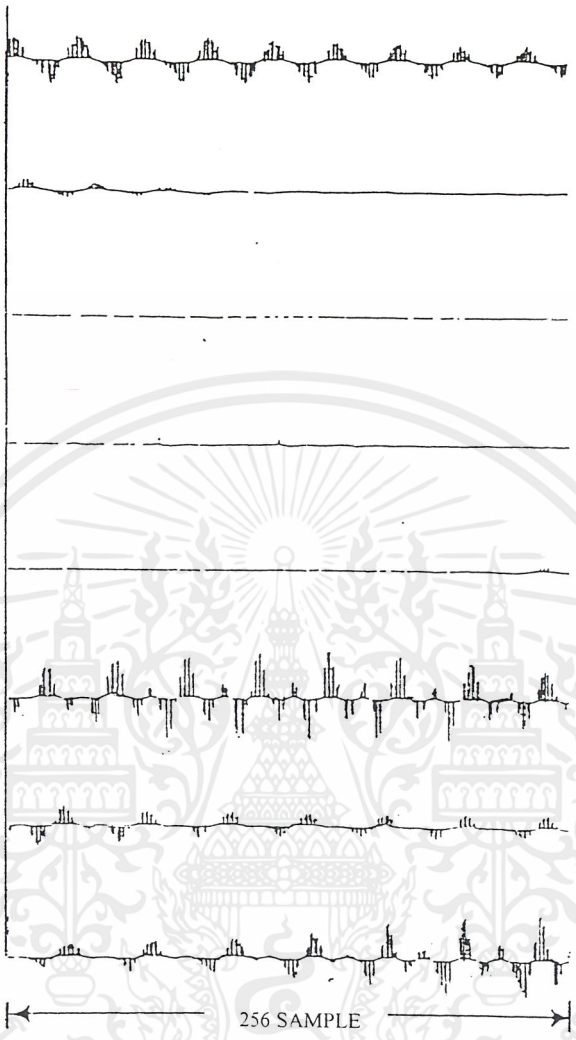
จากรูปที่ 2.33 ซึ่งเป็นสัญญาณเสียงพูดที่มีอัตราการสุ่มตัวอย่างของสัญญาณเป็น 8000 ค่าใน 1 วินาที จะเห็นได้ชัดว่าคุณสมบัติของสัญญาณเสียงพูดจะมีลักษณะที่เปลี่ยนแปลงไปตามเวลา ตัวอย่างเช่น การเปลี่ยนแปลงสัญญาณเสียงพูดระหว่างเสียงที่เป็นเสียงก้องหรือเสียงโฆษะ (voice) และเสียงไม่ก้องหรือเสียงอโฆษะ (unvoice) โดยที่เสียงจะมีการเปลี่ยนแปลงที่เห็นได้ชัดจากขนาด (magnitude) ของสัญญาณ และยังมีการเปลี่ยนแปลงของความถี่มูลฐานของเสียง (fundamental frequency) ภายในช่วงที่เป็นโฆษะด้วย โดยการที่แสดงรูปแบบสัญญาณเป็นรูปคลื่นนี้สามารถทำให้ง่ายต่อการสังเกตถึงองค์ประกอบและคุณลักษณะของเสียงนั้น เช่น ความเข้มของเสียง (intensity) ชนิดของเสียง (เสียงก้องหรือเสียงไม่ก้อง) หรือ excitation mode ความถี่มูลฐาน (pitch หรือ fundamental frequency) และอาจรวมถึงสัมประสิทธิ์ในอวัยวะกำเนิดเสียง เช่น ความถี่กำทอน (resonant หรือ format)

สมมติฐานที่ใช้ในการประมวลสัญญาณเสียงส่วนมากก็คือการที่ถือเอาว่าคุณลักษณะของเสียงนั้นมีการเปลี่ยนแปลงช้ามากเมื่อเทียบกับเวลา และสมมติฐานนี้ทำให้เกิดการประมวลผลที่เรียกว่าการประมวลผลแบบช่วงเวลาสั้น ๆ (short time) โดยที่ส่วนเล็ก ๆ แต่ละส่วนของเสียงพูดนั้นที่ถูกแยกออกมาและจะถูกประมวลผลโดยแต่ละส่วนของเสียงพูดนี้เรียกว่าการวิเคราะห์เฟรม (frame analysis) ซึ่งเฟรมแต่ละเฟรมจะมีส่วนที่ซ้อนทับกัน และผลลัพธ์ที่ได้ของแต่ละส่วนนั้นอาจเป็นตัวเลขเดียว หรืออาจเป็นกลุ่มของตัวเลขก็ได้ ดังนั้นกระบวนการ

กระบวนการวิเคราะห์เสียงแบบช่วงเวลาสั้น ๆ นั้นเกือบทั้งหมดสามารถแสดงในรูปของสมการทางคณิตศาสตร์ที่มีรูปแบบคือ

$$Q_n = \sum_{m=\infty}^{\infty} T[X(m)]W(n-w) \qquad (2.66)$$

สมการนี้สัญญาณจะถูกกรองเอาเฉพาะย่านความถี่ที่ต้องการและจะถูกกระทำโดยฟังก์ชั่น T[ ] ซึ่งอาจเป็นเชิงเส้นหรือไม่ขึ้นอยู่กับสัมประสิทธิ์ที่จะทำการคำนวณ และหลังจากนั้นลำดับที่ได้จะถูกคูณด้วยลำดับวินโดว์ ในตำแหน่งที่สัมพันธ์กับเวลาของตัวอย่างสัญญาณที่ n และผลคูณก็จะถูกรวมกัน โดยทั่วไปแล้วผลลัพธ์จะเกิดจากลำดับที่จำกัด



รูปที่ 2.24 แสดงรูปสัญญาณเสียงที่มีอัตราสุ่มเป็น 8 KHz

2.2.2 ค่าขนาดกำลังสองของสัญญาณเฉลี่ยในช่วงเวลาสั้น ๆ

จากรูปที่ 2.24 เราสามารถสังเกตได้ว่าค่าขนาดของสัญญาณ (Amplitude) นั้นจะมีการเปลี่ยนแปลงไปตามเวลา โดยเฉพาะขนาดของสัญญาณของเสียงไม่ก้อง มักจะมีค่าต่ำกว่าขนาดของสัญญาณเสียงก้องมาก ซึ่งการใช้ค่ากำลังสองของสัญญาณเฉลี่ยในช่วงสั้น ๆ สามารถแสดงให้เห็นถึงความแตกต่างนี้ได้อย่างชัดเจน ได้มากกว่าการใช้ขนาดโดยมีรูปแบบสมการของการใช้ในการคำนวณ คือ

$$E_n = \sum_{m=-\infty}^{\infty} [X(m)W(n-m)]^2 \qquad (2.67)$$

หรือสามารถเขียนได้เป็น

$$E_n = \sum_{m=-\infty}^{\infty} X^2(m)h(n-m) \tag{2.68}$$

โดยที่

$$h(n) = W^2(m) \tag{2.69}$$

สมการนี้สามารถแสดงเป็นบล็อคไดอะแกรมได้ดังรูปที่ 2.25 สัญญาณ $X^2(m)$ จะถูกกรองโดยตัวกรองแบบเชิงเส้น h(n) ก็คือวินโดว์ (window) ลองมาดูว่าวินโดว์ที่แตกต่างกันจะมีผลอย่างไรกับค่าพลังงานผลลัพธ์ที่ได้ โดยถ้า h(n) ให้ค่าขนาดของสัญญาณคงที่ในช่วงเวลาที่กวนานแล้วค่า E จะมีการเปลี่ยนแปลงน้อยมาก เนื่องจากเราต้องการให้ค่ากำลังสองในช่วงเวลาสั้น ๆ แสดงให้เห็นถึงการเปลี่ยนแปลงของขนาดเสียงในสัญญาณเสียง ดังนั้นจึงต้องใช้วินโดว์ที่มีช่วงเวลาสั้นเกินไปก็จะทำให้ไม่ได้ค่าที่มีความเรียบพอเพียง โดยตัวฟังก์ชั่นวินโดว์มีรูปแบบดังนี้

วินโดว์แบบสี่เหลี่ยมผืนผ้า $h(n) = 1$ เมื่อ $0 < n < N-1$

นอกจากนั้น $h(n) = 0$

วินโดว์แบบแฮมมิ่ง

$$h(n) = 0.54 - 0.46\cos\left(\frac{2n\pi}{(N-1)}\right) \quad \text{เมื่อ} \quad 0 \le n \le N-1 \tag{2.70}$$

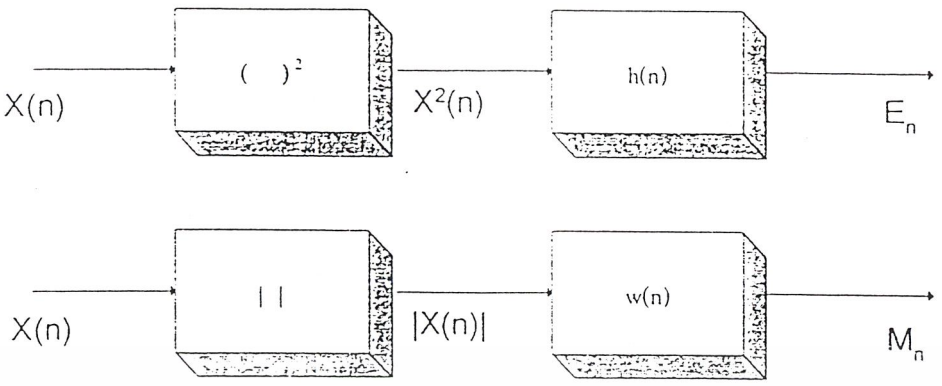
นอกจากนั้น $h(n) = 0$

จากสมการจะเห็นได้ว่าวินโดว์แบบสี่เหลี่ยมนั้นจะมีการถ่วงน้ำหนัก (weight) ค่าเท่ากันตลอดทุกสัญญาณ (sample) ในช่วง (n-N+1) ถึง n ส่วนผลตอบสนองทางความถี่ของวินโดว์แบบสี่เหลี่ยมนั้นจะได้ตามสมการดังนี้

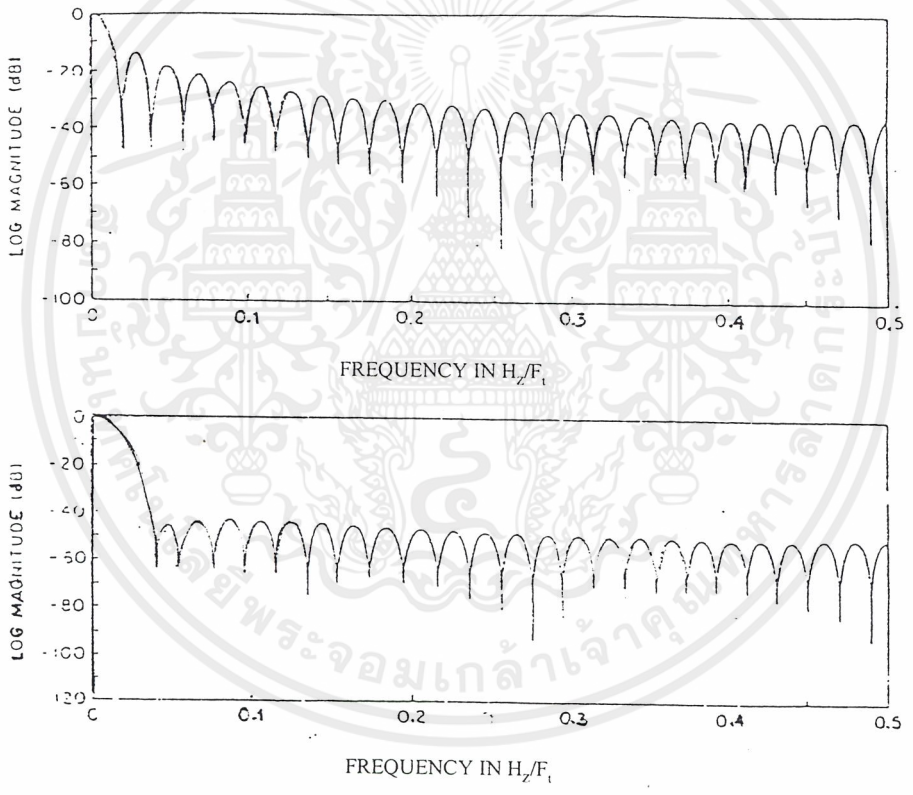
$$H\left(e^{j\Omega T}\right) = \frac{\sin(\Omega NT/2)}{\sin(\Omega T/2)} e^{-j\Omega T(N-1)/2} \tag{2.71}$$

ส่วนแฮมมิ่งวินโดว์นั้นจะใช้ประโยชน์ได้ดีสำหรับการวิเคราะห์เชิงความถี่ เพราะให้ค่าลดทอนภายนอกย่านมากกว่าดังรูปที่ 2.26

จากที่กล่าวมาแล้วว่าถ้าค่า N ในวินโดว์มีค่าน้อยเกินไปก็จะมีการเปลี่ยนแปลงของค่า E_n มากเกินไป แต่ถ้าค่า N มากเกินไปก็จะทำให้สูญเสียคุณสมบัติของเสียงไป แต่เป็นการยากที่จะกำหนดค่า N เพียงค่าเดียวให้เหมาะสมกับเสียงทั้งหมด เนื่องจากคาบเวลาของเสียงที่เรียกว่าพิทช์ (pitch) จะแตกต่างกัน เช่นเด็กหรือผู้หญิงอาจมีค่าเพียง 20 ตัวอย่าง ที่อัตราการสุ่มตัวอย่าง 10 kHz แต่ผู้ชายอาจสูงถึง 250 ตัวอย่าง แต่ค่า N ที่ใช้กันนั้นอยู่ในช่วงประมาณ 100-200 ตัวอย่างที่อัตราการสุ่ม 10 kHz (10-20 มิลลิวินาที)

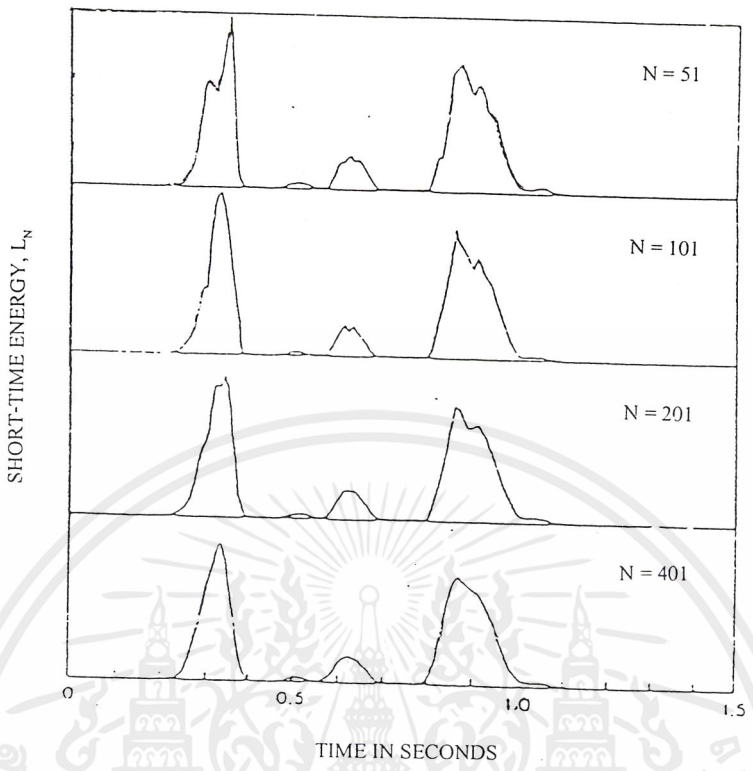


รูปที่ 2.25 แสดงบล็อกไดอะแกรม (a) ขนาดกำลังสองในช่วงเวลาสั้น (b) ขนาดในช่วงเวลาสั้น ๆ



รูปที่ 2.26 แสดงการแปลงฟูเรียร์ของ (a)วินโดว์สี่เหลี่ยม (b) วินโดว์แบบแฮมมิ่ง

จากรูปที่ 2.27 จะแสดงให้เห็นผลจากการเปลี่ยนแปลงค่า N ในวินโดว์ ซึ่งจะสังเกตได้ชัดเจนว่า เมื่อค่า N เพิ่มก็จะทำให้ค่าขนาดกำลังสองในช่วงเวลาสั้น ๆ มีผลออกมามีความราบเรียบมากยิ่งขึ้น



รูปที่ 2.27 แสดงค่าขนาดกำลังสองช่วงเวลาสั้น ๆ สำหรับวินโดว์แบบสี่เหลี่ยมเมื่อเปลี่ยนค่า N

สิ่งสำคัญที่ได้จากค่า E_n นั้นคือการที่เราสามารถแยกแยะส่วนของเสียงก้อง และเสียงไม่ก้องออกจากกันได้โดยส่วนของเสียงที่ไม่ก้องนั้นจะมีค่า E_n ที่ต่ำกว่าเสียงก้องอย่างเห็นได้ชัด ซึ่งจะทำให้สามารถทำการกำหนดจุดอย่างคร่าว ๆ ได้ว่าตำแหน่งไหนเป็นจุดที่เสียงได้ทำการเปลี่ยนจากเสียงก้องหรือในทางกลับกันและสัญญาณที่มีค่า S/N สูง ๆ นั้นค่า E_n จะทำให้เราสามารถแยกเสียงพูดออกจากเสียงเงียบ(silence)ได้ด้วย

2.2.3 การแยกเสียงพูดออกจากเสียงเงียบ (Silence) โดยใช้ค่าขนาดกำลังสองและค่าอัตราการตัดศูนย์เฉลี่ย

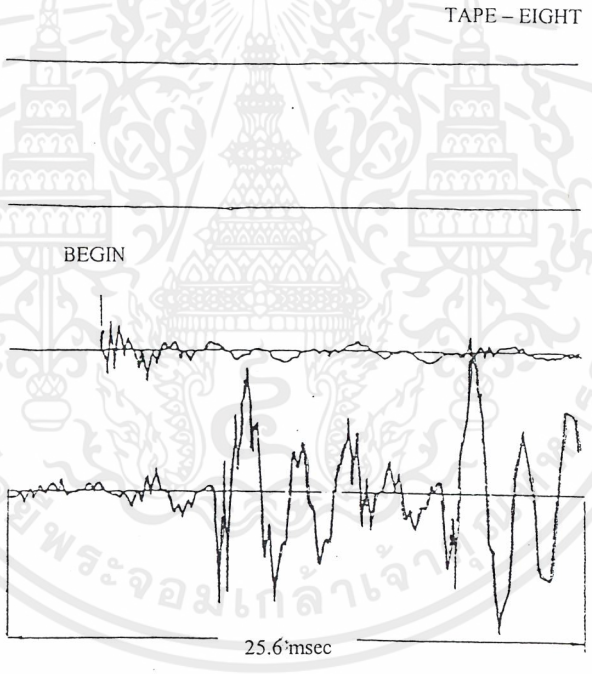
ในระบบการประมวลผลสัญญาณเสียงพูดนั้นสิ่งสำคัญสิ่งหนึ่งคือ การแยกแยะและการหาขอบเขตทั้งจุดเริ่มต้นและจุดสิ้นสุดของเสียงออกจากสัญญาณรบกวน (Background Noise) โดยเฉพาะในกระบวนการรู้จำเสียงพูดชนิดคำโดด (Isolated Word) จำเป็นอย่างยิ่งที่จะต้องกำหนดจุดเริ่มต้นและจุดสิ้นสุดของคำให้ได้อย่างมีประสิทธิภาพ

การใช้ค่าพลังงานและอัตราการตัดศูนย์เฉลี่ย จากในรูปที่ 2.28 เป็นตัวอย่างแรก จะเห็นได้ว่ารูปคลื่นนี้สามารถแยกแยะเสียงออกจากสิ่งแวดล้อมได้ง่าย โดยใช้ค่าขนาดกำลังสองส่วนอีกตัวอย่างหนึ่งในรูปที่ 2.29 เป็นเสียงที่ค่าพลังงานของเสียงใกล้เคียงกับสัญญาณรบกวนมากไม่สามารถแยกแยะออได้ แต่เมื่อมาดูที่ค่าอัตราการศูนย์เฉลี่ยจะมีความแตกต่างกับสัญญาณรบกวนมากพอสมควร

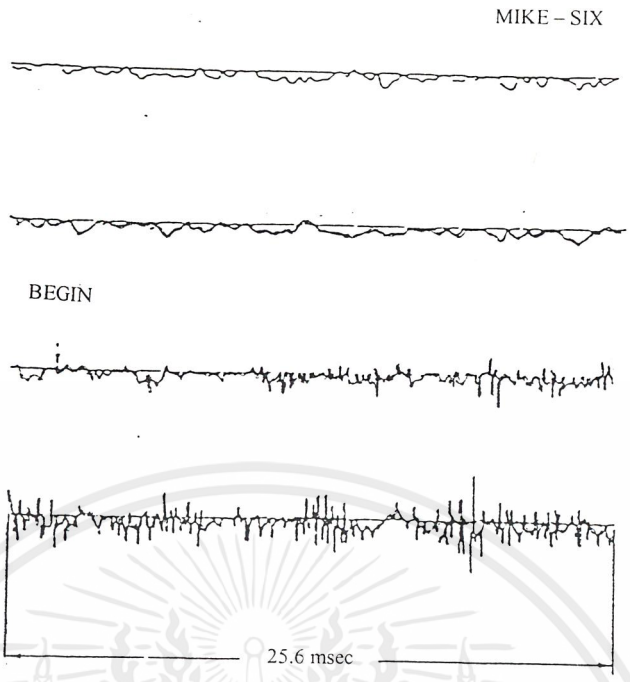
จากรูปที่ 2.30 เป็นตัวอย่างที่ยากมากที่จะทำการกำหนดจุดเริ่มต้นและสิ้นสุดของเสียง ซึ่งจุดเริ่มต้นของเสียงที่มีลักษณะพ่นออกมา (Fricative) ซึ่งจะทำให้ค่าพลังงานมีค่าต่ำมาก ซึ่งลักษณะของเสียงที่ยากต่อการกำหนดขอบเขต มีดังเช่น

1. ลักษณะของเสียงที่พ่นลมออกมา ณ จุดเริ่มต้นหรือสิ้นสุด (Weak fricative)
2. ลักษณะของเสียงที่เป็นการกระเบิดออกมา (Weak plosive busts)
3. ลักษณะของเสียงที่เป็นเสียงนาสิก (Nasals)

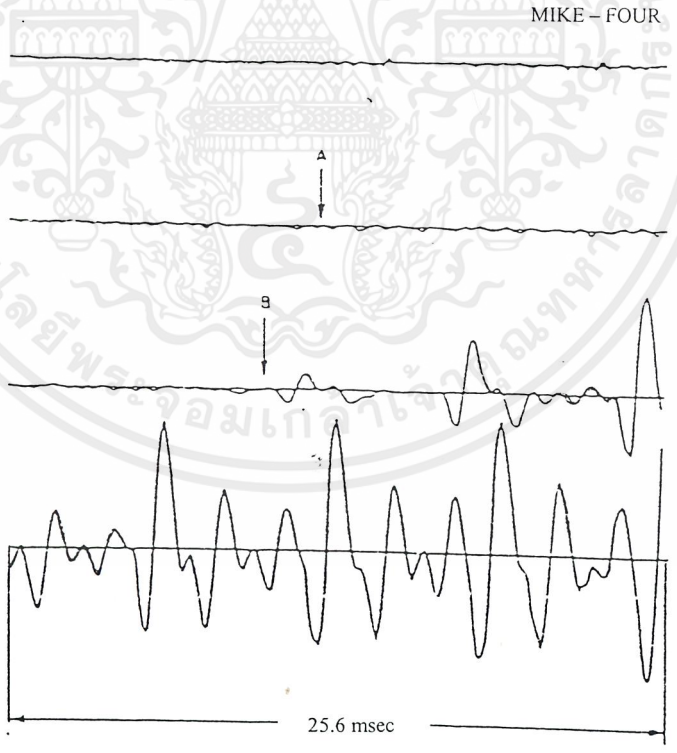
ในการที่จะกำหนดจุดเริ่มต้นและสิ้นสุดของเสียงเหล่านี้ เราสามารถใช้กระบวนการหาค่าขนาดกำลังสอง และอัตราการตัดศูนย์เฉลี่ยร่วมกันได้ ตัวอย่างในระบบรู้จำเสียงพูดแบบคำโดด การบันทึกเสียงจะบันทึกส่วนที่เป็นเสียงพูดและส่วนที่ผู้พูดไม่ได้พูดไว้ด้วย แล้วใช้ระบบหาจุดเริ่มต้นและสิ้นสุดของคำเพื่อที่จะทำให้ผ่านเข้ากระบวนการรู้จำเสียงโดยส่งเฉพาะส่วนที่เป็นเสียงพูดเข้าไปประมวลผลต่อไป



รูปที่ 2.28 แสดงรูปคลื่นของจุดเริ่มต้นเสียงคำว่า /eight/



รูปที่ 2.29 แสดงรูปคลื่นของจุดเริ่มต้นของเสียงคำว่า /six/



รูปที่ 2.30 แสดงรูปคลื่นของจุดเริ่มต้นเสียงคำวา /four/

การทำงานของระบบนี้จะใช้การหาค่าอัตราศูนย์เฉลี่ยที่มีขนาดเฟรมเท่ากับ 10 มิลลิวินาที และการหาค่าพลังงาน และอัตราการตัดศูนย์เฉลี่ยร่วมกัน โดยใช้วินโดว์ขนาด 10 มิลลิวินาที ซึ่งทั้ง 2 กระบวนการจะทำตลอดช่วงของสัญญาณด้วยอัตรา 100 ครั้งต่อวินาที โดยการสมมติให้ช่วงเวลา 100 มิลลิวินาทีแรกไม่มีเสียงพูด หมายความว่าค่าอัตราการตัดศูนย์ และค่าพลังงานในช่วงนี้จะเป็นค่าของสัญญาณรบกวน และทำการคำนวณค่าพลังงานและอัตราการตัดศูนย์เฉลี่ยที่ใช้เป็นเกณฑ์ในการตัดสินใจค่าระดับของพลังงาน

ค่าระดับที่ใช้ส่วนมากจะให้ค่าโดยประมาณซึ่งในความเป็นจริงแล้วจุดเริ่มต้น และจุดสิ้นสุดจะอยู่ภายนอกจากค่าระดับนี้จึงมีการทำซ้ำโดยดูจากจุดที่กำหนดโดยค่าระดับในครั้งแรกและลดลงจนถึงค่าระดับอีกค่าหนึ่งจึงถือได้ว่าเป็นจุดเริ่มต้นและจุดสิ้นสุดของคำ ขอบเขตที่ได้จากการใช้ค่าระดับค่าแรกนั้นเป็นที่น่าเชื่อถือว่าภายในขอบเขตนี้จะไม่มีจุดเริ่มต้นหรือจุดสิ้นสุดของคำอยู่

กระบวนการถัดไปคือ การย้อนกลับจากจุดเริ่มต้นที่ได้จากค่าระดับค่าแรกแล้วทำการเปรียบเทียบค่าอัตราการตัดศูนย์เฉลี่ยกับค่าระดับการตัดศูนย์เฉลี่ยของมันซึ่งได้จากสัญญาณรบกวน โดยถ้าค่าอัตราการตัดศูนย์เฉลี่ยมีค่ามากกว่าค่าระดับเกิน 3 ครั้ง แล้วจะถือว่าจุดเริ่มต้นเป็นจุดที่มีค่าอัตราการตัดศูนย์เฉลี่ยเกินค่าระดับเป็นครั้งแรก

2.3 การเชื่อมต่อส่วนการรู้จำเสียงกับระบบโทรศัพท์

การสื่อสารอาจเกิดขึ้นได้โดยตรงด้วยการเชื่อมต่ออุปกรณ์สองตัวด้วยสายสัญญาณ หรือโดยอ้อมด้วยสื่อกลาง ซึ่งมักจะเป็นโทรศัพท์โดยใช้โมเด็ม (modem) เพื่อแปลงสัญญาณที่ปลายด้านหนึ่งให้เป็นสัญญาณที่เหมาะสมกับการส่งผ่านสายโทรศัพท์

ระบบเครือข่ายโทรศัพท์ที่ใช้กันอยู่ในปัจจุบันนั้นประกอบด้วยสายโทรศัพท์ซึ่งโยงใยไปทั่วและเชื่อมต่อกับอุปกรณ์สวิตชิ่งส่วนกลาง ในขณะที่ผู้ใช้ยกหูโทรศัพท์ก็จะเชื่อมเข้าสู่ระบบสวิตชิ่งส่วนกลางทันที วงจรสวิตชิ่งนี้จะส่งสัญญาณที่เรียกว่า ไดอัลโทน (dial tone) หลังจากที่ผู้ใช้ได้รับสัญญาณนี้ ก็หมายความว่าระบบพร้อมที่จะรับหมายเลขโทรศัพท์ปลายทางแล้ว และทุกครั้งที่กดหมายเลขปลายทางแล้ว ก็จะเชื่อมต่อไปที่หมายเลขปลายทางตามต้องการทันที

มาตรฐานการเชื่อมต่อ RS-232-C มาตรฐาน RS-232-C แบ่งการเชื่อมต่อออกเป็นสองลักษณะ คือ การต่อกับเทอร์มินัล (DTE : Data Terminal Equipment) และการต่ออุปกรณ์สื่อสารข้อมูล (DCE : Data Communication Equipment) ซึ่งในกรณีปกตินั้น DCE จะต้องต่อเข้ากับ DTE เสมอ เช่น การต่อโมเด็มเข้ากับเครื่อง PC จะเป็นอุปกรณ์ DTE และโมเด็มเป็นอุปกรณ์ DCE การเชื่อมต่อตามมาตรฐานนี้โดยปกติจะใช้คอนเน็กเตอร์รูปตัว D ชนิด 25 ขา กำหนดให้ปลายสายสัญญาณด้านหนึ่งเป็นตัวผู้ใช้ต่อกับอุปกรณ์ DCE และปลายสายอีกด้านจะเป็นคอนเน็กเตอร์ตัวเมียใช้ต่อกับอุปกรณ์ DTE แต่อาจมีมาตรฐานแบบใหม่เช่น คอนเน็กเตอร์แบบ 9 ขา สัญญาณที่เกี่ยวข้องกับการสื่อสารอนุกรมมีการแยกประเภทดังนี้

1. สัญญาณข้อมูล

วงจรส่งผ่านข้อมูลมีสองวงจร คือ วงจรที่ทำหน้าที่ส่งข้อมูลจาก DTE ไป DCE และอีกวงจรก็คือวงจรรับข้อมูลจาก DCE เพื่อส่งไป DTE รูปแบบสัญญาณไฟฟ้าที่ใช้แทนข้อมูลเป็นรูปสัญญาณสี่

เหลี่ยมที่ถูกสร้างจากสัญญาณไฟกระแสตรงประมาณ 3 ถึง 25 โวลต์ แทนข้อมูล "0" และ –3 ถึง –25 แทนข้อมูล "1" และช่วงแรงดัน –3 ถึง 3 โวลต์ นั้นจะเป็นช่วงระดับแรงดันที่ใช้ในการแบ่งแยกระดับสัญญาณระหว่างสถานะ "0" และ "1"

2. วงจรควบคุม

มีหน้าที่สร้างสัญญาณควบคุมต่าง ๆ ขึ้นมา เพื่อทำให้เครื่อง PC และโมเด็มทราบสถานะในการทำงานของกันและกัน โดยที่สัญญาณควบคุมจะมีลักษณะทางกายภาพเช่นเดียวกับสัญญาณข้อมูล แต่โมเด็มส่วนใหญ่ไม่ได้ใช้วงจรควบคุมทุกวงจร ซึ่งสัญญาณควบคุมต่าง ๆ มีดังนี้

-Request To Send (RTS) และ Clear To Send (CTS)

เป็นส่วนที่ใช้สำหรับควบคุมการส่งผ่านข้อมูลระหว่างเครื่อง PC และโมเด็ม โดยโมเด็มจะส่งสัญญาณ CTS สถานะ "ON" ให้แก่ PC เพื่อโมเด็มพร้อมที่จะรับข้อมูล และเครื่อง PC ก็จะให้สัญญาณ RTS สถานะ "ON" เมื่อ PC พร้อมที่จะรับข้อมูล

-Data Terminal Ready (DTR) และ Data Set Ready (DSR)

สัญญาณ DTR จะใช้เป็นสิ่งบอกโมเด็มให้ทราบว่าเครื่อง PC นั้นกำลังอยู่ในสถานะที่พร้อมจะติดต่อกับโมเด็มหรือไม่ และในกรณีเดียวกันสัญญาณ DSR นั้นจะใช้เป็นตัวบอกให้เครื่อง PC ทราบว่าโมเด็มพร้อมจะติดต่อหรือไม่ โดยที่สัญญาณ DSR จะอยู่ในสถานะ "ON" ก็ต่อเมื่อโมเด็มได้รับสัญญาณ DTR แล้ว

-Carrier Detect และ Ring Indicator

CD เป็นสัญญาณที่บอก PC ให้ทราบว่าโมเด็มกำลังเชื่อมต่อกับโมเด็มเครื่องอื่น ๆ และกำลังได้รับสัญญาณพาหะจากโมเด็มปลายทาง ส่วนสัญญาณ RI เป็นการบอกเครื่อง PC ให้ทราบว่า มีสัญญาณกริ่งโทรศัพท์เรียกเข้ามาที่โมเด็ม ซึ่งโมเด็มส่วนใหญ่ก็มีวงจรตอบรับโทรศัพท์อัตโนมัติอยู่แล้ว จึงไม่จำเป็นต้องใช้สัญญาณ RI แต่ในบางโปรแกรมก็อาจจะใช้สัญญาณ RI เป็นตัวกำหนดให้เริ่มทำการรันโปรแกรมอื่น ๆ ได้

2.3.1 การเชื่อมต่อโมเด็มเข้ากับคอมพิวเตอร์ โมเด็มภายนอก (External modem) เป็นอุปกรณ์อนุกรมตัวหนึ่ง โมเด็มทั่วไปจะประกอบไปด้วยอุปกรณ์เพิ่มเติมในการใช้งานดังนี้

1. สายสัญญาณและคอนเน็กเตอร์ RS-232-C
2. สายโทรศัพท์และคอนเน็กเตอร์ RJ-11
3. แหล่งจ่ายกำลังไฟฟ้าให้แก่โมเด็ม

โมเด็มจะอยู่ในภาวะใดภาวะหนึ่ง คือภาวะคำสั่ง (Command mode) หรือ ภาวะออนไลน์ (on-line mode) ขณะที่อยู่ในภาวะคำสั่งสามารถสั่งงานโมเด็มได้จากคอมพิวเตอร์ เช่น สั่งให้โมเด็มทำการหมุนหมายเลขโทรศัพท์ เมื่อมีการเชื่อมต่อเกิดขึ้นกับโมเด็มระยะไกล โมเด็มท้องถิ่นจะเข้าสู่สภาวะออนไลน์และจะไม่แปลงความหมายข้อมูลที่ส่งให้กับมัน แต่จะส่งผ่านข้อมูลออกไปแทน ถ้าสัญญาณพาหะหายไป เช่น โมเด็มระยะไกลวางหู โมเด็มจะอยู่ในสภาวะคำสั่ง คำสั่งที่ถูกส่งโดยคอมพิวเตอร์ไปยังโมเด็มสามารถถูกส่งได้โดยซอฟแวร์สื่อสาร หรือพิมพ์เข้าไปจากแป้นพิมพ์ โดยเอาต์พุตของแป้นพิมพ์ถูก

เปลี่ยนทิศทางไปยังพอร์ตอนุกรม คำสั่งที่ถูกส่งในสภาวะคำสั่งควรถูกส่งด้วยเจ็ดบิตข้อมูล หนึ่งพาริตี้ หรือแปดบิตข้อมูล ไม่มีพาริตี้

2.3.2 การทำงานของโมเด็มและชุดคำสั่ง AT

โดยทั่วไปแล้วโมเด็มจะทำงานในโหมดใดโหมดหนึ่ง คือ โหมดคำสั่ง (Command Mode) หรือ โหมดออนไลน์ (Online Mode) เมื่อเปิดเครื่อง โมเด็มจะทำงานในโหมดคำสั่งโดยอัตโนมัติ เมื่อโมเด็มอยู่ในโหมดคำสั่ง ก็สามารถที่จะตั้งค่าคอนฟิเกอเรชันกับระบบคอมพิวเตอร์ หรือ แอปพลิเคชันที่เฉพาะได้ โดยเลือกตัวเลือก และการทำงานที่กำหนดจากเมนูในโปรแกรมซอฟต์แวร์การสื่อสารได้ โปรแกรมนี้จะส่งตัวเลือกผ่านไปยังโมเด็มในรูปของคำสั่ง แล้วโมเด็มจึง ประมวลผลข้อมูลที่ได้รับให้ทำงานตามที่กำหนด อย่างไรก็ตาม ก็สามารถที่จะตั้งคำสั่งจากโหมดเทอร์มินัล (Terminal Mode) ของโปรแกรมการสื่อสารได้โดยตรง ด้วยการใช้ชุดคำสั่ง AT นามสกุลแฟกซ์ระดับ 1 (Class 1 Fax Extension) และ การสนับสนุนรีจิสเตอร์ S (Supporting S Registers) โดยที่สามารถสั่งโมเด็มให้ทำงานตามฟังก์ชั่น หรือชุดฟังก์ชั่นได้ ตัวอย่างเช่น อาจจะสั่งให้โมเด็ม หมุน (ATDn), ตอบ (ATA), และ วางสาย (ATH0) ด้วยคำสั่งที่เหมาะสมได้

แต่จะไม่สามารถที่จะป้อนคำสั่งเมื่อโมเด็มอยู่ในโหมดออนไลน์ ซึ่งก็คือ การส่งหรือรับข้อมูลผ่านสายโทรศัพท์ อย่างไรก็ตาม โมเด็มจะ กลับคืนสู่โหมดคำสั่งภายใต้สถานการณ์ต่อไปนี้:

- เมื่อการโทรศัพท์ถูกตัดการติดต่อ และโมเด็มอยู่ในโหมดออฟไลน์
- เมื่อโมเด็มไม่สามารถติดต่อโทรศัพท์ได้อย่างสมบูรณ์ หรือ ตัวแคริเออร์ข้อมูลของโมเด็มทางไกลหลุด
- เมื่อโมเด็มได้รับสัญญาณให้ออกหรือหยุดในขณะที่อยู่ในโหมดออนไลน์
- เครื่องหมายอัฒภาค (semicolon) ปรากฏขึ้นที่ท้ายสตริงโทรศัพท์

ซึ่งถ้าหากมีข้อผิดพลาดเกิดขึ้นในช่วงปฏิบัติตามคำสั่ง การติดต่อจะหยุดลง และคำสั่งทุกอย่างที่ตามหลังคำสั่งที่ผิดพลาดนั้นจะถูกเพิกเฉย

คำสั่ง AT และรายละเอียด

AT (Attention Code) AT คือตัวอักษรเสริมหน้าคำสั่ง เป็นตัวบอกโมเด็มว่าได้มีการป้อนคำสั่งหรือ คำสั่งต่างๆแล้ว รหัสนี้จะอยู่หน้าคำสั่งทุกชนิด ยกเว้น คำสั่ง A/ (repeat) และ +++ (escape) การป้อนคำสั่ง AT อย่างเดียวจะมีผลทำให้โมเด็มตอบสนองด้วยสถานะ OK หรือ 0 ในกรณีที่พร้อมรับคำสั่งอื่น

A/ ทำซ้ำคำสั่งสุดท้าย คำสั่งA/ จะมีผลทำให้โมเด็มทำซ้ำคำสั่งก่อนหน้านี้ เช่น การหมุนหมายเลขโทรศัพท์ซ้ำอีกครั้งหนึ่ง คำสั่งที่ได้ปฏิบัติก่อนหน้านี้จะคงอยู่ในบัฟเฟอร์คำสั่งจนกระทั่งได้ป้อนคำสั่ง AT หรือได้ปิดเครื่อง ทั้งสองวิธีการจะล้างบัฟเฟอร์และทำให้คำสั่ง A/ เป็นโมฆะ เพราะไม่มีคำสั่งใดที่จะทำซ้ำอีกไม่จำเป็นที่จะต้องป้อน <cr> หรือ AT เพราะคำสั่งทั้งสองจะถูกจัดเก็บอยู่ในบัฟเฟอร์คำสั่งกับคำสั่งก่อนหน้านี้ แล้ว

A คำสั่งตอบรับ (Answer Command) A จะทำให้โมเด็มตอบรับโทรศัพท์โดยไม่ต้องรอสัญญาณเสียงโทรศัพท์ สิ่งนี้เป็นประโยชน์ในการตอบรับโทรศัพท์แบบแมนวล หรือเมื่อต่อกับโมเด็มอื่นในโหมดดั้งเดิมโดยตรง

หมายเหตุ: คำสั่งอื่นใดก็ตามที่ตามด้วยคำสั่ง A บนคำสั่งเดียวกันจะถูกละเว้นไม่ปฏิบัติตามคำสั่งนั้น

คำสั่ง A จะไม่ได้รับอนุญาตให้ใช้ในบางประเทศ ในกรณีนี้ คำสั่ง ATA จะเปลี่ยนสถานะเป็นข้อผิดพลาด

Bn ตัวเลือกมาตรฐานการสื่อสาร (Communications Standard Option) เป็นตัวตัดสินมาตรฐาน ITU กับ มาตรฐาน Bell

ค่าพารามิเตอร์: n = 0 - 3, 15, 16

n = 0 ITU V.22 สำหรับ 1200 bps

n = 1 Bell 212A สำหรับ 1200 bps (ดีฟอลต์)

n = 2, 3 ไม่เลือก ITU V23 แชนเนลกลับด้าน (reverse channel)

n = 15 ITU V.21 สำหรับ 300 bps

n = 16 Bell 103J สำหรับ 300 bps (ดีฟอลต์)

Dn คำสั่งในการหมุน (Dial Command) D จะทำให้โมเด็มหมุนหมายเลขที่ตามหลัง D ในคำสั่งนั้น คำจำกัดความของหมายเลขการหมุน และตัวแปลงการหมุน จะอยู่ในตารางตัวแปลงการหมุน

หมายเหตุ: ในบางประเทศ จะต้องป้อนหมายเลขโทรศัพท์ตามหลังคำสั่ง D

ในโทรศัพท์ระบบพัลซ์ ตัวอักษรที่ไม่เป็นตัวเลขจะไม่มีผล

En ตัวเลือกคำสั่งเสียงสะท้อน (Command Echo Option) ทำหน้าที่ตั้งให้ทำงานหรือตัดการทำงานเสียงสะท้อนของตัวอักษรที่ป้อนเข้าไปในขณะที่โมเด็มอยู่ในโหมดคำสั่ง

ค่าพารามิเตอร์: n = 0, 1

n = 0 ตัดการทำงานของเสียงสะท้อน

n = 1 ตั้งให้ค่าเสียงสะท้อนทำงาน (ดีฟอลต์)

Hn ตัวเลือกการควบคุมการต่อเชื่อม Hn ทำหน้าที่ควบคุม รีเลย์ที่ต่ออยู่

ค่าพารามิเตอร์: n = 0, 1

n = 0 ต่อสายโมเด็มอยู่ (วางสาย) (ดีฟอลต์)

n = 1 ไม่ได้ต่อสายโมเด็ม

หมายเหตุ: ไม่อนุญาตให้ใช้คำสั่ง H1 ในบางประเทศ ในกรณีนี้ ATH1 จะเปลี่ยนสถานะเป็นข้อผิดพลาด

In ตัวเลือกแสดงการระบุ (Request Identification Option) เพื่อถามโมเด็มเกี่ยวกับรหัสสินค้า ตรวจสอบผลรวมของ ROM (ROM Checksum) หรือ สถานะการตรวจสอบผลรวมของ ROM (ROM Checksum Status)

ค่าพารามิเตอร์: n = 0, 1, 2, 4, 9

n = 0 กลับไปที่เวอร์ชันเฟิร์มแวร์

n = 1 คำนวณการตรวจสอบผลรวมของ ROM และแสดงค่า (เช่น 12AB)

n = 2 ตรวจสอบ ROM คำนวณและตรวจเช็คค่าผลรวม แสดงผลในรูป OK หรือ ERROR

n = 4 กลับไปที่เวอร์ชันซอฟต์แวร์ของข้อมูล

n = 9 กลับไปที่รหัสประเทศ

Ln ระดับเสียงลำโพงจอภาพ (MonitorSpeaker Volume) ATLn จะตั้งค่าระดับเสียงลำโพงในระหว่างการสื่อสารทางแฟกซ์และข้อมูล ให้อยู่ในระดับต่ำ กลาง หรือ สูง.

ค่าพารามิเตอร์: n = 0 – 3

n = 0 ระดับเสียงต่ำ

n = 1 ระดับเสียงต่ำ

n = 2 ระดับเสียงปานกลาง (ดีฟอลต์)

n = 3 ระดับเสียงสูงหมายเหตุ: ใช้คำสั่ง M0 ในการปิดลำโพง

Mn ตัวเลือกควบคุมลำโพง (Speaker Control Option) Mn ควบคุมการปิด/เปิดลำโพงในระหว่างการสื่อสารทางข้อมูลและแฟกซ์

ค่าพารามิเตอร์: n = 0 – 3

n = 0 ปิดลำโพง

n = 1 เปิดลำโพงจนกว่าตรวจพบแคริเออร์ (ดีฟอลต์)

n = 2 เปิดลำโพงเมื่อไม่ได้ต่อสายโมเด็ม

n = 3 เปิดลำโพงหลังจากที่หมุนจนกว่าตรวจพบแคริเออร์

Nn Modulation Handshake Nn ทำหน้าที่ควบคุมว่าโมเด็มในเครื่องคอมพิวเตอร์สามารถติดต่อกับโมเด็มทางไกลในช่วงเวลาที่ติดต่อกัน เมื่อความเร็วในการสื่อสารของทั้งสองโมเด็มต่างกันได้หรือไม่

ค่าพารามิเตอร์: n = 0, 1

n = 0 เมื่อเริ่มต้น หรือ ตอบรับ จะเริ่มลองติดต่อเมื่อมาตรฐานการสื่อสารถูกระบุโดยคำสั่ง S37 และ คำสั่ง ATB

n = 1 เมื่อเริ่มต้น หรือ ตอบรับ จะเริ่มลองติดต่อ ก็ต่อเมื่อมาตรฐานการสื่อสารถูกระบุโดยคำสั่ง S37 และ คำสั่ง ATB ในระหว่างที่เริ่มลองติดต่อ อาจมีการลดความเร็วลง (ดีฟอลต์).

On คำสั่งออนไลน์ (Online Command) เป็นการบังคับโมเด็มให้อยู่ในโหมดออนไลน์

ค่าพารามิเตอร์: n = 0, 1, 3

n = 0 อยู่ในสถานะออนไลน์

n = 1 อยู่ในสถานะออนไลน์ และเริ่มตั้งค่า equalizer retrain ก่อนกลับเข้าสู่โหมดข้อมูลออนไลน์

n = 3 อยู่ในสถานะออนไลน์ และตั้งค่าอัตราการลองทำใหม่ ก่อนกลับเข้าสู่โหมดข้อมูลออนไลน์หมายเหตุ: ใช้คำสั่งนี้ในการกลับเข้าสู่โหมดออนไลน์หลังจาก "escaping " ไปที่โหมดคำสั่ง

P หมุนระบบพัลซ์ (Pulse Dial) P ตั้งค่าโหมดการหมุนเป็นแบบระบบพัลซ์ โทรศัพท์ทุกสายจะอยู่ในระบบพัลซ์ จนกว่าจะมีการปรับไปสู่ระบบโทน (คำสั่ง T)คำสั่งนี้สามารถใช้ได้กับ ตัวแปลงการหมุน.

หมายเหตุ: ไม่มีระบบพัลซ์ในบางประเทศ ไม่สามารใช้คำสั่ง P ได้ในประเทศเหล่านั้น

Qn การบีบอัดรหัสผลลัพธ์ (Result Code Suppression) Qn จะตั้งค่าให้โมเด็มทำงานในการส่งรหัสผลลัพธ์

ค่าพารามิเตอร์: n = 0, 1

n = 0 ตั้งให้รหัสผลลัพธ์ ทำงาน(ดีฟอลต์)

n = 1 ตัดการทำงานการกลับไปสู่รหัสผลลัพธ์ (quiet)

T การหมุนระบบโทน (Tone Dial) T กำหนดโหมดการหมุนโทรศัพท์ให้เป็นระบบโทน ซึ่งเป็นค่าในโหมดดีฟอลต์คำสั่งนี้สามารถใช้เป็น ตัวแปลงการหมุนได้

Vn ตัวเลือกรูปแบบรหัสผลลัพธ์ (Result Code Form Option) Vn กำหนดชนิดของ รหัสผลลัพธ์

ค่าพารามิเตอร์: n = 0, 1

n = 0 ส่งรหัสผลลัพธ์เป็นตัวเลข (รูปแบบอย่างสั้น หรือ ตัวเลข)

n = 1 ส่งรหัสผลลัพธ์เป็นข้อความ(รูปแบบอย่างยาว หรือ verbose) (ดีฟอลต์)

Xn ตัวเลือกชุดรหัสผลลัพธ์ /รายงานผลการโทร(Result Code/Call Progress Option) Xn เลือกชุดรหัสผลลัพธ์ และฟังก์ชันการหมุนโทรศัพท์ คำสั่ง Vn จะกำหนดว่ารหัสผลลัพธ์จะถูกส่งในรูปของตัวอักษรหรือตัวเลข โปรดดูคำจำกัดความรหัสผลลัพธ์

ค่าพารามิเตอร์: n = 0 – 4

n = 0 ตั้งรหัสผลลัพธ์ CONNECT ให้ทำงาน และตัดการทำงานรหัสผลลัพธ์ของ CONNECT XXXX สัญญาณสายไม่ว่างและสัญญานการหมุนโทรศัพท์จะตรวจไม่พบ

n = 1 หมุนโมเด็มแบบสุ่ม (blind dial): รหัสผลลัพธ์ CONNECT XXXX จะถูกตั้งให้ทำงาน สัญญาณสายไม่ว่างและสัญญานการหมุนโทรศัพท์จะตรวจไม่พบ

n = 2 โมเด็มรอคอยสัญญาณการหมุนก่อนเริ่มหมุน รหัสผลลัพธ์ CONNECT XXXX จะถูกตั้งให้ทำงาน สัญญาณสายไม่ว่างจะตรวจไม่พบ

n = 3 หมุนโมเด็มแบบสุ่ม (blind dial): รหัสผลลัพธ์ CONNECT XXXX จะถูกตั้งให้ทำงาน โมเด็มจะส่งรหัสผลลัพธ์ BUSY ถ้าตรวจพบสัญญาณสายไม่ว่าง

n = 4 โมเด็มรอคอยสัญญาณการหมุนก่อนเริ่มหมุน รหัสผลลัพธ์ CONNECT XXXX จะถูกตั้งให้ทำงาน โมเด็มส่งจะรหัสผลลัพธ์ BUSY ถ้าตรวจพบสัญญาณสายไม่ว่าง (ดีฟอลต์)

Z ตัวเลือกคำสั่งรีเซ็ต (Reset Command Option) คำสั่งนี้จะสั่งให้โมเด็มอยู่ในสถานะพร้อมทำงาน และจะกู้ไฟล์ที่บันทึกโดยคำสั่ง &W อันสุดท้ายกลับคืนมา

+++ ลำดับ Code Escape (Escape Code Sequence) เมื่อได้ส่งชุดตัวอักษรในรีจิสเตอร์ S2 ไปยังโมเด็ม 3 ครั้งที่ประสบความสำเร็จโดยเร็ว โมเด็มจะออกไปที่สถานะคำสั่งนั้น ค่าดีฟอลต์สำหรับตัว

อักษร escape คือ + เมื่อเอกสารบอกให้ป้อน +++ ก็ให้ป้อนชุดตัวอักษรในรีจิสเตอร์ S2 สามครั้งที่ประสบความสำเร็จโดยเร็ว ห้ามเริ่มต้นลำดับของรหัส escape ด้วยคำสั่ง AT และห้ามกด Enter หลังจากนั้น

2.3.3 ประเภทของสัญญาณโทรศัพท์

สัญญาณที่ส่งจากผู้เช่ากับชุมสาย

1. Off-hook คือ สภาพที่ผู้เช่ายกหูโทรศัพท์สายจะมีสภาพ Closed loop (Low impedance)

2. On-hook คือ สภาพผู้เช่าวางหู หรือ สภาพว่าง สายจะมีสภาพ Open loop (High impedance)

3. Dialing คือ สภาพที่ผู้เช่าหมุนหมายเลขเข้าเครื่องเป็น Rotary dial สัญญาณจะเป็น Pulsing ค่า Impedance จะสูง,ต่ำ สลับกันไปตามที่หมุนเลขหมาย ถ้าเป็นเครื่องแบบกดปุ่ม Touch – Tone สัญญาณออกจะเป็น ความถี่ DTMF ส่งออกไปชุมสาย

สัญญาณที่ส่งมาจากชุมสาย

1. Dialing tone คือ สัญญาณที่บอกถึงสภาพการว่างของอุปกรณ์ชุมสาย และ ชุมสายพร้อมที่จะรับ Code ที่ทำการหมุนเข้ามา สัญญาณ Dialing tone นี้จะเป็นสัญญาณต่อเนื่องความถี่ 400 Hz Modulated ด้วย 50 Hz ผู้เช่าจะได้ยินเมื่อทำการยกหูโทรศัพท์

2. Busy tone คือ สัญญาณที่บอกให้ทราบว่า อุปกรณ์ชุมสายไม่ว่าง แต่ถ้ายกหูแล้วได้ยินสัญญาณนี้แสดงว่า อุปกรณ์ในชุมสายไม่ว่างและถ้าได้ยินเสียงนี้หลังจากหมุนหมายเลขไปแล้ว แสดงว่าผู้เช่าฝ่ายถูกเรียกไม่ว่าง ในกรณีเรียกต่างชุมสาย ลักษณะสัญญาณที่ส่งจะเป็นสัญญาณที่ขาดตอนเป็นช่วงๆ ส่ง 0.5 วินาที หยุด 0.5 วินาที ความถี่ของสัญญาณ 400 Hz sine wave

3. Ringing tone เป็นสัญญาณที่ผู้เรียกได้ยินหลังจากหมุนหมายเลขครบแล้ว เพื่อ บอกให้ทราบว่าการต่อทำได้สำเร็จ ในขณะนี้จะส่งสัญญาณเรียก (Ringing signal) ไปยังผู้ถูกเรียก ความถี่ของสัญญาณ 400 Hz sine wave โดยจะส่ง 1 วินาที หยุด 4 วินาที

4. Ringing signal เป็นสัญญาณต่อเนื่องความถี่ของสัญญาณ 25 Hz ค่าแรงดัน 70-90 Vrm โดยส่งไปยังผู้เช่าฝ่ายถูกเรียก ส่ง 4 วินาที หยุด 4 วินาที

5. สัญญาณโทนอื่นๆ เช่น Nu tone (Number Unobtainble Tone) บอกให้ทราบว่า เลขหมายที่หมุนมาไม่มีการใช้งานอยู่ เป็นต้น

บทที่ 3
การคำนวณและการสร้าง

3.1 หลักการทำงานของโปรแกรมการต่อหมายเลขโทรศัพท์โดยใช้เสียง (Speech Telephone)

โปรแกรมการต่อหมายเลขโทรศัพท์โดยใช้เสียง ประกอบด้วย 2 ส่วนหลัก ได้แก่

3.1.1 ส่วนของโมเดลเสียง

3.1.2 ส่วนของการต่อหมายเลขโทรศัพท์

โปรแกรมทั้ง 2 ส่วนนี้เขียนและรันโปรแกรมด้วยภาษา Visual C++

3.1.1 ส่วนของโมเดลเสียง

ในส่วนของโมเดลเสียงนี้จะเป็นการวิเคราะห์เสียงซึ่งสามารถแบ่งออกเป็น 3 ขั้นตอนหลักคือ

3.1.1.1 ขั้นตอนของการตัดคำให้เป็นคำเดี่ยว

3.1.1.2 ขั้นตอนการเรียนรู้

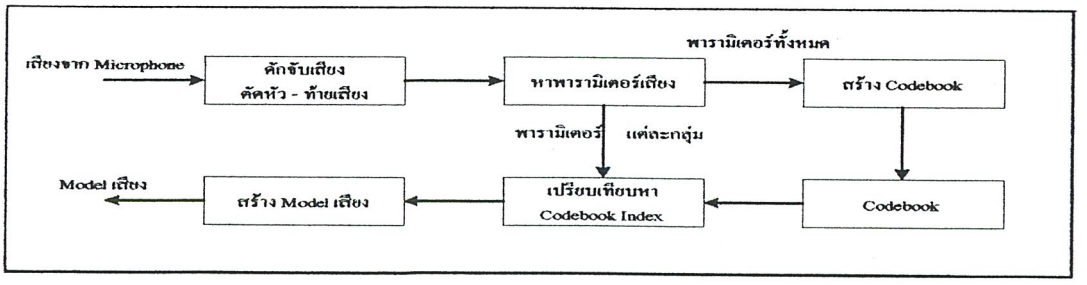
3.1.1.3 ขั้นตอนการวิเคราะห์จดจำ

3.1.1.1 ขั้นตอนของการตัดคำให้เป็นคำเดี่ยว

ในขั้นตอนนี้จะทำการหาค่าขนาดกำลังสอง เพื่อใช้ในการกำหนดขอบเขตของคำ และทำการตัดคำให้เป็นคำเดี่ยวเพื่อใช้ในการจดจำ โดยจะมีขั้นตอนย่อย ๆ ดังนี้

- ในขั้นตอนนี้เราจะทำการกำหนดค่าพารามิเตอร์ต่าง ๆ เพื่อใช้ในการคำนวณ เช่น ขนาดของเฟรม ขนาดความกว้างที่ใช้ในการเลื่อนเฟรม โดยค่าพารามิเตอร์จะถูกนำไปใช้ในการหาค่าขนาดกำลังสองของสัญญาณเสียงที่ป้อนเข้าไป
- อ่านข้อมูลของสัญญาณเสียง และทำการคำนวณหาค่าเฟรมทั้งหมดของเสียงนั้น
- คำนวณหาค่ากำลังสองในแต่ละเฟรมจนครบเฟรม
- ทำการหาค่าขนาดกำลังสองเฉลี่ยจากเสียงเพื่อที่จะทำการกำหนดขอบเขตของคำแต่ละคำอย่างคร่าว ๆ แต่การกำหนดขอบเขตของคำนั้นจะไม่สามารถครอบคลุมคำได้ทั้งหมด
- ทำการหาค่าขนาดกำลังสองเฉลี่ยระดับต่ำ เพื่อช่วยในการขยายขอบเขตของคำ และทำให้สามารถครอบคลุมขอบเขตของคำได้เกือบทั้งหมด
- ทำการขยายขอบเขตของคำโดยใช้ค่าขนาดกำลังสองเฉลี่ยระดับต่ำอีกครั้ง
- นำขอบเขตคำที่ได้มาใช้ในการตัดแบ่งคำให้เป็นคำเดี่ยว

3.1.1.2 ขั้นตอนการเรียนรู้



รูปที่ 3.1 แสดงขั้นตอนการเรียนรู้

เป็นขั้นตอนที่ต้องการสร้างแบบจำลองของเสียงขึ้นมา เพื่อนำไปใช้ในขั้นตอนการรู้จำเสียงพูดต่อไป ประกอบด้วยส่วนย่อย ๆ ดังนี้

การวิเคราะห์สัญญาณเสียงเบื้องต้น

มีการเลือกใช้ ค่าต่าง ๆ ในการคำนวณและออกแบบออกโปรแกรมดังนี้

1. การพรีเอมฟาซีส ใช้วงจรอันดับหนึ่ง ซึ่งมีฟังก์ชั่นถ่ายโอน คือ

$$H(z) = 1 - \alpha z^{-1} \qquad (3.1)$$

ค่า α ที่ใช้คือ 15/16 = 0.9375

2. การแบ่งช่วงสัญญาณ ขนาดของช่วงสัญญาณมีเงื่อนไขในการเลือก คือ

- ค่า N ต้องสั้นพอที่คุณสมบัติของเสียงไม่เปลี่ยนแปลง
- ค่า N ต้องยาวพอที่จำนวนของตัวอย่างมีเพียงพอสำหรับการหาสัมประสิทธิ์
- การเลื่อนในการวิเคราะห์(ค่า M) ต้องไม่ข้ามข้อมูล

ดังนั้นค่า M จะน้อยกว่า N แต่ถ้าค่า M มีขนาดเล็กเกินไปจะทำให้การคำนวณช้าลง ดังนั้นจึงเลือกค่า M = 100 แซมเปิ้ล และค่าN = 300 แซมเปิ้ล

3. ความถี่ที่ใช้ในการสุ่มสัญญาณ เนื่องจากความถี่ที่ใช้ในการแซมปลิ้ง มากกว่าหรือเท่ากับสองเท่าของความถี่เสี่ยง ($f_s \geq f_N$) ดังนั้น $f_s \geq$ 8 KHz แต่เนื่องจากใน wave studio มีความถี่ที่ใกล้เคียงที่สุดคือ 11.025 Khz

ดังนั้น ช่วงเวลาที่ใช้ในการวิเคราะห์แต่ละเฟรม คือ 300/11.025 =27.21 ms และระยะที่ใช้ในการเลื่อนเฟรม คือ 100/11.025 =9.07 ms

4. การเลือกวินโดว์ที่เหมาะสมสำหรับการวิเคราะห์เสียง

โดยพิจารณาลักษณะสเปคตรัมคือ คือ

- ความถี่เรโซลูชั่นสูง (high frequency resolution) คือ มีโลบหลักแคบและแหลม
- การลดทอน (Attenuation) นอกช่วงความถี่ที่ผ่านได้ต่ำ คือ ไซด์โลบมีค่าน้อยฟังก์ชั่นวินโดว์มีหลายชนิด แต่ชนิดที่เหมาะสมที่สุดที่นำมาใช้ได้แก่ แฮมมิ่งวินโดว์ ซึ่งมีค่าไซด์โลบต่ำ (-40 dB) และเมนโลบแคบพอใช้

5. การหาคุณลักษณะของเสียง ใช้การประมาณพันธะเชิงเส้นในการวิเคราะห์หาค่าสัมประสิทธิ์ LPC ซึ่งการประมาณเชิงเส้นที่เลือกใช้ คือ วิธีอัตสัมพันธ์ (autocorrelation method) ซึ่งวิธีนี้มีการคำนวณที่ง่ายกว่าวิธีอื่น ๆ และมีความแน่นอนด้านเสถียรภาพ อีกทั้งมีวิธีการเก็บข้อมูลที่น้อยกว่า

เนื่องจากเป็นวิเคราะห์โดยวิธีอัตสัมพันธ์ อันดับการประมาณเชิงเส้น (P) ที่มากจะทำให้การประมาณเสียงมีความใกล้เคียงมากยิ่งขึ้น แต่ถ้าอันดับ P มีค่ามากเกินไปจะทำให้การคำนวณมีความยุ่งยากและใช้เวลานาน ดังนั้น เพื่อความเหมาะสมค่าอันดับ P ที่ใช้ คือ 12

การสร้างโค้ดบุก

จากการทดลอง (L.R. Rabiner, S.E. Levinson และ Sondhi, 1982) จะได้ว่าที่ขนาดโค้ดบุคเท่ากับ 64 จะมีค่าความคลาดเคลื่อนเฉลี่ยประมาณ 0.2 ซึ่งเป็นค่าที่น้อยมากสำหรับเวคเตอร์ควอนไตซ์เซชั่น

การวัดค่าความคลาดเคลื่อน ใช้วิธีการคำนวณแบบ square error distortion ในการหาระยะทาง เนื่องจากเป็นวิธีที่ง่าย และรวดเร็ว

การสร้างโค้ดบุค โดยนำเทรนนิ่งเซตที่ได้จากการประมาณเชิงเส้นมาผ่านกระบวนการดังนี้

1. สุ่มค่าเริ่มต้น a, b และ กำหนดให้ π = [100000] ตามเงื่อนไขในการใช้แบบจำลองแบบ Left – right

2. หาค่า α,β จากค่า a,b เริ่มต้น และลำดับค่าปรากฏ O = $\{O_1\ O_2\ O_3 \ldots O_t\}$ ซึ่งเรียกว่าลำดับเทรนนิ่ง ตามวิธีของ Forward – Backward Procedure โดยใช้ลำดับของค่าปรากฏหลาย ๆ ลำดับเข้ามาเทรน เพื่อความถูกต้องมากขึ้น

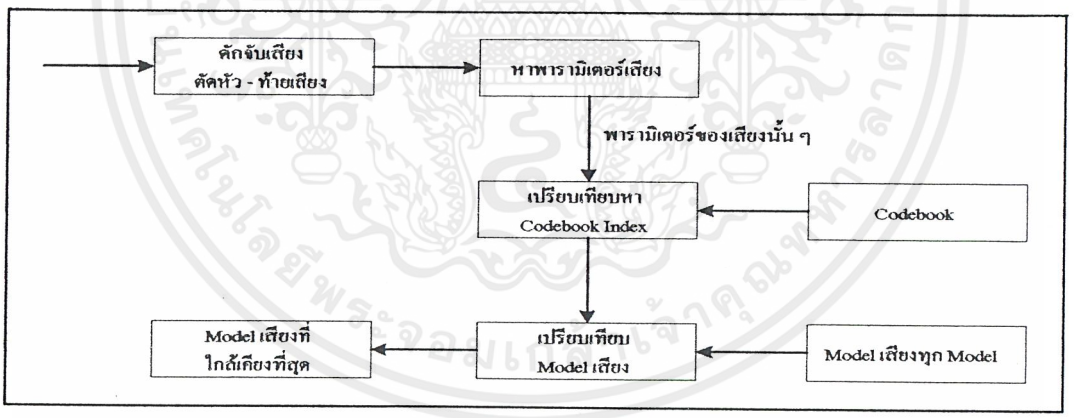
3. ทำการสเกลลิ่ง α เพื่อให้ค่าอยู่ในย่านที่คอมพิวเตอร์สามารถคำนวณได้อย่างถูกต้อง

4. หาค่าพารามิเตอร์ a,b,π ที่ให้ค่าความน่าจะเป็นสูงสุด ที่จะเป็นแบบจำลอง λ ที่เหมาะสมของคำ

5. ตรวจสอบค่าพารามิเตอร์ของแบบจำลองที่ได้ว่า ลู่เข้าหรือยัง โดยใช้วิธีการคำนวณค่า a,b ซ้ำประมาณ 50 รอบ เมื่อการเปลี่ยนแปลงน้อยมากจนเป็นที่พอใจตามระดับค่าที่ตั้งไว้ในที่นี้ใช้ค่าเท่ากับ 10^{-5} ก็จะหยุด และได้ค่าพารามิเตอร์ a,b และ π ของแบบจำลองที่ต้องการ

6. เก็บค่าพารามิเตอร์ a,b.π ที่ได้จากข้อ 5. เป็นพารามิเตอร์ของแบบจำลองไว้

3.1.1.3 ขั้นตอนการวิเคราะห์จดจำ



รูปที่ 3.2 แสดงขั้นตอนการวิเคราะห์และจดจำ

ในขั้นตอนนี้จะมีขั้นตอนย่อย ๆ อีก 2 ขั้นตอน ดังนี้

1. การหาดัชนีโค้ดบุค

โดยการนำเวกเตอร์เสียงจากการประมาณเชิงเส้นมาทีละเสียง แล้วเปรียบเทียบกับโค้ดบุคที่ได้จากการสร้างในขั้นตอนการเรียนรู้ ทีละเฟรม โดยวิธีความคลาดเคลื่อนกำลังสอง เวกเตอร์เสียงห่างจาก

โค้ดบุคใดน้อยที่สุดจะได้ว่าเป็นดัชนีโค้ดบุคของเฟรมเสียงนั้น และเก็บดัชนีโค้ดบุคของแต่ละเฟรมในแต่ละเสียงไว้เป็นลำดับค่าปรากฏ (observation sequence) สำหรับการสร้างแบบจำลองต่อไป

2. การรู้จำเสียง

หลังจากที่ได้แบบจำลอง HMM ของแต่ละคำศัพท์แล้ว เมื่อมีลำดับของค่าปรากฏ $O = \{O_1\ O_2\ O_3 \ldots O_T\}$ ของเสียง unknown ซึ่งเป็นเสียงที่ต้องการทดสอบเข้ามา เราจะทำการคำนวณหาความน่าจะเป็น $P(OI\ \lambda)$ ทุกแบบจำลองของแต่ละคำศัพท์โดยใช้วิธี viterbi algorithm แล้วเลือกเอาคำศัพท์ที่มีความน่าจะเป็นสูงสุด ซึ่งก็คือ คำศัพท์ที่แบบจำลองจำได้นั่นเอง

3.1.2 ส่วนของการต่อหมายเลขโทรศัพท์

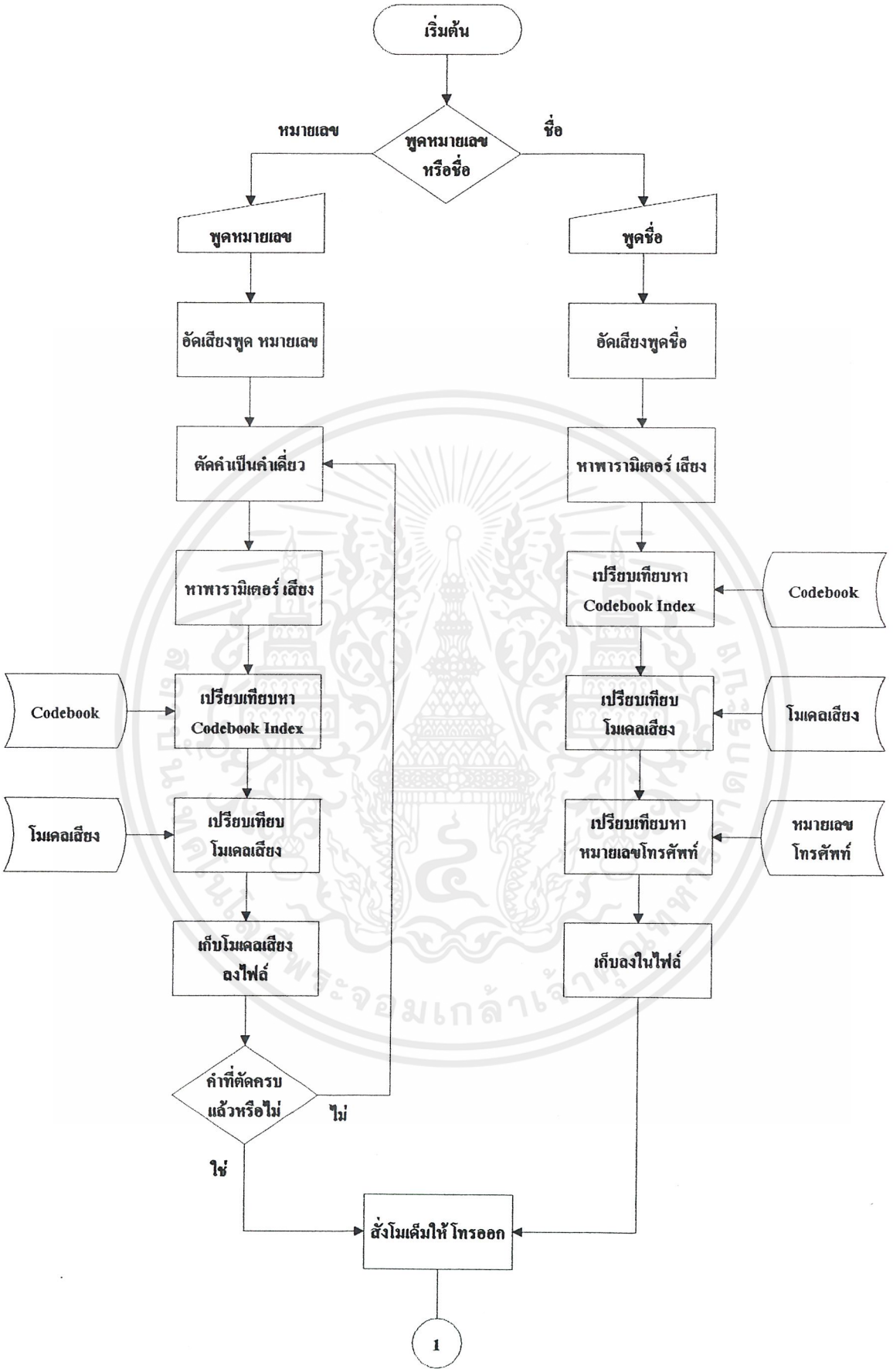
เมื่อส่วนของการรู้จำเสียงสามารถรู้จำได้ว่าเป็นคำ ๆ ใดหรือโมเดลใดแล้ว ก็จะกระทำตามกระบวนการต่าง ๆ โดยแบ่งออกเป็น 2 กรณี ดังต่อไปนี้

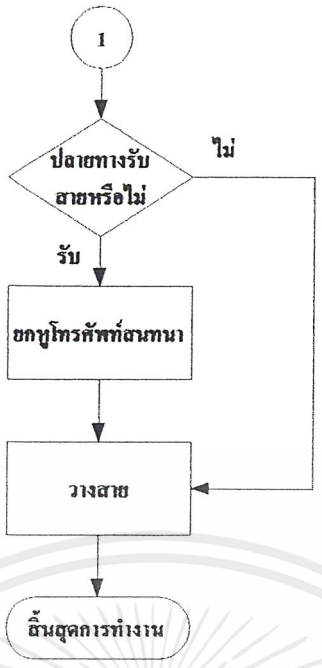
1. พูดหมายเลขโทรศัพท์
2. พูดชื่อที่มีเก็บไว้ในฐานข้อมูล

ซึ่งจะมีขั้นตอนการทำงานแตกต่างกันดังนี้

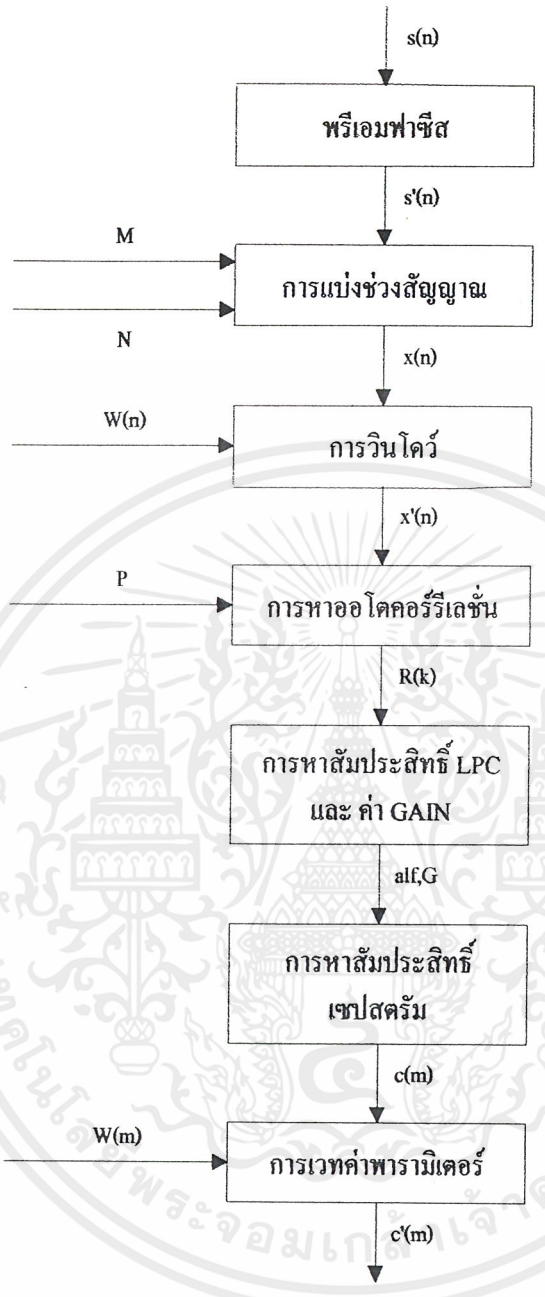
1. พูดหมายเลขโทรศัพท์ เมื่อพูดหมายเลขโทรศัพท์เข้าไป เช่น "9241878" ในส่วนของโปรแกรมการตัดคำจะทำการตัดคำเป็นคำเดี่ยว แล้วจะนำไปทำการหารู้จำเสียงดังที่ได้กล่าวมาแล้วข้างต้น ซึ่งจะได้ผลเป็นคำเดี่ยวคือ 9,2,4,1,8,7 และ 8 แล้วนำผลที่ได้ส่งไปให้โมเด็มต่อหมายเลขโทรศัพท์ออกไป โดยที่จะทำการรีเซตโมเด็มก่อนแล้วจึงทำการต่อโทรศัพท์ไปยังเลขหมายปลายทางที่เรียก

2. พูดชื่อที่มีเก็บไว้ในฐานข้อมูล เมื่อพูดชื่อเข้าไป เช่น "มล" โปรแกรมก็จะทำการนำคำที่ได้ไปผ่านขั้นตอนในการรู้จำเสียง ซึ่งเมื่อได้ผลลัพธ์ออกมาแล้วนำไปเทียบคำในฐานข้อมูลที่มีอยู่ว่าคำที่ได้ตรงกับหมายเลขโทรศัพท์หมายเลขใด ก็จะนำหมายเลขโทรศัพท์ดังกล่าวไปสั่งให้โมเด็มต่อไปยังหมายเลขนั้นเหมือนกระบวนการในข้อที่ผ่านมา

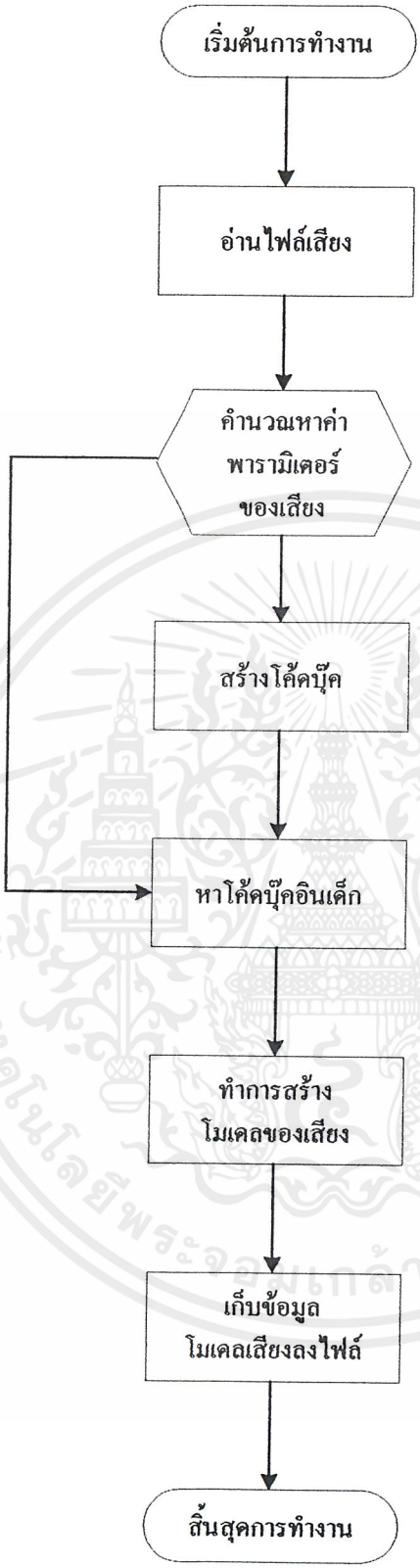




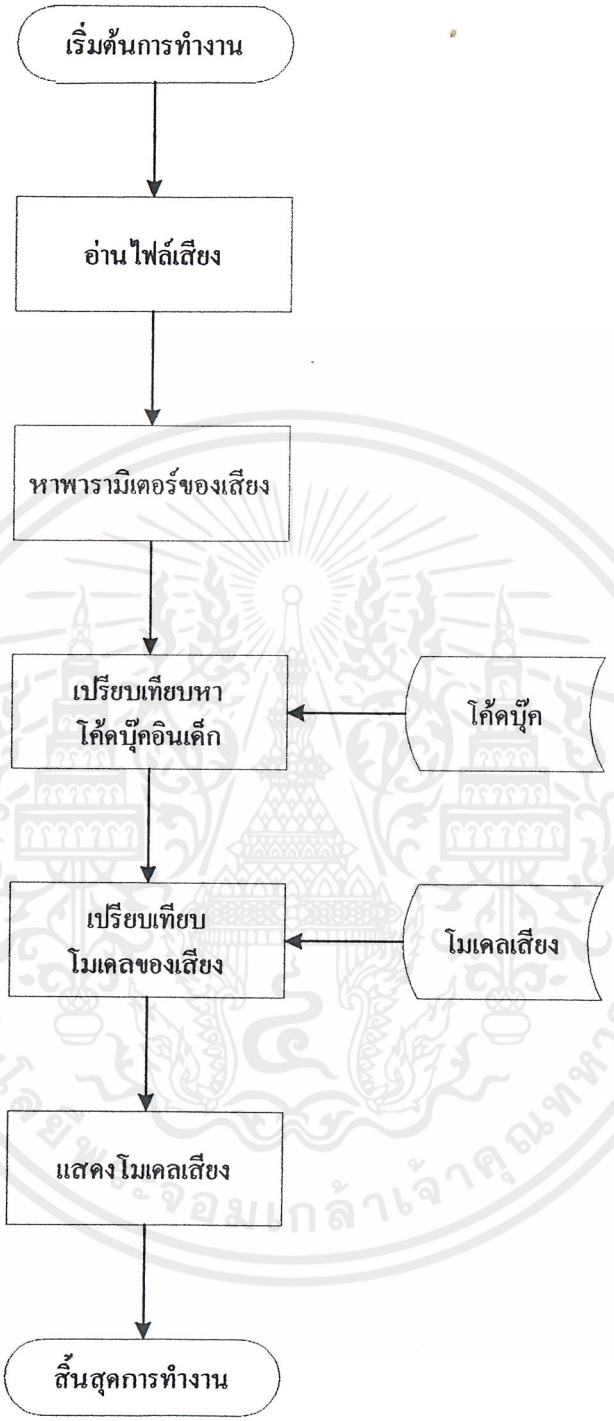
รูปที่ 3.3 แสดง flow Chart ของการทำงานของโครงงานทั้งหมด



รูปที่ 3.4 แสดงขั้นตอนการเตรียมสัญญาณในการวิเคราะห์



รูปที่ 3.5 แผนผังขั้นตอนการทำงานของโปรแกรมการเรียนรู้เสียงและโปรแกรมสร้างโมเดลเสียง



รูปที่ 3.6 แสดงแผนผังขั้นตอนการทำงานของโปรแกรมวิเคราะห์เสียง

3.2 การใช้งานโปรแกรม Speech Telephone

รูปที่ 3.7 แสดงโปรแกรมหลักของโครงงาน

โดยที่โปรแกรมหลักจะสามารถแบ่งออกเป็น 2 ส่วนหลักด้วยกันดังนี้

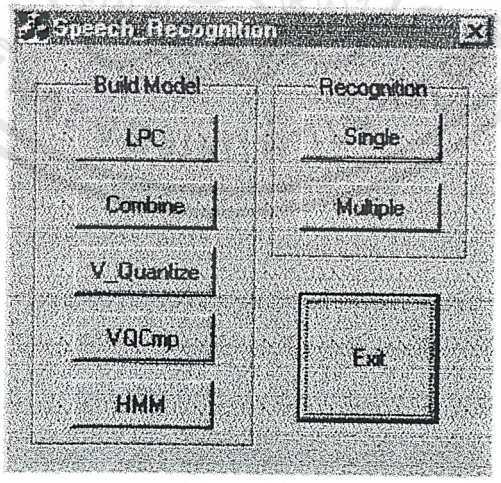
3.2.1 ส่วนของโมเดลเสียง

3.2.2 ส่วนของการสั่งโทรออก

โดยที่แต่ละส่วนจะมีรายละเอียดการใช้งานดังต่อไปนี้

3.2.1 ส่วนของโมเดลเสียง

ส่วนของโมเดลเสียงนี้จะเป็นส่วนที่เป็นโปรแกรม Speech Recognition ซึ่งเป็นโปรแกรมที่แยกย่อยออกไปจากโปรแกรมหลักดังรูปที่ 3.8



รูปที่ 3.8 แสดงโปรแกรม Speech Recognition

ซึ่งโปรแกรม Speech Recognition นี้สามารถแบ่งออกได้เป็น 2 ส่วนหลัก ดังนี้

3.2.1.1 ส่วนของการสร้างโมเดลเสียง

3.2.1.2 ส่วนของการจดจำเสียง

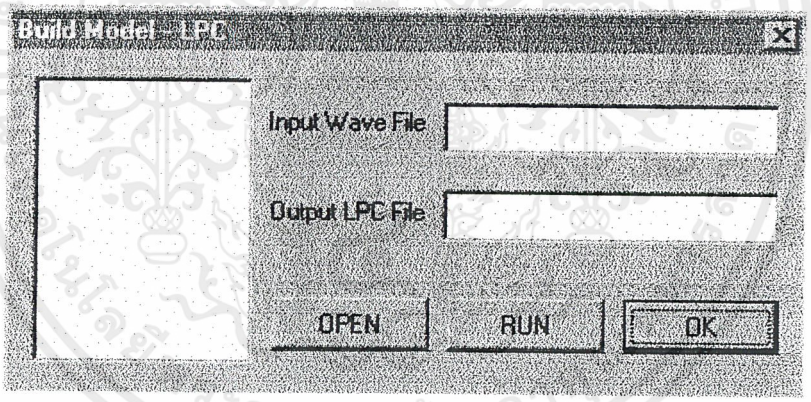
ในแต่ละส่วนจะมีการทำงานดังที่กล่าวมาตามรายละเอียดของโปรแกรม โดยที่ในแต่ละส่วนจะมีการทำงานแยกกันไป โดยจะกล่าวถึงรายละเอียดของการใช้งานดังต่อไปนี้

3.2.1.1 ส่วนของการสร้างโมเดลเสียง ก็จะมีโปรแกรมย่อยอีก 5 ส่วนย่อย ดังนี้

- LPC
- Combine
- V_Quantize
- VQcmp
- HMM

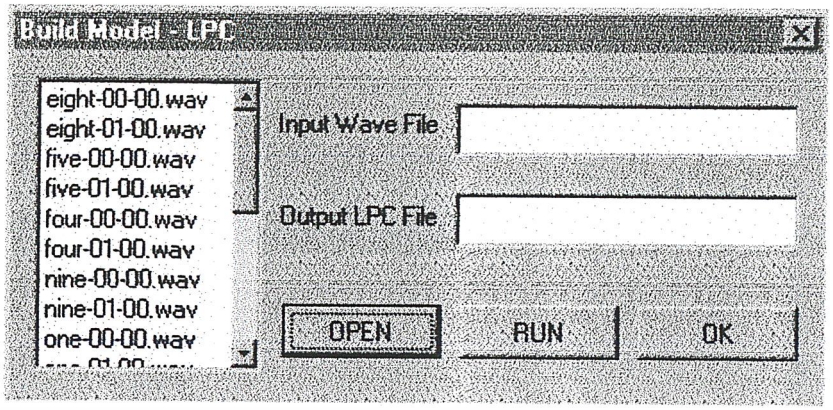
- LPC

โปรแกรมนี้เป็นโปรแกรมย่อยนี้มีหน้าที่ในการหาค่าสัมประสิทธิ์ LPC โดยที่ไฟล์อินพุทจะเป็นไฟล์เสียง ซึ่งในส่วนนี้จะมีรูปร่างดังรูป 3.9



รูปที่ 3.9 แสดงโปรแกรมในส่วนของการหาค่า LPC

การใช้งานนั้นจะมีดังนี้ เราจะเริ่มทำการคัดลอกไฟล์ที่ต้องการลงไปในไดเร็กทอรี่ (directory) ที่มีโปรแกรมอยู่ แล้วกดปุ่ม OPEN โปรแกรมก็จะทำการแสดงไฟล์ที่มีอยู่ในไดเร็กทอรี่ของโปรแกรม โดยที่มันจะแสดงใน "LIST BOX" (บล็อกซ้ายสุด) ดังรูปที่ 3.10 หลังจากนั้นเราต้องการที่จะหาค่าสัมประสิทธิ์ของ LPC ที่ไฟล์ใดก็ "DOUBLE-CLICK" ที่ไฟล์นั้น ชื่อไฟล์นั้นก็จะขึ้นที่อินพุทไฟล์และเอาท์พุทไฟล์โดยอัตโนมัติ หลังจากนั้นเราก็กดปุ่ม RUN โปรแกรมก็จะเริ่มทำการหาค่าสัมประสิทธิ์ LPC พอเสร็จแล้วก็จะมีแมสเสจบล็อกออกมาดังรูป 3.11 แสดงให้เรารู้ว่า โปรแกรมได้ทำการคำนวณเสร็จแล้ว และโปรแกรมนี้ได้ถูกออกแบบไว้พิมพ์ออกมาเป็นเท็กซ์ไฟล์ (Text File) ด้วย

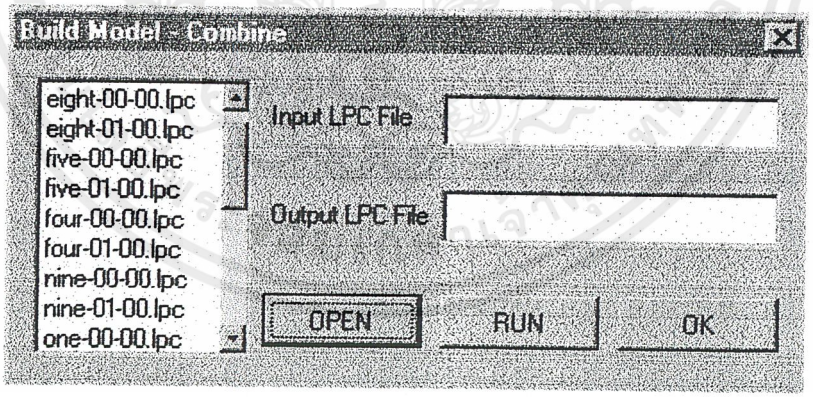


รูปที่ 3.10 แสดงไฟล์เมื่อเราทำการกดปุ่ม open

รูปที่ 3.11 แสดงแมสเสจบล็อคที่แสดงออกมาเมื่อโปรแกรมคำนวณหาค่าสัมประสิทธิ์ LPC เสร็จ

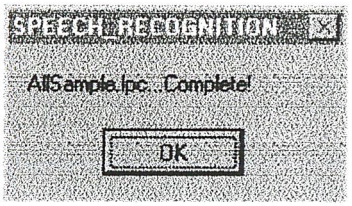
- Combine

โปรแกรมในส่วนนี้มีหน้าที่ในการรวมไฟล์ค่าสัมประสิทธิ์ LPC ทั้งหมดไว้ในไฟล์เดียว เพื่อใช้ในการคำนวณหาโค้ดบุคต่อไป ดังรูปที่ 3.11



รูปที่ 3.12 แสดงรูปโปรแกรมส่วนของการรวมไฟล์ค่าสัมประสิทธิ์ LPC

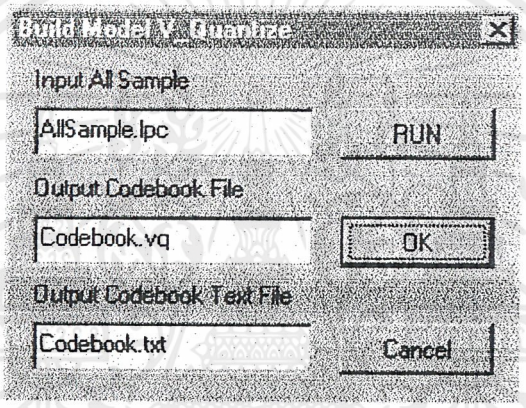
ส่วนการใช้งานก็จะคล้ายกับโปรแกรมส่วนของ LPC ปุ่มที่ใช้งานการใช้งานก็จะเหมือนกัน และเมื่อโปรแกรมทำงานเสร็จก็จะมีแมสเสจบล็อคแสดงบอกเหมือนกัน ดังรูปที่ 3.13



รูปที่ 3.13 แมสเสจบล็อคเมื่อโปรแกรมทำงานเสร็จ

- V_Quantize

โปรแกรมส่วนนี้มีหน้าที่ในการหาค่าการจัดระดับเวกเตอร์ หรือที่เรียกว่า Vector Quantization โดยการทำงานในส่วนนี้จะนำเอาค่าไฟล์ที่รวมค่าสัมประสิทธิ์ LPC (AllSample.lpc) นำมาทำการหาโค้ดบุค ดังรูปที่ 3.14

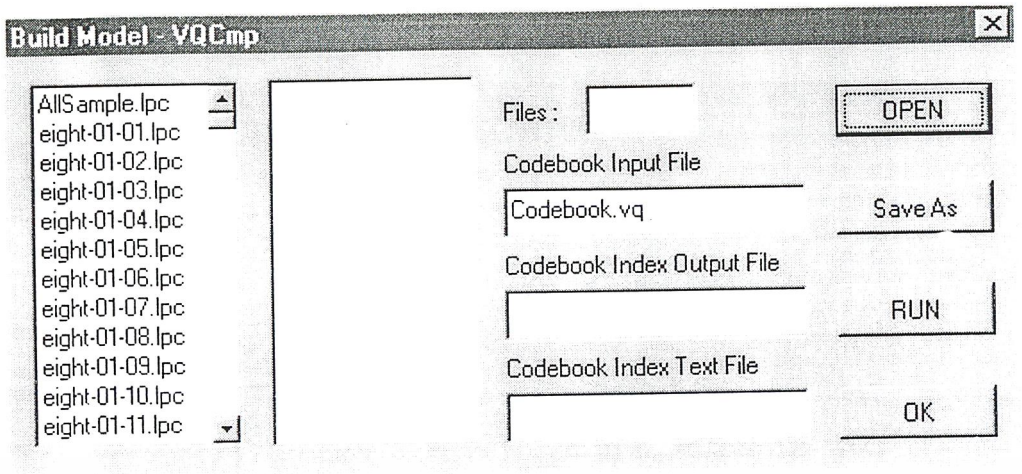


รูปที่ 3.14 แสดงรูปโปรแกรมย่อยของ V_Quantize

ในส่วนของการทำงานของโปรแกรมในส่วนนี้เราก็จะมีอินพุทไฟล์ "AllSample.lpc" ซึ่งเมื่อมีการเรียกโปรแกรมออกมาเป็นไฟล์ "Codebook.vq" และเมื่อโปรแกรมทำงานเสร็จแล้วก็จะมีแมสเสจบล็อคออกมาบอกเหมือนกัน

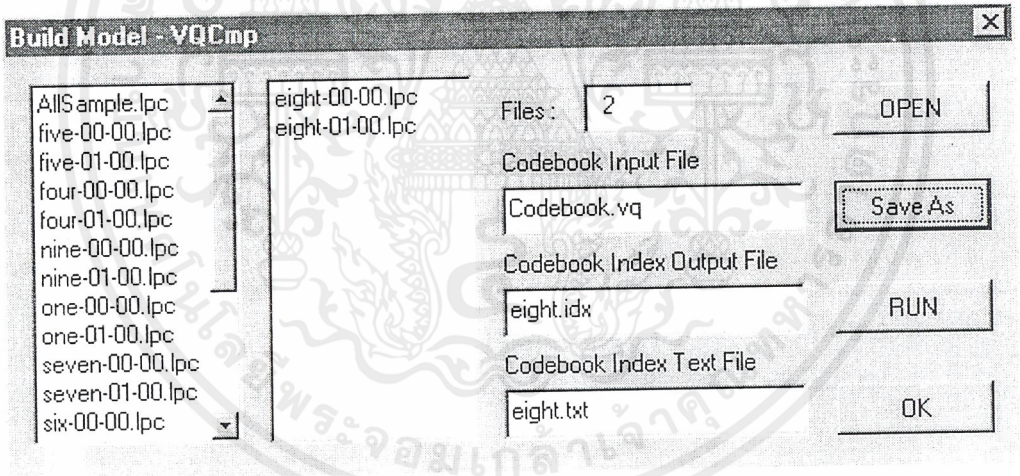
-VQcmp

โปรแกรมในส่วนนี้เป็นการทำการเปรียบเทียบค่าสัมประสิทธิ์ LPC ของแต่ละเสียงกับโค้ดบุคที่ได้ทำการสร้างไว้ เพื่อที่จะได้โค้ดบุคอินเด็ก (Codebook Index) ออกมา ดังรูปที่ 3.15



รูปที่ 3.15 แสดงโปรแกรมในส่วนของ Vqcmp

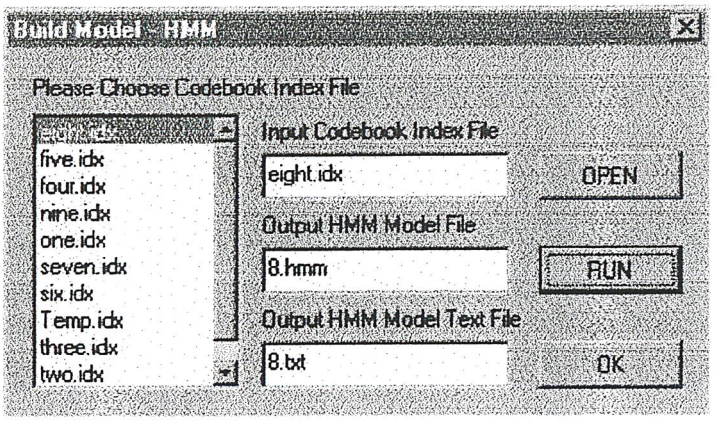
การใช้งานก็จะคล้าย ๆ กับโปรแกรมที่ได้กล่าวมาข้างต้น แต่จะซับซ้อนกว่าตรงที่ แต่ละเสียงเราอาจจะพูดหลายครั้งหรือหลายคนจึงต้องทำบล็อคเอาไว้แสดงไฟล์ต่างหาก และเราจำเป็นที่จะต้องทำการนับจำนวนไฟล์ว่าแต่ละเสียงนำมาคำนวณกี่เสียง และเมื่อเราเลือกไฟล์เป็นที่เรียบร้อยแล้ว เราก็จะทำการกด save asโปรแกรมก็จะทำการแสดงเอาท์พุทไฟล์อัตโนมัติ ดังรูปที่ 3.16



รูปที่ 3.16 แสดงการตั้งค่าต่าง ๆ พร้อมที่จะกดปุ่ม RUN

- HMM

โปรแกรมย่อยในส่วนนี้จะมีหน้าที่ใน การคำนวณหาโมเดลแบบเสียงเพื่อใช้ในการเปรียบเทียบในการจดจำเสียง ดังรูปที่ 3.17 โดยวิธีการแบบ Hidden Markov Model (HMM)

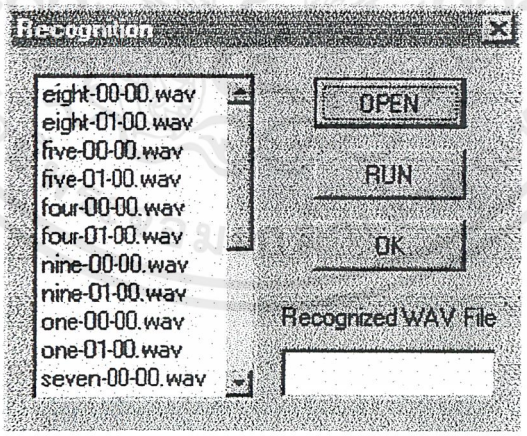


รูปที่ 3.17 แสดงโปรกรมในส่วนของ HMM

การใช้งานของโปรแกรมในส่วนนี้ก็จะคล้าย ๆ กับโปรแกรมก่อนหน้านี้ คือในการป้อนค่าอินพุท เราก็สามารถที่จะ Doble – Click ที่ไฟล์ได้เลย แต่ในส่วนของเอาท์พุทโปรแกรมในส่วนนี้ได้ออกแบบให้เราป้อนชื่อของโมเดลไฟล์ตามที่เราต้องการ โดยเราจะใช้นามสกุลของไฟล์เป็น “hmm” เช่น “1.hmm” ในส่วนของเอาท์พุทที่เป็นเท็กซ์ไฟล์เราได้ออกแบบให้มีชื่อเดียวกับโมเดลไฟล์ โดยจะเป็นโดยอัตโนมัติโดยที่เราไม่ต้องป้อนชื่อไฟล์เอง โดยที่เอาท์พุทเท็กซ์ไฟล์จะมีนามสกุลเป็น “.txt” เช่น “1.txt”

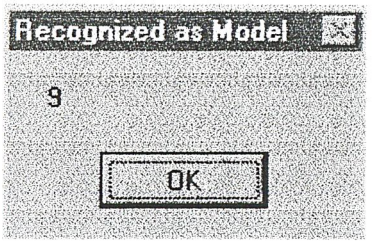
3.2.1.2 ส่วนของการจำเสียง

โปรมแกรมนี้เป็นการจดจำเสียงแบบที่เราต้องอัดไฟล์เสียงที่เราต้องการให้จดจำไปเองก่อน แล้วจึงค่อยเรียกโปรแกรมส่วนนี้มาใช้ ดังรูปที่ 3.18



รูปที่ 3.18 แสดงโปรแกรมของการจดจำเสียงแบบ Manual

โดยที่การใช้งานก็จะคล้ายกับโปรแกรมก่อนหน้านี้ เมื่อเราทำการเลือกไฟล์เสียงที่เราต้องการจะให้จดจำได้แล้ว เราก็จะทำการกดปุ่ม Recognizing เพื่อที่จะทำการจดจำเสียง

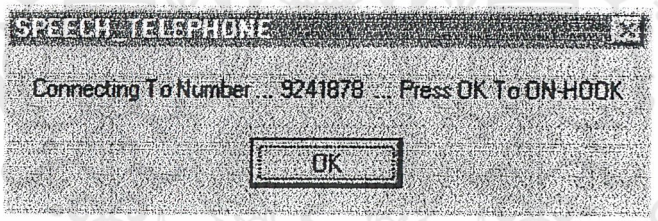


รูปที่ 3.19 แสดงผลลัพธ์ที่ได้จากการจำเสียง

3.2.2 ส่วนของการสั่งโทรออก

จากรูปที่ 3.7 ซึ่งเป็นหน้าหลักของโปรแกรมการต่อหมายเลขโทรศัพท์โดยใช้เสียง โปรแกรมนี้สามารถที่จะทำการต่อโทรศัพท์ไปยังปลายทางโดยใช้เสียงในการสั่งงาน โดยเมื่อเราต้องการโทรออกต้องกดปุ่มเลือกลักษณะการใช้งาน ซึ่งแบ่งออกเป็น 2 แบบคือ

1. พูดหมายเลขโทรศัพท์ เมื่อเรากดปุ่ม " หมายเลข " โปรแกรมจะทำการเริ่มอัดเสียงให้เราพูดหมายเลขปลายทางที่ต้องการ หลังจากนั้นโปรแกรมก็จะทำการวิเคราะห์เสียงว่าตรงกับโมเดลใดในฐานข้อมูลและทำการสั่งโมเด็มให้ต่อไปยังหมายเลขปลายทางที่ทำการรู้จำได้ โดยเมื่อโมเด็มทำการติดต่อแล้วจะเป็นดังรูปที่ 3.20 และเมื่อต้องการวางสายให้กดปุ่ม OK โมเด็มจะทำการวางสาย



รูปที่ 3.20 แสดงหน้าต่างเมื่อโมเด็มทำการต่อโทรศัพท์

2. พูดชื่อ เมื่อเรากดปุ่ม " ชื่อ " โปรแกรมจะทำการเริ่มอัดเสียง ให้เราพูดชื่อที่มีเก็บไว้ในฐานข้อมูล หลังจากนั้นโปรแกรมจะทำการวิเคราะห์เสียงว่าตรงกับโมเดลใดในฐานข้อมูลและนำไปเทียบกับหมายเลขโทรศัพท์ของชื่อนั้น แล้วจึงทำการสั่งโมเด็มให้ต่อไปยังหมายเลขปลายทางที่ตรงกับโมเดลเสียงที่วิเคราะห์ได้ ซึ่งจะได้ผลลัพธ์เหมือนดังข้อที่แล้ว

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลอง

4.1.1 การทดลองในส่วนของโปรแกรม Speech Recognition

การทดลองในส่วนของโปรแกรมการรู้จำเสียงซึ่งเขียนด้วยภาษา Visual C++ เวอร์ชั่น 6.0 สามารถจำแนกได้เป็น 2 ขั้นตอน คือ ขั้นตอนการสร้างแบบจำลองของเสียง(ขั้นตอนการเรียนรู้) และ ขั้นตอนการรู้จำเสียง (ขั้นตอนการวิเคราะห์)

4.1.1.1 ขั้นตอนการสร้างแบบจำลองของเสียง

เป็นขั้นตอนที่สร้างฐานข้อมูลของเสียงขึ้นมา ซึ่งเป็นเสียงที่ต้องการให้ระบบรับรู้ได้ โดยมีขั้นตอนดังนี้

1. เก็บเสียงตัวเลข 0-9 เพื่อนำไปสร้างแบบจำลองของเสียง โดยเก็บเป็นไฟล์ (*.wav) ซึ่งแบ่งเป็นกรณีต่างๆดังนี้

กรณีที่1 เสียงผู้ชาย 1 คน พูด 0-9 เสียงละ 5 ครั้ง รวมเป็น 50 เสียง

กรณีที่2 เสียงผู้หญิง 1 คน พูด 0-9 เสียงละ 5 ครั้ง รวมเป็น 50 เสียง

กรณีที่3 เสียงผู้ชาย 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวมเป็น 80 เสียง

กรณีที่4 เสียงผู้หญิง 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวมเป็น 80 เสียง

กรณีที่5 เสียงผู้ชาย 1 คน ผู้หญิง 1 คน พูด 0-9 เสียงละ 5 ครั้ง รวมเป็น 100 เสียง

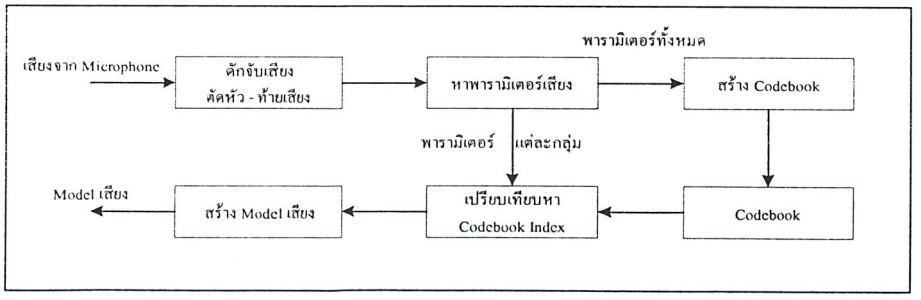
กรณีที่6 เสียงผู้ชาย 4 คน ผู้หญิง 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวมเป็น 160 เสียง

2. นำข้อมูลเสียงที่เก็บไว้ทั้งหมดในแต่ละกรณี มาหาค่าพารามิเตอร์ที่ใช้เป็นตัวแทนของแต่ละเสียง โดยวิธี LPC (Linear Predictive Coding) และเก็บไว้เป็นไฟล์ (*.lpc)

3. นำค่าพารามิเตอร์ของเสียงทั้งหมดในแต่ละกรณีมาทำการลดจำนวนข้อมูล โดยวิธีการ VQ (Vector Quantization) จะได้เวกเตอร์ตัวแทนของเสียงที่ระบบสามารถรับรู้ได้ทั้งหมดในแต่ละกรณีเรียกว่า โค้ดบุ๊ค (Codebook) โดยเก็บเป็นไฟล์ชื่อ codebook.vq

4. ทำการหา Code Book Index ของแต่ละกลุ่มเสียงโดยเก็บเป็นไฟล์ (*.idx) ซึ่งมีลักษณะเป็นตัวเลขที่บอกถึงความสัมพันธ์ของกลุ่มเสียงนั้นๆกับ Codebook

5. นำ Codebook Index ของแต่ละกลุ่มเสียง มาสร้างแบบจำลองของเสียง โดยวิธี Hidden Markov Model (HMM) จนครบทุกกลุ่มเสียง และพร้อมจะนำไปใช้ในขั้นตอนรู้จำเสียงต่อไป



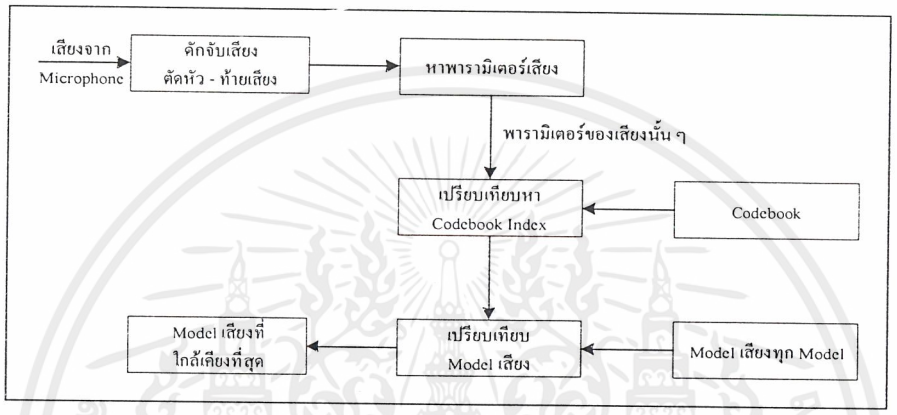
รูปที่ 4.1 แสดงขั้นตอนการสร้างแบบจำลองของเสียง

4.1.1.2 ขั้นตอนการรู้จำเสียง

เป็นขั้นตอนการวิเคราะห์เสียงจากที่อื่นๆว่ามีความเหมือนกับแบบจำลองของเสียงใดที่เก็บไว้ในขั้นตอนแรกมากที่สุด โดยโปรแกรมจะสามารถจำเสียงนั้นด้วยแบบจำลองที่ใกล้เคียงที่สุด

1. นำเสียงที่ต้องการทดสอบ(ครั้งละ 1 เสียง) มาผ่านขั้นตอน LPC จะได้ค่าพารามิเตอร์ของเสียงนั้น และนำค่าพารามิเตอร์มาเปรียบเทียบกับ Codebook ที่สร้างไว้ในขั้นตอนที่แล้ว จะได้ Codebook Index

2. นำ Codebook Index มาหาค่าความน่าจะเป็นกับทุกๆแบบจำลองของเสียงและดูว่าค่าความน่าจะเป็นเมื่อเปรียบเทียบกับแบบจำลองใดมีค่าสูงสุด ผลก็คือเสียงที่นำมาทดสอบจะถูกจำด้วยแบบจำลองนั้นนั่นเอง



รูปที่ 4.2 แสดงขั้นตอนการรู้จำเสียง

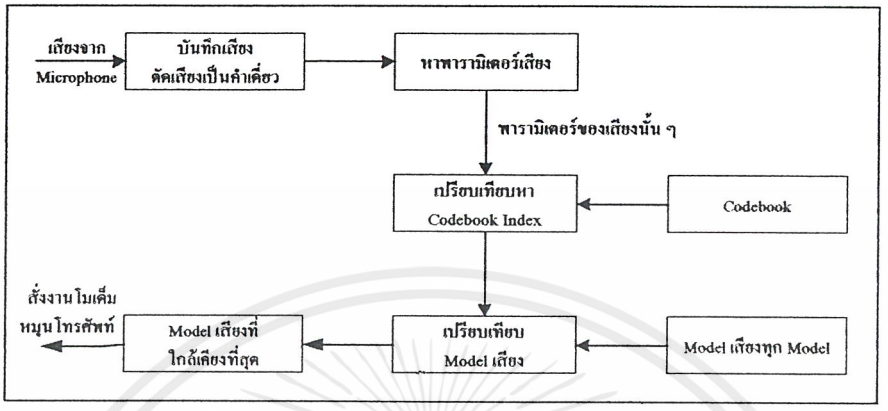
4.1.2 การทดลองในส่วนของโปรแกรม Speech Telephone

โปรแกรม Speech Telephone เป็นโปรแกรมที่ใช้สำหรับการต่อโทรศัพท์ออกโดยใช้เสียงพูด ซึ่งเป็นการนำทฤษฎีการรู้จำเสียงพูดมาประยุกต์ใช้งาน โดยมีลักษณะขึ้นกับผู้พูด กล่าวคือ ก่อนการใช้งาน ผู้พูดจะต้องสร้างฐานข้อมูลของเสียงของตนเองขึ้นมา โดยใช้โปรแกรม Speech Recognition ในที่นี้จะมีการใช้งานอยู่ 2 ลักษณะคือ

4.1.2.1 การต่อโทรศัพท์โดยพูดเป็นเลขหมายโทรศัพท์ปลายทาง มีขั้นตอนดังนี้

1. บันทึกเสียง 0-9 ของผู้พูด เลขละ 50 เสียง รวมเป็น 500 เสียง โดยบันทึกเป็นไฟล์ *.wav
2. เปิดโปรแกรม Speech Recognition โดยคลิกที่ปุ่ม MODEL ของโปรแกรมหลัก (โปรแกรม Speech Telephone) นำไฟล์ *.wav จากข้อ1 มาผ่านขั้นตอนต่างๆจนกระทั่งได้เป็นแบบจำลองของเสียง (ไฟล์ *.hmm)
3. ที่หน้าต่างโปรแกรมหลัก คลิกปุ่ม Number หลังจากนั้นพูดหมายเลขโทรศัพท์ปลายทางที่ต้องการติดต่อผ่านไมโครโฟน ทดลองพูดหมายเลขโทรศัพท์ 10 หมายเลข
4. เสียงของผู้พูดจะถูกบันทึกและผ่านขั้นตอนการตัดเสียงเพื่อแยกเสียงให้เป็นเสียงตัว

เลขเดี่ยวๆ แต่ละคำเดียวจะผ่านขั้นตอนการรู้จำเสียงดังที่กล่าวไว้ในหัวข้อ 4.1.1.2 ทำจนครบทุกๆคำเดี่ยว หลังจากที่โปรแกรมสามารถจดจำหมายเลขโทรศัพท์ที่ผู้พูดพูดเข้าไปได้ แล้วก็จะสั่งงานโมเด็มให้ต่อโทรศัพท์ออกไปยังเลขหมายปลายทางนั้น ตรวจสอบความถูกต้องของหมายเลขจาก Message Box ที่แสดงขึ้นมา รวมทั้งพิจารณา ความถูกต้องของจำนวนคำที่ตัดได้ด้วย บันทึกผลที่ได้



รูปที่ 4.3 แสดงขั้นตอนการต่อโทรศัพท์ออกโดยใช้เสียงพูดเลขหมายปลายทาง

4.1.2.1 การต่อโทรศัพท์โดยพูดเป็นชื่อของผู้รับปลายทาง มีขั้นตอนดังนี้

1. บันทึกชื่อของผู้รับปลายทางที่ต้องการติดต่อ ในที่นี้สมมุติให้มี 5 ชื่อดังนี้

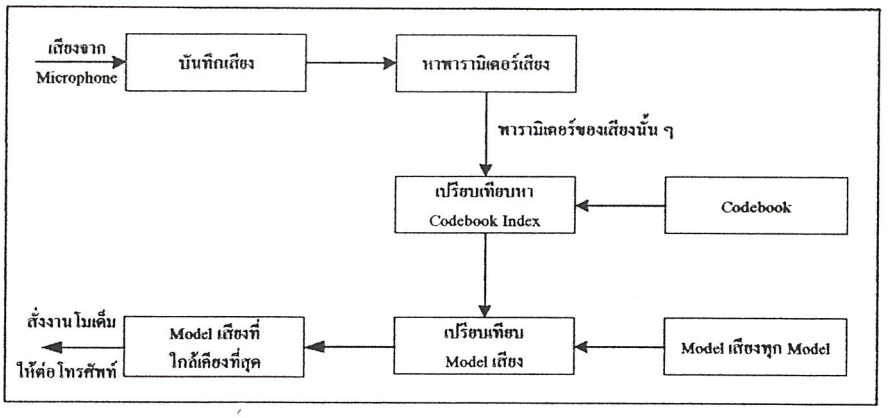
สา	50 เสียง ตรงกับหมายเลข	3360
แจ็ค	50 เสียง ตรงกับหมายเลข	3361
เอก	50 เสียง ตรงกับหมายเลข	3362
เป้	50 เสียง ตรงกับหมายเลข	3363
มล	50 เสียง ตรงกับหมายเลข	3364

2. คลิกปุ่ม MODEL ของโปรแกรมหลัก เพื่อสร้างแบบจำลองของเสียงในข้อ 1 และเก็บไว้เป็นฐานข้อมูล

3. คลิกปุ่ม Name ของโปรแกรมหลัก หลังจากนั้นพูดชื่อผู้รับที่มีในฐานข้อมูลทั้ง 5 ชื่อ

4. ชื่อที่พูดเข้าไปจะผ่านกระบวนการการรู้จำเสียงพูดโปรแกรมก็จำชื่อที่พูดด้วยแบบจำลองในฐานข้อมูลที่ใกล้เคียงที่สุด หลังจากนั้นก็จะสั่งงานให้ modem หมุนหมายเลขของผู้รับออกไป ตรวจสอบความถูกต้องจาก Message Box ที่แสดงขึ้นมา

5. ทำซ้ำข้อ 3 และ ข้อ 4 จนครบชื่อละ 5 ครั้ง บันทึกผลที่ได้



รูปที่ 4.4 แสดงขั้นตอนการต่อโทรศัพท์ออกโดยใช้เสียง พูดชื่อผู้รับปลายทาง

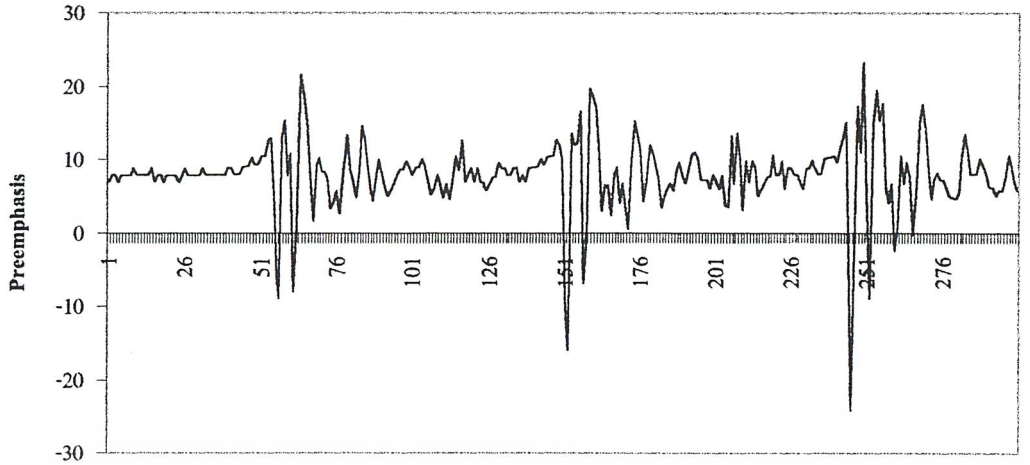
4.2 ผลการทดลอง

4.2.1 ผลการทดลองในส่วนโปรแกรม Speech Recognition

1. เมื่อทำการเก็บเสียงโดยใช้โปรแกรมบันทึกเสียงทั่วไป เสียงที่อัดนี้จะเป็น input เริ่มแรกของโปรแกรม สำหรับการบันทึกเสียงจะใช้โหมด Mono 8 bits 11.025 kHz และมีการสุ่มค่าตัวอย่าง (Sampling) เฟรมละ 300 ค่า ดังรูปที่ 4.5

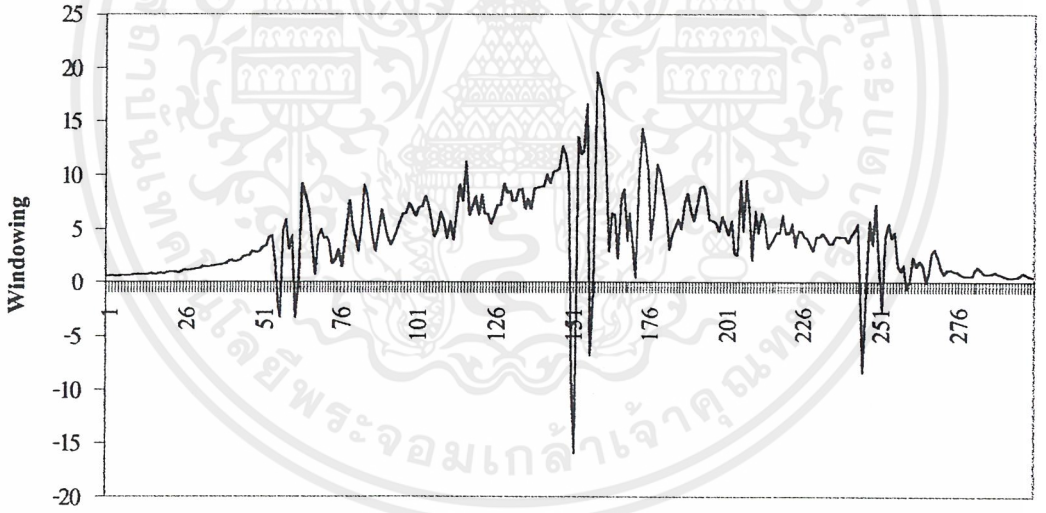
รูปที่ 4.5 แสดงสัญญาณเสียง 1 เฟรม ของเสียง 9 (เสียงผู้ชาย)

2. เมื่อนำเสียง input มาผ่านกระบวนการพรีเอมฟาซีส จะได้ output ออกมาดังรูปที่ 4.6



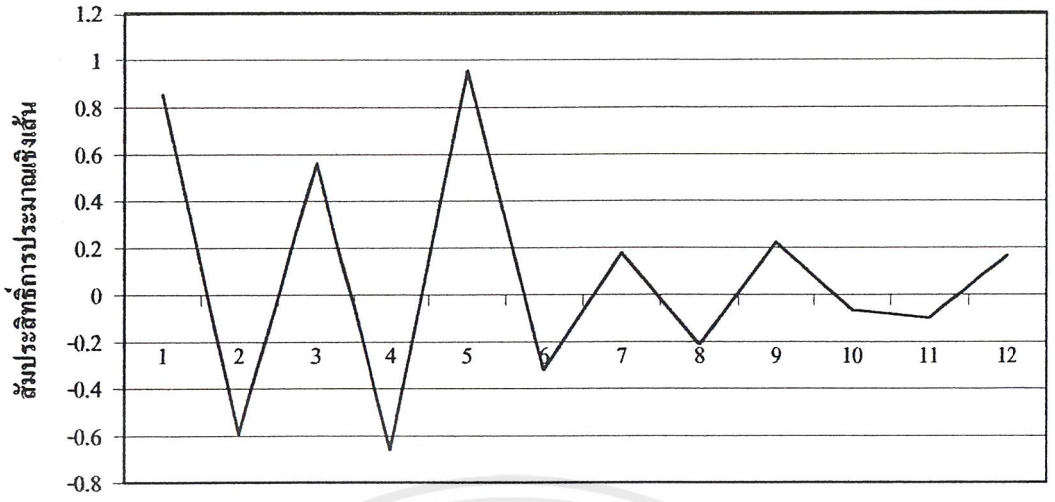
รูปที่ 4.6 แสดงสัญญาณที่ผ่านขั้นตอนพรีเอมฟาซีส 1 เฟรม ของเสียง 9(เสียงผู้ชาย)

3. เมื่อนำสัญญาณที่ผ่านการพรีเอมฟาซีส มาผ่านกระบวนการ windowing แบบ แฮมมิ่งจะได้ output ออกมา ดังรูปที่ 4.7



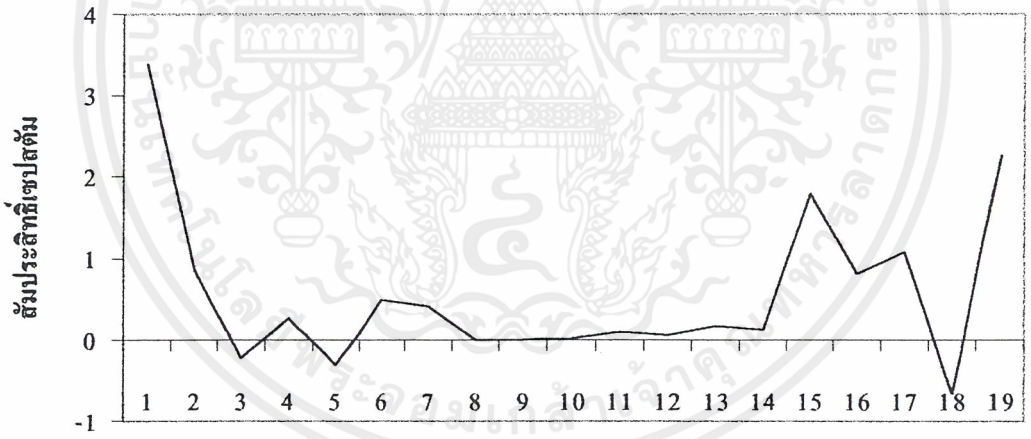
รูปที่ 4.7 แสดงสัญญาณที่ผ่านกระบวนการ window 1 เฟรม ของเสียง 9 (เสียงผู้ชาย)

4. สัญญาณที่ผ่านการ windowing จะเข้าสู่กระบวนการหาค่าสัมประสิทธิ์การประมาณเชิงเส้น โดย 1 เฟรมจะประกอบด้วยสัมประสิทธิ์ 12 ค่า ดังรูปที่ 4.8



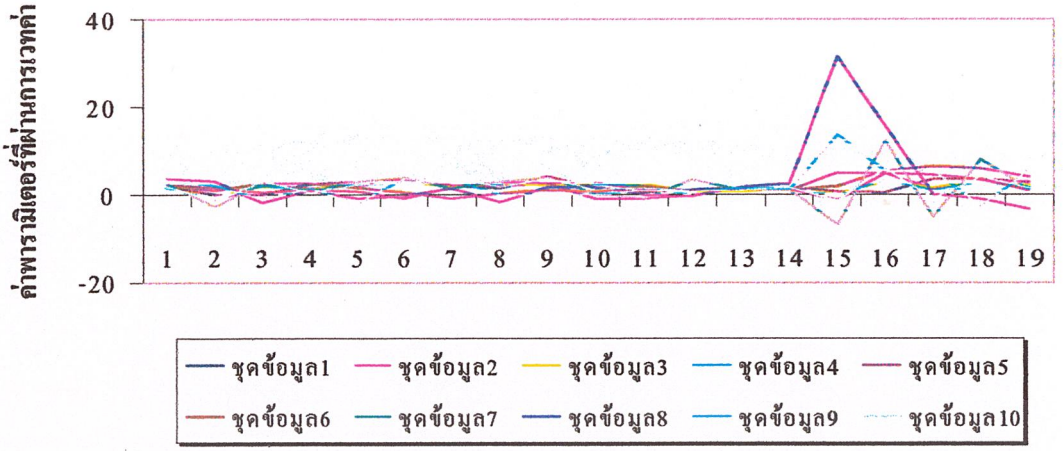
รูปที่ 4.8 แสดงสัมประสิทธิ์การประมาณเชิงเส้นทั้ง 12 ค่า ใน 1 เฟรม ของเสียง 9 (เสียงผู้ชาย)

5. เพื่อให้มีความน่าเชื่อถือมากขึ้นจึงมีการแปลงสัมประสิทธิ์การประมาณเชิงเส้นให้เป็นสัมประสิทธิ์เซปตัม โดยใน 1 เฟรมประกอบด้วยสัมประสิทธิ์เซปสตัม 19 ค่า ดังรูปที่ 4.9



รูปที่ 4.9 แสดงสัมประสิทธิ์เซปสตัมทั้ง 19 ค่า ใน 1 เฟรม ของเสียง 9 (เสียงผู้ชาย)

6. จากค่าสัมประสิทธิ์เซปสตัมที่ได้ นำมาเวทค่า ดังนั้น เอาท์พุทที่ได้จึงเป็นค่าพารามิเตอร์ที่เวทค่าแล้วโดยใน 1 เฟรมจะมี 19 ค่า เช่นกัน โดยค่าพารามิเตอร์นี้จะเป็นตัวแทนของเสียงเพื่อที่จะนำไปประมวลผลในขั้นต่อๆไป



รูปที่ 4.10 แสดงค่าพารามิเตอร์ที่ผ่านการเวทค่า 1 เฟรมของเสียง 0-9 (เสียงผู้ชาย)

7. นำพารามิเตอร์ของเสียงทั้งหมดที่จะสร้างแบบจำลองในแต่ละกรณี ผ่านกระบวนการจัดระดับเวกเตอร์

8. สร้างแบบจำลองของเสียงด้วยกระบวนการของ Hidden Markov Model (HMM) จนได้แบบจำลองทั้ง 10 ตัว ตั้งแต่ 0-9 ในที่นี้จะสร้างแบบจำลองทั้งหมด 6 กรณี เมื่อนำเสียงแต่ละกรณีมาสร้างแบบจำลองเรียบร้อยแล้วก็นำไปทดสอบต่อไป

9. ทำการทดสอบเสียงตามแบบจำลองที่สร้างไว้ 6 กรณี โดยแต่ละกรณีจะทำการทดสอบ 2 แบบ คือ ใช้เสียงจากผู้พูดที่เป็นต้นแบบเสียงในการสร้างแบบจำลอง และใช้เสียงจากบุคคลอื่น พูดตามเสียงต้นแบบ หลังจากนี้กระบวนการการรู้จำก็จะจำเสียงต่างๆด้วยโมเดลเสียงที่มีความใกล้เคียงกับเสียงมากที่สุดตามกรณีต่างๆดังนี้

กรณีที่1 แบบจำลองเสียงต้นแบบจากผู้ชายคนเดียว พูด 0-9 เสียงละ 5 ครั้ง รวมเป็น 50 เสียง

เสียงต้นแบบ	เสียงของผู้ชายคนเดิม(เสียงต้นแบบ)				
	พูดครั้งที่1	พูดครั้งที่2	พูดครั้งที่3	พูดครั้งที่4	พูดครั้งที่5
0	0	0	0	0	0
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	8	5	5	8	5
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	8	9	9	8	8
เปอร์เซนต์	80	100	100	80	90

ตารางที่ 4.1 ผลจากการทดสอบโดยให้ผู้พูดคนเดิมพูด 0-9 เสียงละ 5 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	2	0	6	6
1	1	1	4	6
2	0	2	2	3
3	2	3	5	0
4	4	4	4	4
5	9	9	1	9
6	6	0	6	2
7	7	7	4	7
8	8	9	1	4
9	9	9	9	9
เปอร์เซนต์	50	70	40	30

ตารางที่ 4.2 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2 คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ ครั้ง

กรณีที่2 แบบจำลองเสียงต้นแบบจากผู้หญิงคนเดียว พูด 0-9 เสียงละ 5 ครั้ง รวมเป็น 50 เสียง

เสียงต้นแบบ	เสียงของผู้หญิงคนเดิม(เสียงต้นแบบ)				
	พูดครั้งที่1	พูดครั้งที่2	พูดครั้งที่3	พูดครั้งที่4	พูดครั้งที่5
0	2	0	0	0	0
1	1	1	1	1	1
2	2	2	2	0	2
3	3	3	3	2	3
4	4	4	4	4	4
5	5	5	5	5	8
6	6	6	6	6	6
7	7	7	7	7	7
8	8	8	8	8	8
9	8	9	9	9	9
เปอร์เซนต์	80	100	100	80	90

ตารางที่ 4.3 ผลจากการทดสอบโดยให้ผู้พูดคนเดิมพูด 0-9 เสียงละ 5 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	0	0	0	0
1	5	3	0	1
2	0	0	0	2
3	2	2	3	3
4	4	1	0	4
5	3	9	3	5
6	6	6	6	6
7	7	0	4	7
8	8	8	8	5
9	3	3	3	9
เปอร์เซนต์	50	30	40	90

ตารางที่ 4.4 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ ครั้ง

กรณีที่ 3 แบบจำลองเสียงต้นแบบจากผู้ชาย 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวมเป็น 80 เสียง

เสียงต้นแบบ	ผู้ชายคนที่1		ผู้ชายคนที่2		ผู้ชายคนที่3		ผู้ชายคนที่4	
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	2
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	8	5	5	8	8	8	5
6	6	6	6	0	6	6	6	6
7	7	7	7	7	7	7	7	4
8	8	8	8	8	8	8	8	8
9	8	9	8	8	9	9	9	8
เปอร์เซนต์	90	90	90	80	90	90	90	70

ตารางที่ 4.5 ผลการทดสอบโดยให้ผู้ชายกลุ่มเดิม 4 คน พูด 0-9 เสียงละ 2 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	0	0	0	0
1	1	1	1	4
2	0	0	9	3
3	0	3	3	3
4	4	4	4	4
5	3	9	1	4
6	0	0	6	6
7	7	7	7	4
8	1	9	1	1
9	9	9	9	9
เปอร์เซนต์	50	60	70	40

ตารางที่ 4.6 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ1 ครั้ง

กรณีที่ 4 แบบจำลองเสียงต้นแบบจากผู้หญิง 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวมเป็น 80 เสียง

เสียงต้นแบบ	ผู้หญิงคนที่1		ผู้หญิงคนที่2		ผู้หญิงคนที่3		ผู้หญิงคนที่4	
	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2	ครั้งที่ 1	ครั้งที่ 2
0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1
2	0	2	2	2	2	2	2	2
3	3	3	0	0	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	8	8	5	5	5	8
6	6	6	6	6	6	6	6	6
7	7	4	7	7	7	4	7	7
8	8	8	8	8	8	8	8	8
9	9	9	9	9	8	8	9	9
เปอร์เซนต์	90	90	80	80	90	80	100	90

ตารางที่ 4.7 ผลการทดสอบโดยให้ผู้หญิงกลุ่มเดิม 4 คน พูด 0-9 เสียงละ 2 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	0	0	0	0
1	1	3	0	1
2	0	0	9	2
3	2	2	3	3
4	7	7	4	4
5	9	2	9	8
6	0	6	6	2
7	7	4	7	4
8	8	8	9	8
9	9	9	9	9
เปอร์เซนต์	50	40	60	60

ตารางที่ 4.8 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ1 ครั้ง

กรณีที่ 5 แบบจำลองเสียงจากผู้ชาย และ ผู้หญิง 1 คน พูด 0-9 เสียงละ 5 ครั้ง รวม 100 เสียง

เสียงต้นแบบ	เสียงผู้ชายคนเดิม					เสียงผู้หญิงคนเดิม				
	ครั้ง 1	ครั้ง 2	ครั้ง 3	ครั้ง 4	ครั้ง 5	ครั้ง 1	ครั้ง 2	ครั้ง 3	ครั้ง 4	ครั้ง 5
0	0	0	0	0	0	0	0	0	0	0
1	3	1	1	1	1	1	1	1	1	1
2	2	2	0	0	2	2	2	2	2	0
3	3	3	3	3	3	0	3	3	2	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	8	5	8	8	5	5	5
6	6	6	6	6	2	6	6	6	6	6
7	7	7	7	4	4	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	8	9	9	9	8	8	9	9
เปอร์เซนต์	90	100	80	70	80	80	80	90	90	90

ตารางที่ 4.9 ผลจากการทดสอบโดยให้ผู้ชายและผู้หญิงคนเดิมพูด 0-9 เสียงละ 5 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	0	0	0	0
1	1	1	2	4
2	2	0	9	3
3	3	3	3	3
4	4	4	4	4
5	3	9	3	5
6	0	6	2	6
7	7	7	7	7
8	8	5	8	5
9	8	9	9	8
เปอร์เซนต์	70	70	60	50

ตารางที่ 4.10 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ1 ครั้ง

กรณีที่ 6 แบบจำลองเสียงจากชาย และ หญิง 4 คน พูด 0-9 เสียงละ 2 ครั้ง รวม 160 เสียง

		0	1	2	3	4	5	6	7	8	9	%
ผู้ชาย	ครั้งที่1	0	1	0	3	4	5	6	7	8	9	90
คนที่ 1	ครั้งที่2	0	1	2	0	4	5	6	7	8	8	80
ผู้ชาย	ครั้งที่1	0	1	0	3	4	8	6	7	8	9	80
คนที่ 2	ครั้งที่2	0	1	0	3	4	8	2	7	8	9	70
ผู้ชาย	ครั้งที่1	0	1	2	3	4	5	6	7	8	8	90
คนที่ 3	ครั้งที่2	0	1	2	3	4	8	6	7	8	9	80
ผู้ชาย	ครั้งที่1	0	1	2	3	4	5	2	7	8	8	80
คนที่ 4	ครั้งที่2	2	1	2	3	4	5	2	4	8	9	70
ผู้หญิง	ครั้งที่1	0	1	0	3	7	5	6	7	8	9	80
คนที่ 1	ครั้งที่2	0	1	2	3	4	5	6	4	8	8	80
ผู้หญิง	ครั้งที่1	0	1	9	3	4	5	6	7	8	9	90
คนที่ 2	ครั้งที่2	0	1	2	3.	7	5	6	7	8	9	90
ผู้หญิง	ครั้งที่1	0	1	2	3	4	8	6	7	8	8	90
คนที่ 3	ครั้งที่2	0	1	3	3	4	5	6	7	8	9	90
ผู้หญิง	ครั้งที่1	0	5	2	3	4	5	6	7	8	9	90
คนที่ 4	ครั้งที่2	0	1	2	3	4	5	6	7	8	8	90

ตารางที่ 4.11 ผลจากการทดสอบโดยให้ผู้ชายและผู้หญิงกลุ่มเดิมพูด 0-9 เสียงละ 2 ครั้ง

เสียงต้นแบบ	เสียงผู้ชายคนอื่น		เสียงผู้หญิงคนอื่น	
	คนที่1	คนที่2	คนที่1	คนที่2
0	0	0	2	0
1	1	1	1	1
2	3	3	2	2
3	3	3	3	0
4	4	4	4	4
5	9	8	8	5
6	6	6	6	2
7	4	7	7	4
8	8	8	8	8
9	9	9	8	8
เปอร์เซนต์	70	80	70	60

ตารางที่ 4.12 ผลจากการทดสอบโดยให้ผู้ชายคนอื่น 2คน ผู้หญิงคนอื่น 2 คน พูด 0-9 เสียงละ1 ครั้ง

10. หลังจากที่ทำการทดสอบเสียงตามแบบจำลองครบทั้ง 6 กรณีแล้ว เราสามารถเปรียบเทียบความถูกต้องของการรู้จำเสียงเป็นเปอร์เซนต์ได้ดังนี้

ผู้ทดสอบ		เปอร์เซนต์ความถูกต้อง
กรณีที่1 เสียงจากผู้ชายคนเดียว	การทดสอบกับคนเดิม	90
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	60
	ผู้หญิง	35
	โดยรวม	47.5
กรณีที่2 เสียงจากผู้หญิงคนเดียว	การทดสอบกับคนเดิม	85
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	40
	ผู้หญิง	65
	โดยรวม	52.5
กรณีที่3 เสียงจากผู้ชาย 4 คน	การทดสอบกับคนเดิม	86.25
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	55
	ผู้หญิง	55
	โดยรวม	55
กรณีที่4 เสียงจากผู้หญิง 4 คน	การทดสอบกับคนเดิม	87.5
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	45
	ผู้หญิง	60
	โดยรวม	52.5
กรณีที่5 ผู้ชาย 1 คน ผู้หญิง 1 คน	การทดสอบกับคนเดิม	85
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	70
	ผู้หญิง	55
	โดยรวม	62.5
กรณีที่6 ผู้ชาย 4 คน ผู้หญิง 4 คน	การทดสอบกับคนเดิม	83.75
	การทดสอบกับกลุ่มทดสอบ	
	ผู้ชาย	75
	ผู้หญิง	65
	โดยรวม	70

ตารางที่ 4.13 เปรียบเทียบผลที่ได้จากการทดลองทั้ง 6 กรณี

4.2.2 ผลการทดลองในส่วนของโปรแกรม Speech Telephone

4.2.2.1 กรณีต่อโทรศัพท์โดยพูดหมายเลขปลายทาง

1. หมายเลขโทรศัพท์ที่พูดออกไปซึ่งมีหลายๆพยางค์ จะผ่านกระบวนการการตัดเสียงโดยวิธีการหาค่าขนาดกำลังสองเฉลี่ย เพื่อแยกเสียงออกเป็นคำเดี่ยวๆ

2. ผลการทดลองเมื่อพูดหมายเลขปลายทาง 10 เลขหมาย โดยตรวจสอบความถูกต้องของ แต่ละเลขหมายกับ Message Box ที่แสดงขึ้นมา แสดงดังตาราง

หมายเลขโทรศัพท์	0123456	หมายเลขที่จำได้	ครั้งที่1	0123486	จำนวนคำจริง	7	จำนวนคำที่ตัดได้	7
			ครั้งที่2	0103456		7		7
			ครั้งที่3	6123486		7		7
			ครั้งที่4	0123456		7		7
			ครั้งที่5	0123486		7		7
%ความถูกต้องของการจำเสียง			85.7		%การตัดเสียง			100
หมายเลขโทรศัพท์	3456789	หมายเลขที่จำได้	ครั้งที่1	3456788	จำนวนคำจริง	7	จำนวนคำที่ตัดได้	7
			ครั้งที่2	3486789		7		7
			ครั้งที่3	3456789		7		7
			ครั้งที่4	3486788		7		7
			ครั้งที่5	3456788		7		7
%ความถูกต้องของการจำเสียง			85.7		%การตัดเสียง			100
หมายเลขโทรศัพท์	9241878	หมายเลขที่จำได้	ครั้งที่1	9241878	จำนวนคำจริง	7	จำนวนคำที่ตัดได้	7
			ครั้งที่2	8247878		7		7
			ครั้งที่3	8247878		7		7
			ครั้งที่4	9241878		7		7
			ครั้งที่5	9241878		7		7
%ความถูกต้องของการจำเสียง			88.6		%การตัดเสียง			100
หมายเลขโทรศัพท์	8482679	หมายเลขที่จำได้	ครั้งที่1	8482678	จำนวนคำจริง	7	จำนวนคำที่ตัดได้	7
			ครั้งที่2	8482678		7		7
			ครั้งที่3	8482678		7		7
			ครั้งที่4	8482679		7		7
			ครั้งที่5	8482679		7		7
%ความถูกต้องของการจำเสียง			91.4		%การตัดเสียง			100
หมายเลขโทรศัพท์	7390213	หมายเลขที่จำได้	ครั้งที่1	7390273	จำนวนคำจริง	7	จำนวนคำที่ตัดได้	7
			ครั้งที่2	7380213		7		7
			ครั้งที่3	7390213		7		7
			ครั้งที่4	7390213		7		7
			ครั้งที่5	7380213		7		7
%ความถูกต้องของการจำเสียง			91.4		%การตัดเสียง			100
ค่าเฉลี่ยความถูกต้อง			88.56		ค่าเฉลี่ยความถูกต้อง			100

ตารางที่ 4.14 ผลการทดสอบเมื่อพูดเลขหมายปลายทาง 5 เลขหมายๆละ 5 ครั้ง

4.2.2.2 กรณีต่อโทรศัพท์โดยพูดชื่อผู้รับปลายทาง

ผลการทดลองเมื่อพูดชื่อผู้รับปลายทางทั้ง 5 ชื่อ ๆละ 5 ครั้ง โดยตรวจสอบความถูกต้องของ แต่ละชื่อกับ Message Box ที่แสดงขึ้นมา แสดงดังตาราง

ชื่อ	หมายเลขโทรศัพท์	เสียงที่จำได้					% ความถูกต้อง
		ครั้งที่1	ครั้งที่2	ครั้งที่3	ครั้งที่4	ครั้งที่5	
สา	3360	สา	สา	สา	สา	สา	100
แจ็ค	3361	แจ็ค	มล	แจ็ค	แจ็ค	แจ็ค	80
เอก	3362	เอก	เอก	เป้	แจ็ค	เอก	60
เป้	3363	มล	เป้	เป้	เป้	เป้	80
มล	3364	มล	มล	มล	มล	มล	100
					ค่าเฉลี่ย		84

ตารางที่ 4.15 ผลการทดสอบเมื่อพูดชื่อผู้รับปลายทาง 5 ชื่อๆละ 5 ครั้ง

บทที่ 5
บทสรุปและวิจารณ์

5.1 สรุปผลการทดลอง

ในส่วนของการวิเคราะห์สัญญาณเสียงเพื่อเข้ารหัสนั้นได้ใช้วิธีการประมาณเชิงเส้น (Linear Predictive Coding : LPC) เนื่องจากเป็นวิธีที่วิเคราะห์พารามิเตอร์ได้แม่นยำ และสามารถย่อข้อมูลได้อย่างมีประสิทธิภาพ ในการประมาณเชิงเส้น ซึ่งมีอยู่หลายวิธีโดยได้เลือกวิธีออโตคอร์รีเลชั่น เพราะเป็นวิธีที่มีการคำนวณที่ใช้สมการน้อยที่สุด

จากการทดลองในบทที่ 4 โดยใช้สัญญาณเสียงที่สุ่มด้วยความถี่ 11.025 กิโลเฮิร์ท ใช้การประมาณเชิงเส้นอันดับที่ 12 สรุปผลได้ดังนี้

1. ค่าสัมประสิทธิ์ของเสียงตัวเลขใด ๆ ที่วิเคราะห์ได้ใน 1 เฟรม ประกอบไปด้วย
   เกน 1 ค่า
   สัมประสิทธิ์ LPC 12 ค่า
   สัมประสิทธิ์เซปสตรัม 19 ค่า
   โดยใน 1 เสียง จำนวนเฟรมที่วิเคราะห์จะขึ้นอยู่กับสัญญาณเสียงที่พูด
2. ค่าสัมประสิทธิ์เซปสตรัมได้จากการปรับปรุงสัมประสิทธิ์ LPC โดยการใช้สมการเซปสตรัม เพื่อให้คงลักษณะเสียงได้มากขึ้น เนื่องจากสัมประสิทธิ์เซปสตรัมเป็นพารามิเตอร์ที่มีลักษณะน่าเชื่อถือได้ดีกว่าสัมประสิทธิ์ LPC และมีความสัมพันธ์ใกล้ชิดกับการรับรู้เสียงตามความรู้สึกของมนุษย์โดยแท้จริง และจะได้จำนวนสัมประสิทธิ์จากการวิเคราะห์สัญญาณเสียงมากกว่าจำนวนสัมประสิทธิ์ LPC ซึ่งจะทำให้วิเคราะห์สัญญาณเสียงได้ละเอียดมากยิ่งขึ้น
3. การพูดเสียงต่างกันก็จะได้สัมประสิทธิ์แตกต่างกัน
4. การพูดเสียงเดียวกันหลาย ๆ ครั้ง ก็จะได้สัมประสิทธิ์ต่างกันทุกชุด
5. ใน 1 เสียงจะวิเคราะห์ทีละเฟรม ซึ่งในแต่ละเฟรมก็จะมีสัมประสิทธิ์เซปสตรัม 1 ชุ ด คือ 19 ตัว ดังนั้นใน 1 เสียงก็จะมีสัมประสิทธิ์หลายชุดที่แตกต่างกัน
6. ในการพูดนั้น ถ้าพูดด้วยเสียงที่เป็นธรรมชาติมากที่สุดก็จะได้ค่าสัมประสิทธิ์ที่ถูกต้องมากยิ่งขึ้น
7. ในกรณีเสียงต้นแบบที่นำมาทำแบบจำลองมาจากหลายคนและมีทุกเพศทุกวัย จะทำให้แบบจำลองนี้สามารถครอบคลุมความแปรปรวนต่างๆ ได้ดี เมื่อนำมาทดสอบกับเสียงอื่น ๆ จะมีความถูกต้องค่อนข้างมาก
8. ในส่วนของการพูดต่อหมายเลขโทรศัพท์จะเป็นแบบขึ้นกับผู้พูด ซึ่งถ้าเป็นกรณีพูดหมายเลขโทรศัพท์ โปรแกรมสามารถตัดคำได้ถูกต้องเป็นส่วนใหญ่ ส่วนกรณีของการพูดชื่อโปรแกรมก็สามารถรู้จำเสียงพร้อมกับเปรียบเทียบกับโมเดลเสียงในฐานข้อมูลและสั่งโมเด็มให้โทรออกไปยังเลขหมายของเสียงนั้นได้

5.2 วิจารณ์

1. สัมประสิทธิ์ LPC ที่ได้จำเป็นจะต้องเปลี่ยนเป็นสัมประสิทธิ์เซปสตรัม เนื่องจากสัมประสิทธิ์ LPC จะมีความคลาดเคลื่อนมากกว่า
2. การออกเสียงแต่ละครั้งจะต้องให้เป็นธรรมชาติมากที่สุด มิฉะนั้นจะได้ค่าสัมประสิทธิ์ที่คลาดเคลื่อนจากความเป็นจริง
3. การออกเสียงแต่ละครั้งของคำหนึ่งคำจะไม่เหมือนเดิม อาจจะสั้นหรือยาวกว่าเดิม ทำให้ความยาวของสัญญาณเสียงและจำนวนเฟรมไม่เท่าเดิม และ codebook index ที่ได้ก็จะไม่เหมือนเดิม เมื่อนำไปผ่านกระบวนการทดสอบการรู้จำโดย Viterbi Algorithm จะทำให้การรู้จำผิดพลาดได้ นอกจากนี้ความผิดพลาดอาจเกิดขึ้นเนื่องจาก
   - ในขั้นตอนการจัดเก็บเสียง ทำการจัดเก็บเสียงที่ผิดปกติเพื่อนำไปสร้างแบบจำลอง ทำให้ได้แบบจำลองที่ไม่สมบูรณ์
   - ในกระบวนการตัดหัวท้ายของเสียงนั้นไม่สมบูรณ์เพียงพอ ยังมีการตัดคำเกินหรือขาดไป จึงควรมีอัลกอริทึมที่มีประสิทธิภาพในการตัดหัวท้ายเสียง จะทำให้ได้หัวเสียงและท้ายเสียงของคำนั้นอย่างสมบูรณ์
   - โค้ดบุคที่ได้ไม่มาตรฐานเพียงพอ ทำให้ได้โค้ดบุคอินเด็กซ์จากขั้นการควอนไตซ์ซึ่งต้องอ้างอิงกับโค้ดบุคออกมาผิดพลาด ทำให้ความน่าจะเป็นเปลี่ยนไป จึงทำให้การตัดสินใจในขั้นตอนการทดสอบการรู้จำผิดพลาดได้
4. ในขั้นตอนการสร้างโค้ดบุคและการสร้างแบบจำลองเสียงต้องทำการสุ่มค่าเริ่มต้นหลาย ๆ ครั้งเพื่อให้ได้ค่าที่เหมาะสมที่สุดจึงจะได้แบบจำลองที่ดีที่สุด
5. ในขั้นตอนการต่อโทรศัพท์นี้เป็นเพียงแนวทางที่จะนำไปสู่ความสะดวกในการใช้งานจริง การเขียนโปรแกรมสั่งงานให้โมเด็มหมุนหมายเลขโทรศัพท์ที่ตรงกับคำที่จำได้จึงยังต้องใช้ระบบแมนนวล (manual) อยู่บ้าง

แนวทางการศึกษาและพัฒนา

แนวทางที่ 1 : พัฒนาโปรแกรมให้มีความสามารถในการตรวจจับสัญญาณในสายโทรศัพท์

ในขณะนี้หลังจากที่โปรแกรมสั่งโมเด็มให้โทรศัพท์ออกไปยังปลายทางแล้ว ถ้าไม่มีข้อผิดพลาดเกิดขึ้นทางผู้รับจะรับสาย และสนทนากับผู้ส่งได้ตามปกติ แต่กรณีที่ สายไม่ว่าง (BUSY) หรือไม่พบสัญญาณให้หมุนบนสายโทรศัพท์ท้องถิ่น (NO DIALTONE) โปรแกรมไม่สามารถจะตรวจจับสัญญาณเหล่านี้ได้ ทางผู้ส่งจะรู้ได้โดยฟังสัญญาณ BUSY TONE หรือสัญญาณอื่นๆ จากหูฟังโทรศัพท์หรือลำโพงเท่านั้น ทางผู้จัดทำโครงงานจึงมีแนวความคิดในการพัฒนาโปรแกรมให้สามารถตรวจจับสัญญาณเหล่านี้และแสดงผลออกมาทางหน้าจอ หรือแสดงผลเป็นเสียงพูดให้ผู้ใช้สามารถรับรู้ได้ เพื่อเป็นการเพิ่มประสิทธิภาพโปรแกรมให้มีความน่าใช้มากขึ้น จากการศึกษาเกี่ยวกับโมเด็มพบว่าหลังจากที่โมเด็มได้รับคำสั่ง มันจะส่งรหัสผลลัพธ์กลับมา รหัสนี้อาจจะอยู่ในรูปข่าวสารหรือตัวเลข ดังตาราง

รหัสตัวเลข	รหัสข้อความ	ความหมาย
0	OK	คำสั่งถูกนำไปปฏิบัติ
1	CONNECT	เชื่อมต่อที่ 0-300 bps
2	RING	ตรวจพบสัญญาณกริ่ง
3	NO CARRIER	ไม่พบโมเด็มระยะไกล
4	ERROR	ความผิดพลาดในคำสั่ง
5	CONNECT 1200	เชื่อมต่อที่ 1200 bps
6	NO DIALTONE	ไม่พบสัญญาณให้หมุนบนสายโทรศัพท์ท้องถิ่น
7	BUSY	สายไม่ว่าง
8	NO ANSWER	ไม่มีการตอบสนองจากโมเด็มระยะไกล
10	CONNECT 2400	เชื่อมต่อที่ 2400 bps

ถ้าเรารู้ว่าโมเด็มเก็บผลลัพธ์เหล่านี้ไว้ที่ใด เราก็เขียนโปรแกรมให้โมเด็มส่งค่าเหล่านั้นมายัง DTE (ในที่นี้คือ Computer) เมื่อได้รับค่าผลลัพธ์แล้วก็จะเขียนโปรแกรมให้แสดงผลลัพธ์ออกมาทาง Message Box พร้อมๆกับแสดงผลทางเสียง

แนวทางที่2 : พัฒนาโปรแกรมให้ผู้ใช้สามารถติดต่อกับผู้รับโดยผ่านไมโครโฟนและลำโพง

ในการทำงานของโปรแกรม หลังจากที่สั่งให้โมเด็มหมุนไปยังเลขหมายปลายทางแล้วเมื่อทางด้านรับยกหูโทรศัพท์ขึ้น ถ้าผู้ส่งต้องการสนทนาจะต้องยกหูโทรศัพท์ที่ต่ออยู่กับโมเด็มด้วยเช่นกัน เมื่อพิจารณาแล้วอาจจะไม่เป็นการอำนวยความสะดวกมากนัก ดังนั้นผู้จัดทำโครงงานจึงมีแนวความคิดว่าผู้ใช้โปรแกรม น่าจะสนทนากับปลายทางผ่านทางไมโครโฟนและลำโพงที่ต่ออยู่กับ sound card แทนการยกหูโทรศัพท์ โดยจะต้องศึกษาหาความรู้เพิ่มเติมในส่วนของการเขียนโปรแกรมให้ sound card สามารถติดต่อกับ โมเด็มได้

แนวทางที่ 3 : เพิ่มเติม OPTION ต่างๆ ให้โปรแกรมอำนวยความสะดวกมากขึ้น

ฟังก์ชันเสริมหรือ Option ต่างๆ ที่น่าจะเพิ่มเติมเข้าไปเพื่อทำให้โปรแกรมมีความน่าใช้และอำนวยความสะดวกมากยิ่งขึ้น ได้แก่

1. ฟังก์ชัน REDIAL โดยจะทำเป็นปุ่มพิเศษขึ้นมา เมื่อผู้ใช้คลิกปุ่มนี้แล้วก็สามารถจะต่อโทรศัพท์ไปยังเลขหมายปลายทางล่าสุดได้อีกครั้ง โดยไม่ต้องผ่านกระบวนการตัดเสียงและกระบวนการรู้จำเสียงพูด ทำให้การทำงานของโปรแกรมมีความรวดเร็วขึ้น

2 ฟังก์ชัน เครื่องตอบรับโทรศัพท์อัตโนมัติ เมื่อมีการโทรเข้ามาโปรแกรมก็จะทำหน้าที่เป็นเครื่องตอบรับอัตโนมัติเมื่อไม่มีใครรับสายก็จะให้ผู้โทรเข้ามาสามารถฝากข้อความไว้ได้

3. ฟังก์ชัน การสอบถามเลขหมายโทรศัพท์ ในกรณีที่ผู้ใช้ลืมหมายเลขโทรศัพท์ใครก็สามารถโทรมาสอบถามเบอร์โทรศัพท์ได้ โดยพูดชื่อเจ้าของเลขหมาย หลังจากที่โปรแกรมได้รับเสียงแล้ว ก็จะนำเสียงนั้นมาผ่านกระบวนการการรู้จำเสียง เมื่อจำได้แล้วก็จะตรวจดูว่าตรงกับเลขหมายใด ก็จะตอบหมายเลขนั้นไปให้ผู้ใช้

แนวทางที่ 4: พัฒนาโปรแกรมที่ได้สร้างเป็นวงจร (Hardware) เพื่อเป็นแนวทางในการพัฒนาเพื่อสร้างเป็นเครื่องอำนวยความสะดวกทางโทรศัพท์โดยใช้เสียงต่อไป