บทที่ 3

การทดลอง

วิทยานิพนธ์นี้ได้พัฒนาโปรแกรมในส่วนต่าง ๆ ดังนี้

- ส่วนของ Linear Predictive Coding (LPC)
- ส่วนของ Vector Quantization (VQ)
- ส่วนของ Hidden Markov Model (HMM)
- ส่วนของ Viterbi Algorithm (VA)

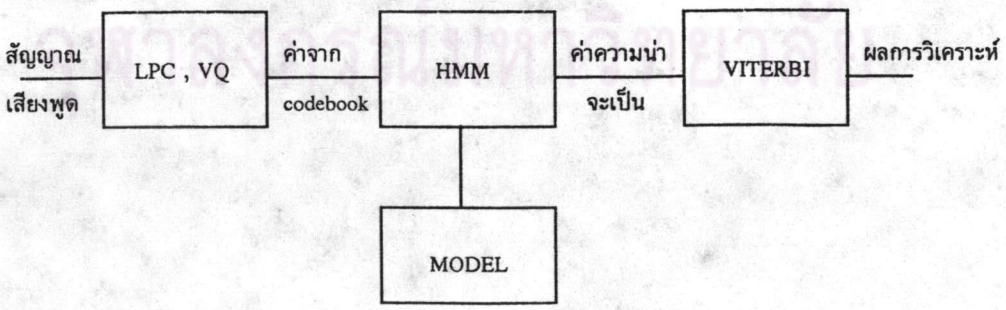
ได้นำโปรแกรมบางส่วน เช่น ส่วนของการปรับแต่งหัวท้ายคำจาก (ระพีพัฒน์ เพ็ญศิริ, 2537)มาใช้ปรับแต่งข้อมูลก่อนนำไปคำนวณ

<u>วิธีการดำเนินการวิจัย</u>

บันทึกเสียงพูดคำว่า 0-9 ของกลุ่มบุคคลที่ใช้เป็นเสียงต้นแบบและใช้ทดสอบเบื้องต้น คนละ 2 ครั้งจำนวน 45 คน เพื่อใช้ในการหาโมเดลของเสียง (ครั้งแรกใช้ในการหาโมเดล ครั้งที่ 2 ใช้ในการทดสอบเบื้องต้น) และนำบุคคลที่ไม่เคยนำมาเป็นตัวอย่างต้นแบบ มาบันทึกเสียงพูดคำว่า 0-9 คนละ 1 ครั้ง จำนวน 10 คน โดยข้อมูลทั้งหมดใช้ค่าสุ่มสัญญาณ 8 KHz ในการสุ่มสัญญาณเสียง จากนั้นนำข้อมูลที่บันทึกไว้ ปรับแต่งให้เหมาะสมโดยการตัดหัวท้ายคำพูดที่บันทึกไว้ออกไป เนื่องจากเป็นส่วนของสัญญาณรบกวน จากนั้นนำข้อมูลไปผ่านขั้นตอนต่าง ๆ ในการหาโมเดล โดยใช้ทฤษฎีที่ได้กล่าวไว้ในบทที่ 2 ดังรูป 2.1

ในโมเดลรูปแบบจะแบ่งเป็น 3 ส่วน

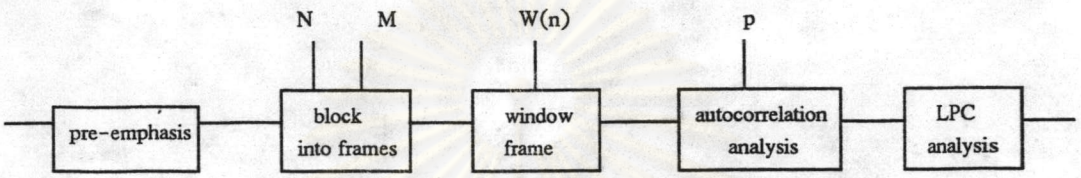
1. Feature measurement ใช้ LPC และ VQ ในการวิเคราะห์
2. Pattern similarity determination ใช้ HMM
3. Decision rule ใช้ Viterbi Algorithm



รูป 3.1 ขั้นตอนการหาโมเดลเสียง

3.1 การประมาณพันธะเชิงเส้น

เป็นการวิเคราะห์เพื่อหาค่าพารามิเตอร์ที่เหมาะสมให้กับโครงสร้างของระบบกำเนิดเสียง โดยแบ่งสัญญาณเสียงพูดที่จะวิเคราะห์ออกเป็นส่วนๆ แต่ละส่วนใช้ระยะเวลาช่วงสั้นๆ ประมาณ 15-20 มิลลิวินาที (msec) ซึ่งช่วงนี้สัญญาณเสียงพูด จะมีการเปลี่ยนแปลงคุณลักษณะอย่างช้าๆ จนอาจถือว่าระบบกำเนิดเสียงมีคุณลักษณะไม่เปลี่ยนแปลง (stationary) มีขั้นตอนการวิเคราะห์ดังรูป 3.2 (Rabiner and E. Levinson, 1981)



รูป 3.2 ขั้นตอนการทำงานของ LPC

3.1.1 Preemphasized ใช้วงจรกรองแบบดิจิตอลแบบ first order

$$H(z) = 1 - az^{-1}, \quad a = 0.95 \tag{3.1}$$

$$\tilde{s}(n) = s(n) - as(n-1) \tag{3.2}$$

3.1.2 การแบ่งช่วงเวลาในการพิจารณา

จะพิจารณาในช่วงเวลา 20 ms

$$x_l(n) = \tilde{s}(Ml+n), \quad n = 0, 1, \ldots, N-1$$
$$l = 0, 1, \ldots, L-1 \tag{3.3}$$

3.1.3 window

ในขั้นตอนนี้คือการนำข้อมูลมาผ่านฟังก์ชันที่กำหนดโดยใช้ Hamming window

$$\tilde{x}_l(n) = x_l(n) \cdot w(n) \tag{3.4}$$

$$w(n) = 0.54 - 0.46\cos(2\pi n/N-1) \tag{3.5}$$

3.1.4 autocorrealtion analysis

$$R_l(m) = \sum_{n=0}^{N-1-|M|} \tilde{x}_l(n)\tilde{x}_l(n+m), \quad m = 0, 1, \ldots, p \tag{3.6}$$

โดย order ที่ใช้พิจารณาอยู่ระหว่าง 8-12 แต่ใช้ค่า 10 (ภาคผนวก ก.)

$$X(l) = \{R_l(0), R_l(1), \ldots, R_l(p)\} \tag{3.7}$$

3.1.5 ขั้นตอนการหาค่าสัมประสิทธิ์ PARCOR ด้วยวิธีอัตสัมพันธ์ (สุเธียร เกียรติสุนทร, 2525)

เมื่อ m = 0 จะกำหนดค่าเริ่มต้น

$$a_{00} = 1$$

$$b_{01} = 1$$

$$\alpha_0 = r_0$$

$$\beta_0 = r_0 \qquad (3.8)$$

เมื่อ m = 1

$$k_1 = \frac{-r(1)}{\beta_0}$$

$$A_1(z) = 1+k_1 z^{-1}$$

$$B_1(z) = k_1 z^{-1} + z^{-2}$$

$$\alpha_1 = \alpha_0 + k_1^2 \beta_0$$

$$\beta_1 = \alpha_1 \qquad (3.9)$$

เมื่อ m = 2, 3, ... , M

$$\alpha_m k_m = \frac{-1}{\alpha_{m-1}} \sum_{i=0}^{m-1} r(m-i) a_{m-1\ i}$$

$$a_{m0} = 1$$

$$a_{mi} = a_{m-1\ i} + k_m b_{m-1\ i} \quad \text{เมื่อ } i=1, 2, \ldots, m-1$$

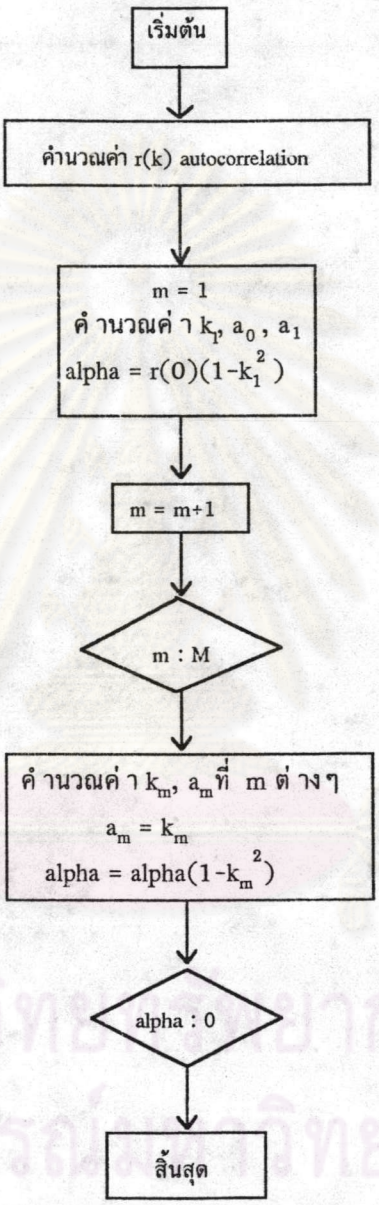
$$a_{mn} = k_m$$

$$b_{mi} = a_{m\ m+1-i} \quad \text{เมื่อ } i=1, 2, .., m+1$$

$$\alpha_m = \alpha_{m-1}\left(1+k_m^2\right)$$

$$\beta_m = \alpha_m \qquad (3.10)$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



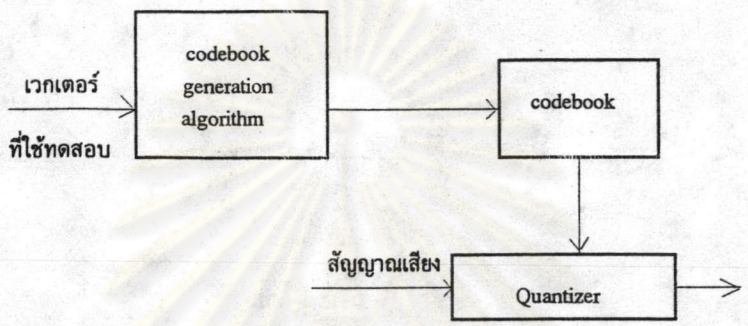
รูป 3.3 แสดงขั้นตอนการประมาณพันธะเชิงเส้นด้วยวิธีอัตสัมพันธ์

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

3.2 Vector Quantization

รูป 3.4 แสดงถึงขั้นตอนในการหา codebook และการทดลอง โดยแบ่งเป็น 2 ขั้นตอน (Furui, 1989)

ก. หา codebook ใช้ K-means algorithm โดยมีการสุ่มค่าเริ่มต้น ใช้ขนาด codebook 64 (ภาคผนวก ก.)



รูป 3.4 ขั้นตอนการทำงานของ VQ

ข. ในการทดสอบ ใช้การหาความคลาดเคลื่อนโดย square-error

$$d(x,y) = \|x-y\|^2$$

$$= \sum_{i=0}^{k-1} (x_i - y_i)^2 \qquad (3.11)$$

โดยการหาความคลาดเคลื่อนแบบ square-error ถูกใช้ทั้งตอนหา codebook และตอนทดสอบ

3.2.1 การออกแบบ ในการออกแบบ codebook ถ้าเวกเตอร์ x มี k มิติ ถูกเปรียบเทียบกับ codebook ขนาด K ค่า ขอบเขตที่เวกเตอร์ x จะอยู่ได้มี K ขอบเขต $\{C_i\}$ $(1 \le i \le K)$ และถูกแทนด้วย y_i ที่เป็นเวกเตอร์ code ของแต่ละเซล C_i โดยเขียนแทนได้ว่า

$$q(x) = y_i \qquad (3.12)$$

เมื่อ y คือค่าควอนไตซ์ของ x การหาความคลาดเคลื่อนคือ ระยะห่างระหว่างเวกเตอร์ x และ y เขียน โดย $d(x,y)$ และสามารถหาความคลาดเคลื่อนรวมได้คือ

$$D = \lim_{M \to \infty} \frac{1}{M} \sum_{n=1}^{M} d[x(n), y(n)] \qquad (3.13)$$

เราจะบอกว่า codebook ที่ได้เหมาะสมขึ้นกับความคลาดเคลื่อนรวมต้องต่ำที่สุด การที่ความคลาดเคลื่อนรวมต่ำเนื่องจาก 2 ปัจจัย

ก. การใช้ nearest-neighbour

$$q(x) = y_i, d(x, y_i) \le d(x, y_j) \quad j \ne i, 1 \le j \le K \quad (3.14)$$

ข. ค่าเวกเตอร์ code y_i

y_i เป็นค่ากลางของ C_i หาโดยวิธีหาค่ากลาง ซึ่งขึ้นกับการวัดความคลาดเคลื่อน

3.2.2 Lloyd's Algorithm (K-means Algorithm)

เป็นวิธีหา codebook แบบทำซ้ำ โดยพยายามแบ่งเวกเตอร์ {x(n)}ให้อยู่ในเซล $\{C_i\}$ จำนวน K เซล

ก. เริ่มต้น

เมื่อ m=0 (m คือตัวบอกจำนวนการทำซ้ำ) เลือกค่าเริ่มต้นเวกเตอร์ code $\{y_i(0)\}$, $(1 \le i \le K)$

ข. Classification

ทำการแบ่งเวกเตอร์ $\{x(n)\}(1 \le n \le M)$ ไปตามเซล $\{C_i(m)\}$ โดยใช้ทฤษฎี nearest-neighbour

$$x \in C_i(m)$$

$$\text{เมื่อ} \quad d[x, y_i(m)] \le d[x, y_j(m)], i \ne j \quad (3.15)$$

ค. ทำการปรับเวกเตอร์ code

เพิ่มค่า m ← m+1 ปรับเวกเตอร์ code ทุกๆ รอบการทำซ้ำ โดยคำนวณค่ากลางใหม่

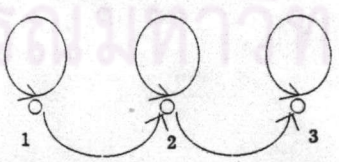
$$y_i(m) = centroid(C_i(m)) \quad , 1 \le i \le K \quad (3.16)$$

ง. เงื่อนไข

นำค่าความคลาดเคลื่อนรวม $D(m)$ ที่คำนวณได้เปรียบเทียบกับ $D(m-1)$ ถ้าต่ำกว่า 0.00001 (threshold) จะหยุดการทำงาน (เนื่องจากค่า 0.00001 ให้ผลความคลาดเคลื่อนของ codebook กับข้อมูลที่นำมาเปรียบเทียบน้อย)

3.3 Hidden Markov Model (HMM)

ในการเริ่มต้นต้องมีการกำหนดค่าเริ่มต้นบางค่า เช่น จำนวน state จำนวน codebook และโครงสร้าง ของโมเดล จากรูป 3.5 เป็นโมเดลที่กำหนดขึ้น (ภาคผนวก ก.)



รูป 3.5 โครงสร้างของ HMM

3.3.1 เงื่อนไขในการหาโมเดล เมื่อกำหนดโครงสร้างของโมเดลเป็นดังรูป 3.5 จะเห็นว่าเป็นโครงสร้างแบบ constrain serial แบบ ซ้ายไปขวา มีความหมายว่าเมื่อออกจาก state ใดๆ จะไม่ย้อนกลับไปที่ state นั้นๆ อีก โครงสร้างเช่นนี้เพียงพอ ต่อการวิเคราะห์ (ภาคผนวก ก.)

ก. ค่า b_{ij} จะต้องไม่เป็น 0 เมื่อคำนวณ b_{ij} ได้ต่ำกว่าค่าที่กำหนดขึ้น จะทำการปรับ b_{ij} ตัวนั้นให้เท่ากับค่าที่กำหนดขึ้น ค่าที่ใช้อยู่ระหว่าง 10^{-3}-10^{-10}(ในที่นี้ใช้ 10^{-7})(Rabiner, Levinson and Sondhi, 1983)

$$\sum_{j=1}^{M} b_{ij} = 1 \quad \text{เมื่อ } i = 1, 2, \ldots, N \tag{3.18}$$

ข. ค่า $a_{ij} = 0$ เมื่อ $i > j$

$$\sum_{j=1}^{N} a_{ij} = 1 \quad \text{เมื่อ } i = 1, 2, \ldots, N \tag{3.19}$$

ค. ค่า $\pi_1 = 1$, π_i ที่ state อื่นๆ เป็น 0 หมด

$$\sum_{i=1}^{N} \pi_i = 1 \tag{3.20}$$

โครงสร้างเช่นนี้จะเริ่มต้นที่ state ที่ 1 และสิ้นสุดที่ state สุดท้าย

3.3.2 ขบวนการคำนวณ กำหนดค่า π,A,B เริ่มต้น เป็นไปตามเงื่อนไขที่กล่าวมาข้างต้น

3.3.2.1 Forward-Backward procedure

forward variable $\alpha_t(i)$

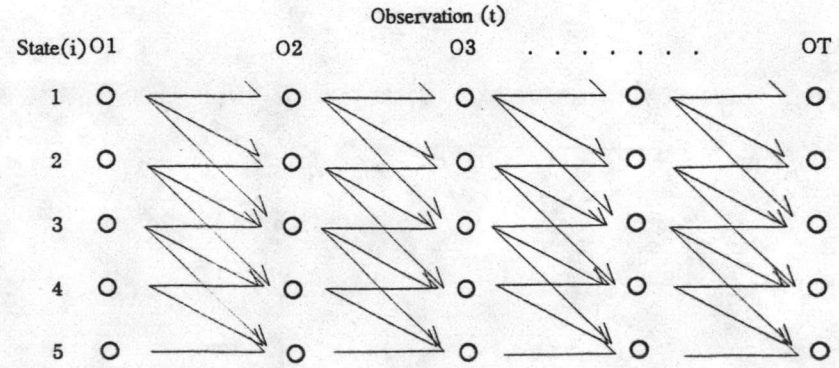
$$\alpha_t(i) = prob(O_1, O_2, \ldots, O_t, i_t = q_i / \lambda) \tag{3.21}$$

ขั้นตอน 1 $$\alpha_1(i) = \pi_i b_i(O_1) \qquad \text{เมื่อ } 1 \le i \le N \tag{3.22}$$

ขั้นตอน 2 t=1, 2, ..., T-1 $\quad 1 \le i \le N$

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}) \tag{3.23}$$

จากสมการ (3.23) สามารถอธิบายโดย lattice diagram ดังรูป 3.6



รูป 3.6 Lattice diagram ของ HMM

backward variable $\beta_t(i)$

$$\beta_t(i) = prob(O_{t+1}, O_{t+2}, \ldots, O_T / i_t = q_i, \lambda) \qquad (3.24)$$

ขั้นตอน 1 $\beta_T(i) = 1$ เมื่อ $1 \le i \le N$ (3.25)

ขั้นตอน 2 t=T-1, T-2, ... , 1 $1 \le i \le N$

$$\beta_t(i) = \sum_{j=1}^{N} \beta_{t+1}(j) a_{ij} b_j(O_{t+1}) \qquad (3.26)$$

3.3.2.2 Baum-Welch algorithm

เมื่อคำนวณ forward-backward เรียบร้อยจะนำ $\alpha_t(i)$, $\beta_t(i)$ ที่ได้มาประมาณค่า A,B ใหม่ โดยใช้สมการ

$$\gamma_{ij} = \frac{1}{p} \sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j) \qquad (3.27)$$

$$\gamma_i = \sum_{j=1}^{N} \gamma_{ij} = \frac{1}{p} \sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i) \qquad (3.28)$$

จากค่า γ_{ij} / γ_i เราสามารถประมาณค่า $\bar{a}_{ij}$ ได้

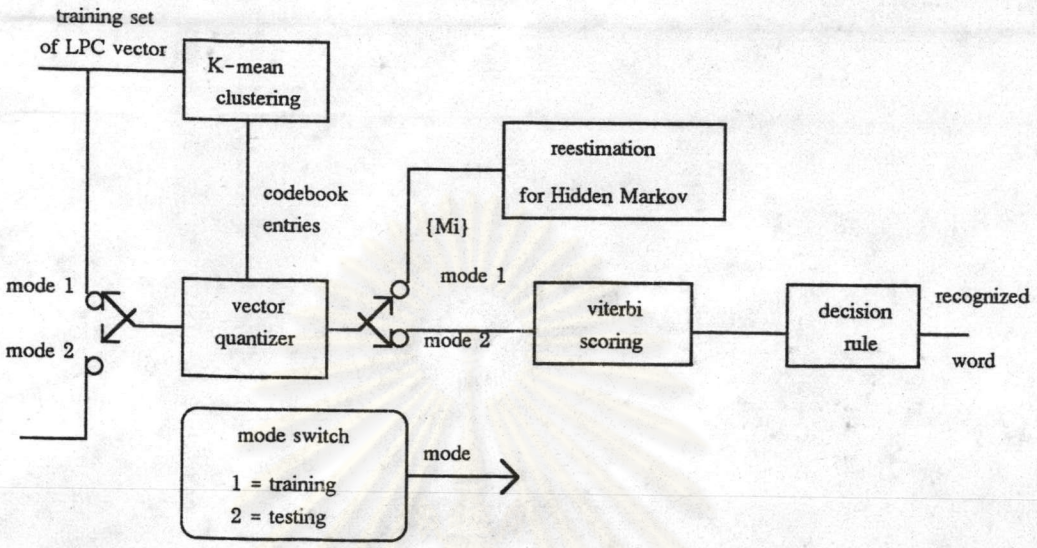
$$\bar{a}_{ij} = \frac{\gamma_{ij}}{\gamma_i} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \alpha_t(i) \beta_t(i)} \qquad (3.29)$$

$$\bar{b}_{jk} = \frac{\sum_{t \ni O_t = v_k}^{T} \alpha_t(j) \beta_t(j)}{\sum_{t=1}^{T} \alpha_t(j) \beta_t(j)} \qquad (3.30)$$

$$\pi_1 = 1 \qquad \text{ที่ state อื่น ๆ จะเป็น 0} \qquad (3.31)$$

เมื่อประมาณค่า $\bar{a}_{ij}$ และ $\bar{b}_{ij}$ เรียบร้อยทำการหา $\alpha_t(i)$, $\beta_t(i)$ ใหม่ เพื่อทำการประมาณ $\bar{a}_{ij}$ และ $\bar{b}_{ij}$ ใหม่ จนกว่าจะครบจำนวนครั้งที่กำหนดไว้ (ในวิทยานิพนธ์ใช้ 50 ครั้ง เนื่องจากหลังจาก 50 ครั้ง จะมีการเปลี่ยนแปลงของพารามิเตอร์ค่อนข้างช้า)

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย



รูป 3.7 ขั้นตอนในการหาโมเดลและทดสอบ

3.4 Viterbi Algorithm

การหาระยะทางที่สั้นที่สุดจากเส้นทางที่เป็นไปได้ทั้งหมด

ขั้นตอน 1: การหาค่าเริ่มต้น

$$\delta_1(i) = \pi_i b_i(O_1) \qquad \text{เมื่อ } 1 \le i \le N \tag{3.32}$$

$$\psi_1(i) = 0 \tag{3.33}$$

ขั้นตอน 2: การวนหาค่า เมื่อ $2 \le t \le T$, $1 \le i \le N$

$$\delta_t(i) = \max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(O_t) \tag{3.34}$$

$$\psi_t(j) = \arg\max_{1 \le i \le N} \left[\delta_{t-1}(i) a_{ij}\right] \tag{3.35}$$

ขั้นตอน 3: การหาค่าสิ้นสุด

$$P^* = \max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{3.36}$$

$$i_T^* = \arg\max_{1 \le i \le N} \left[\delta_T(i)\right] \tag{3.37}$$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

ขั้นตอน 4: backtracking

t=T-1, T-2, ... , 1

$$i_t^* = \psi_{t+1}(i_{t+1}^*) \qquad (3.38)$$

3.5 ขั้นตอนการทดลอง

ขั้นตอน 1

1. เมื่อผ่านขั้นตอนการตัดหัวท้ายของเสียงแล้ว จะผ่านมาค่าสัมประสิทธิ์การสะท้อน ตามสมการในหัวข้อ3.1 ข้อมูลที่ออกมาจะเป็นข้อมูลที่แบ่งเป็นเฟรม ในแต่ละเฟรมจะประกอบด้วยตัวเลข 10 ค่าที่แทน เอกลักษณ์ของการเกิดสัญญาณในแต่ละเฟรม

2. เมื่อผ่าน LPC เรียบร้อยจะทำการสุ่มค่าเริ่มต้นเพื่อใช้ในการหา codebook

3. นำข้อมูลทั้งหมดมาหา codebook ที่เหมาะสมโดยวิธีจาก 3.2

4. นำข้อมูลดังกล่าวเปรียบเทียบกับ codebook ว่า codebook ใดจะเป็นตัวแทนของข้อมูล

5. ทำการสุ่มค่าเริ่มต้น $\overline{a}_{ij}$ และ $\overline{b}_{ij}$

6. ทำการหาโมเดลของเสียง โดยหา 1 โมเดลต่อ 1 เสียง จากบุคคล 45 คน โดยใช้ ขั้นตอนจาก 3.3

ขั้นตอน 2 ใช้ในการวิเคราะห์

1. จาก unknown word ที่ผ่านการตัดหัวท้ายแล้ว จะถูกนำมาหาค่าสัมประสิทธิ์การสะท้อน 10 ค่าในแต่ละเฟรม

2. นำค่าสัมประสิทธิ์การสะท้อนมาหา code เวกเตอร์ที่เหมาะสม

3. นำ unknown ที่แปลงเป็น code เวกเตอร์เรียบร้อยแล้ว มาหาความน่าจะเป็นที่โมเดลต่าง ๆ จะให้กำเนิดค่าตาม unknown โดยใช้วิธีการใน 3.4

4. ตัดสินใจว่าเป็นโมเดลใด โดยเลือกค่าความน่าจะเป็นที่มากที่สุด

$$P_i = prob(O / M_i) \qquad (3.39)$$

โดย $1 \le i \le V$ (จำนวนคำ) เลือก i ที่ $P_i \ge P_j, 1 \le j \le V$

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย