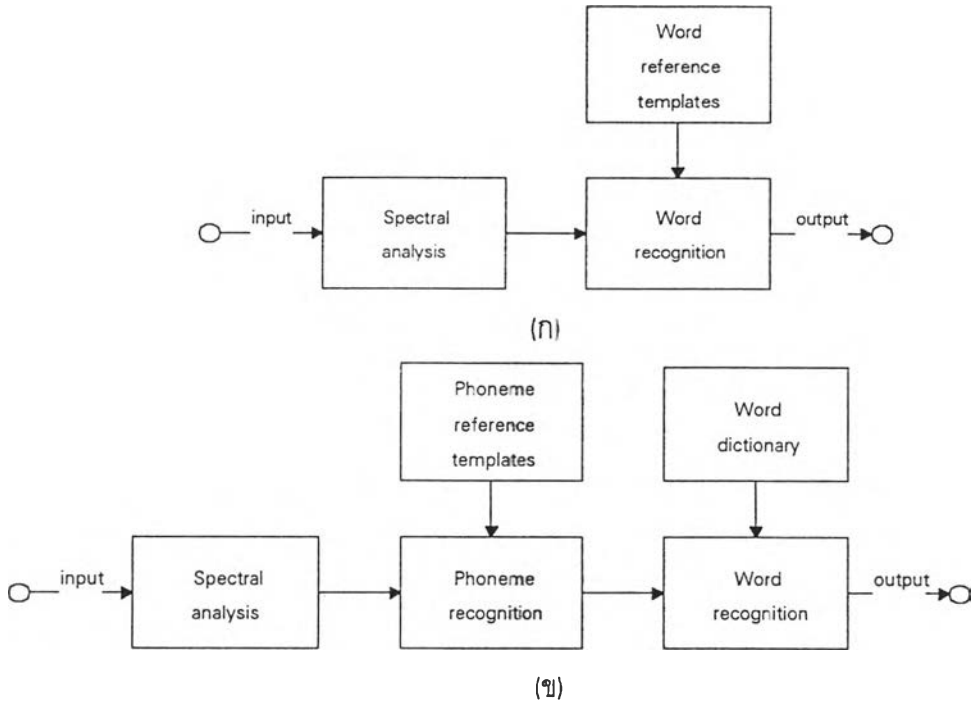
บทที่ 3

การรู้จำเสียงพูด

การรู้จำเสียงพูดนั้นจะเกี่ยวข้องกับการตัดสินใจระหว่าง แบบทดสอบ (test pattern) กับแบบอ้างอิง (reference pattern) หรือที่เรียกว่า template ซึ่งจะเป็นการวิเคราะห์ถึงความสัมพันธ์ของตัวแปร (parameter) ที่ใช้เป็นรูปแบบทั้งสอง เพื่อที่จะระบุว่าแบบทดสอบที่นำมาทดสอบนั้นมีความสัมพันธ์กับแบบอ้างอิงใดมากที่สุด จากนั้นจะนำไปสู่การตัดสินใจต่อไป

3.1 โครงสร้างของระบบการรู้จำ

ในระบบการรู้จำที่เป็นแบบ isolated word recognition นั้นจะสามารถแบ่งออกได้เป็น 2 รูปแบบ ดังแสดงในรูปที่ 3.1.1



รูปที่ 3.1.1 โครงสร้างของระบบการรู้จำ (Furui, 1989)

(ก) word-based recognition

(ข) phoneme-based recognition

จากรูปแบบโครงสร้างของการรู้จำ isolated word recognition ที่แสดงนั้นจะเห็นได้ว่าจะมีส่วนที่แตกต่างกันอยู่ที่การเลือกใช้ parameter ที่จะนำมาใช้ในคำ กล่าวคือ parameter ที่ใช้นี้จะแทนเป็นคำหรือว่าเป็นหน่วยเสียง โดยที่การเลือกใช้ template ที่เป็นหน่วยเสียงนั้นจะมีขนาดที่สั้นกว่าการเลือกใช้ template แบบคำ แต่จะมีความยุ่งยากกว่าเพราะจะต้องมีส่วนของ word dictionary เข้ามาร่วมด้วย ซึ่งได้ทำการเปรียบเทียบดังในตารางที่ 3.1.1 ทั้งนี้วิทยานิพนธ์นี้เลือกใช้โครงสร้างของระบบการรู้จำบนพื้นฐานของคำ เนื่องจากกำหนดการรู้จำกับเสียงตัวเลข ซึ่งเป็นคำสั้น ๆ และมีจำนวนคำไม่มาก

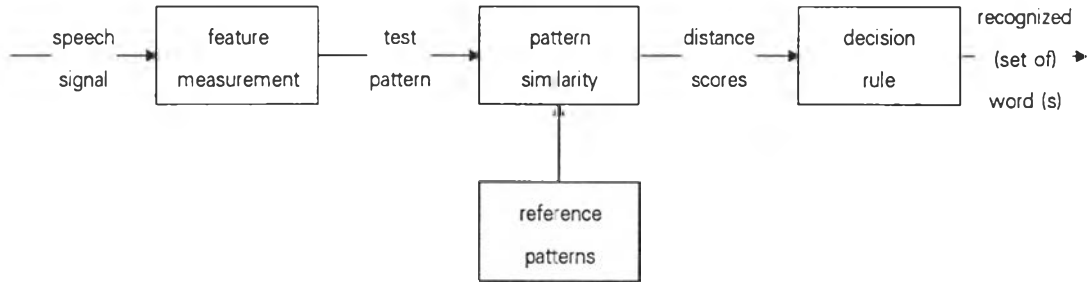
ในรูปที่ 3.1.2 จะแสดงถึงรูปแบบของ isolated word speech recognition system จะประกอบด้วยส่วนหลัก ๆ 3 ขั้นตอน (Rabiner and Levinson, 1981) คือ feature measurement, pattern similarity determination และ decision rule

ตารางที่ 3.1.1 แสดงการเปรียบเทียบข้อดี ข้อเสียของการเลือกใช้ template (Furui, 1989; ไพศาส ธรรมโพธิทอง, 2533)

ข้อดี	ข้อเสีย
<u>คำ</u> 1. เหมาะสำหรับคำสั้น ๆ 2. ไม่มีผลกระทบอันเกิดจากฐานเสียง 3. ใช้งานง่ายและใช้ได้กับทุกภาษา	1. template มีขนาดยาว 2. กรณีที่มีจำนวนของ template มากจะเสียเวลาในการคำนวณ
<u>หน่วยเสียง</u> 1. template มีขนาดสั้น 2. ไม่ขึ้นกับความยาวของคำที่ใช้และขนาดของความจำที่ใช้ในการเก็บ template 3. ใช้เวลาในการคำนวณน้อยกว่า	1. ยากต่อการหาขอบเขตของหน่วยเสียง 2. มีส่วนของ word dictionary ที่ซับซ้อน

ซึ่งเป็นส่วนที่นำมาใช้ในการวิจัยครั้งนี้โดยในส่วนก่อนหน้าของ feature mesurement module นั้นจะมีส่วนของการทำ preprocessing เพิ่มมาอีกส่วนหนึ่งคือ การเตรียมข้อมูลและการตัดคำ (segmentation) เพื่อที่จะแยกส่วนของเสียงทางด้าน input ให้ได้เสียงที่เป็นคำเดี่ยวโดด ๆ เช่น ศูนย์, หนึ่ง เป็นต้น โครงสร้างระบบการรู้จำตามรูปที่ 3.1.2 ได้รับความนิยมใช้กันแพร่หลาย

และนำไปประยุกต์ในงานด้านต่าง ๆ ทั้งนี้เพราะ (Rabiner and Levinson, 1981)



รูปที่ 3.1.2 โครงสร้างระบบการรู้จำแบบ isolated word recognition
(Rabiner and Levinson, 1981)

ก) รูปแบบ model ไม่เปลี่ยนแปลงถึงแม้ว่า จำนวนคำที่ระบบจะสามารถจำได้ไม่เท่ากัน หรือวิธีในการหา parameter เพื่อที่จะแทนรูปแบบของคำจะต่างกัน เช่น LPC,FFT เป็นต้น ตลอดจน ในส่วนของ pattern similarity algorithms เช่น การใช้ DTW , รวมทั้งกฎการตัดสินใจ (decision rules) เพื่อที่จะหาผลลัพธ์ของการรับรู้จะแตกต่างกันก็ตาม model นี้ก็ยังคงสามารถใช้งานได้เป็นอย่างดี

ข) ง่ายต่อการนำมาใช้งาน

ค) ทำงานได้ดีในทางปฏิบัติ

3.1.1 feature measurement

โดยพื้นฐานแล้วในส่วนนี้จะเป็น data reduction technique โดยจะแปลงข้อมูลเสียงจำนวนมากให้เป็น feature ที่มีขนาดเล็กลง ซึ่งมีวิธีอยู่หลายวิธีเช่น energy and zero crossing rate (ปกติใช้ใน selected frequency band) หรืออาจใช้ short-time spectrum , linear-predictive coding (LPC) และ homomorphic model ในการเลือกใช้ feature ว่าจะใช้แบบใดนั้นจะขึ้นกับ

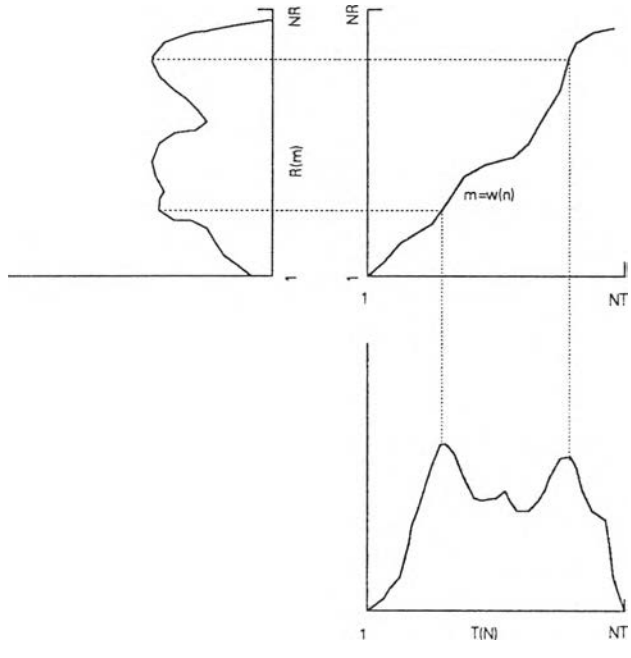
ก) เวลาในการคำนวณ (computation time)

ข) ขนาดของหน่วยความจำที่ต้องการ (storage)

ค) ความยากง่ายในการดำเนินการ (ease of implementation)

3.1.2 Time registration of pattern

เนื่องจากว่าในการพูดแต่ละครั้งถึงแม้จะเป็นผู้พูดคนเดียวกันจะมีอัตราการพูดที่เปลี่ยนแปลงไม่เท่ากัน หรือการพูดคำ ๆ เดียวกันแต่ต่างผู้พูดก็จะมีความแตกต่างกัน จึงต้องมีการวัดความคล้ายคลึงกันระหว่าง แบบทดสอบกับแบบอ้างอิง ซึ่งวิธีการนี้จะเกี่ยวข้องกับ time alignment และ distance computation



รูปที่ 3.1.2.1 ตัวอย่างของ time registration ของแบบทดสอบและแบบอ้างอิง (Rabiner and Levinson, 1981)

ในรูปที่ 3.1.2.1 จะแสดงถึง function time alignment ระหว่างแบบทดสอบ T(t) และแบบอ้างอิง R(t) โดยเราจะหา alignment function w(t) ซึ่งได้จากการ map R ไปบน T จากนั้นจะทำการหาค่า distance ของ function w(t) ที่ได้ดังสมการ

$$D(T,R) = \min_{\{w(t)\}} \int_{t0}^{t1} d(t,w(t))G(t,w(t),w(t))dt \qquad \text{....3.1.1}$$

w(t) เป็น set of monotonically increas ng , continuous differentiable function

d(t,w(t)) เป็น metric d(T(t),R(w(t))) ซึ่งเป็น pointerwise distance จาก R ไปยัง T

G เป็น weighting function

ในการหา optinum time alignment path w นี้จะเป็น curve แสดงความสัมพันธ์ระหว่าง m time axis ของแบบอ้างอิงต่อ n time axis ของแบบทดสอบ มีความสัมพันธ์เป็น

$$m = w(n)$$

โดยที่วิธีในการหา w(t) path มีหลายวิธีเช่น

3.1.2.1. linear time alignment ตัวอย่างเช่น

$$m = w(n) = (n-1)\frac{(NR-1)}{(NT-1)} + 1 \qquad \text{....3.1.2}$$

3.1.2.2. time event matching เป็นวิธีการที่ให้ความสำคัญของเหตุการณ์ที่เกิดขึ้นในส่วนของแบบทดสอบและแบบอ้างอิง โดยอยู่ในขอบเขตที่กำหนด

$$\begin{aligned} m1 &= w(n1) \\ m2 &= w(n2) \\ &\vdots \\ mQ &= w(nQ) \end{aligned} \qquad \text{....3.1.3}$$

3.1.2.3. correlation maximization จะเปลี่ยนแปลงเพื่อให้เกิดค่ามากที่สุดของ การทำ correlation ระหว่างแบบอ้างอิงและแบบทดสอบ

$$R^{*} = \max_{w(n)} \sum_{n} (T(n)R(w(n))) \qquad \text{....3.1.4}$$

3.1.2.4. dynamic time warping warp curve จะถูกกำหนดโดย

$$D^{*} = \min_{w(n)} \left[\sum_{n=1}^{NT} d(T(n), R(w(n))) \right] \qquad \text{....3.1.5}$$

โดยที่ $d(T(n), R(w(n)))$ จะเป็นค่าของ distance ของ frame n ของแบบทดสอบ และ frame ที่ w(n) ของแบบอ้างอิง ซึ่งวิธีการนี้นำมาประยุกต์ใช้อย่างกว้างขวางใน speech recognition system

3.1.3 decision rule for recognition

จากในรูปที่ 3.2 จะเห็นได้ว่า ในส่วนนี้จะเป็นส่วนท้ายสุด ซึ่งจะเป็นส่วนที่เลือกรูปแบบอ้างอิงที่เหมาะสมที่สุดสำหรับแบบทดสอบที่นำมาทดสอบ ซึ่งมี 2 วิธีคือ nearest neighbor rule (NN rule) และ K-nearest neighbor rule (KNN rule)

3.1.3.1 nearest neighbor rule (NN rule) กำหนดให้มีรูปแบบอ้างอิงอยู่ V รูปแบบ R^i, $i = 1,2,\ldots,V$ โดยที่แต่ละแบบจะให้ average distance เป็น D^i ซึ่งได้จาก DTW algorithm ผลการรับรู้จากกฎการตัดสินใจนี้จะนำไปทำการเลือกแบบอ้างอิงที่มีระยะทางจากแบบน้อยที่สุด, R^{i^*}

$$i^* = \underset{i}{argmin}\left[D^i\right] \qquad \ldots.3.1.6$$

ซึ่งเราสามารถจัดเรียงลำดับของระยะการวัดใหม่ได้เป็น

$$D^{[1]} \le D^{[2]} \le \cdots \le D^{[V]} \qquad \ldots.3.1.7$$

3.1.3.2 K-nearest neighbor rule (KNN rule) ในกรณีที่รูปแบบอ้างอิงแต่ละรูปแบบ (pattern) มีด้วยกันหลายชุด กำหนดให้แต่ละรูปแบบอ้างอิง (reference pattern) V แบบ มีอยู่ P ชุด ซึ่งจากการวัดระยะทางจาก DTW ของรูปแบบ (pattern) ที่ i จำนวน P ชุด แสดงได้ด้วย R^{ij} , $1 \le i \le V, 1 \le j \le P$ ซึ่งเราสามารถจัดเรียงได้ใหม่เป็น

$$D^{i,[1]} \le D^{i,[2]} \le \cdots \le D^{i,[P]} \qquad \ldots.3.1.8$$

โดยที่ KNN rule จะหา average distance ได้จาก

$$r^i = \frac{1}{K}\sum_{k=1}^{K} D^{i,[k]} \qquad \ldots.3.1.9$$

โดยที่เราสามารถเลือกผลของการรู้จำได้จาก

$$i^* = \underset{i}{argmin}\left[r^i\right] \qquad \ldots.3.1.10$$

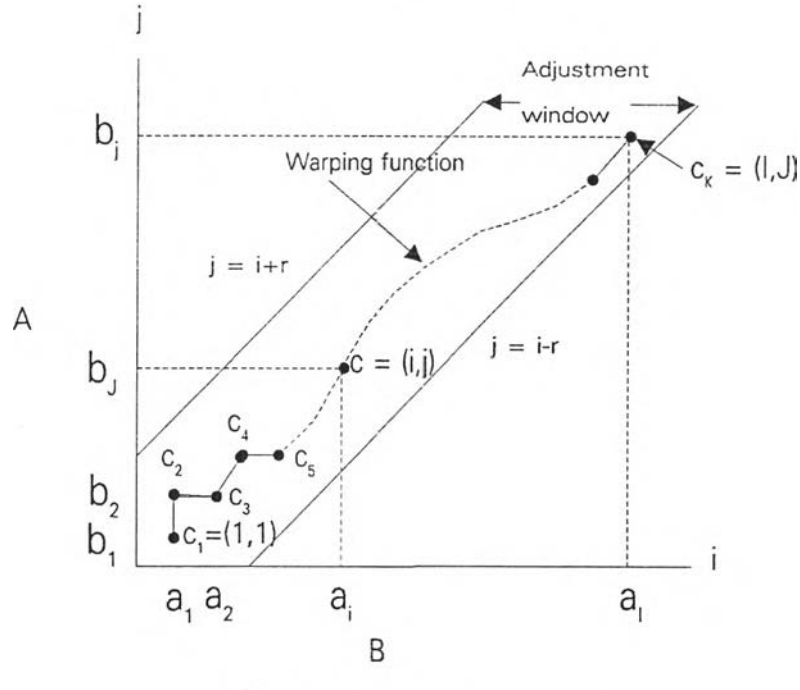
3.2 ไดนามิกไทม์วาร์ปปิง (Dynamic Time Warping) (Furui, 1989)

เทคนิคของ dynamic programming ที่นำมาใช้กับ time registration ของแบบทดสอบและแบบอ้างอิงอย่างกว้างขวาง ใน isolated word recognition จากที่ผ่านมาพื้นฐานของ time warping algorithm ถูกนำเสนอโดย Sakoe และ Chiba และ Rabiner , Rosenberg และ Levinson ซึ่ง algorithms เหล่านี้จะใช้ time input ในรูปของ time pattern of feature vector ซึ่งได้จาก isolated word ซึ่งรู้จุดสิ้นสุดที่แน่นอน ซึ่งในการวิเคราะห์เราจะกำหนดลำดับของข้อมูลตามแกนเวลาจำนวน 2 ชุด ซึ่งในแต่ละชุดจะประกอบด้วยเวกเตอร์ตัวแปร (feature vectors) คือ

$$\begin{aligned} A &= a_1, a_2, \ldots, a_i, \ldots, a_I \\ B &= b_1, b_2, \ldots, b_j, \ldots, b_J \end{aligned} \qquad \ldots.3.2.1$$

โดยที่ A และ B จะถูกแสดงอยู่บนระนาบ i-j ดังแสดงในรูปที่ 3.2.1 ซึ่ง time warping function จะแทนตำแหน่งของจุดต่าง ๆ ในระนาบ i-j เมื่อ c = (i,j) แทนจุดต่าง ๆ ในระนาบ i-j จะสามารถเขียนลำดับได้เป็น

$$F = c_1, c_2, \ldots, c_k, \ldots c_K \qquad \ldots.3.2.2$$



รูปที่ 3.2.1 DTW ระหว่าง A และ B

(Furui, 1989)

โดยที่ d(c) = d(i,j) จะแทน spectral distance ระหว่าง feature vectors ทั้งสอง a_i และ b_i และผลรวมของ distance ตาม F จะสามารถหาได้จาก

$$D(F) = \frac{\sum_{k=1}^{K} d(c_k) w_k}{\sum_{k=1}^{K} w_k} \qquad \text{....3.2.3}$$

ค่า D(F) ที่คำนวณได้ยิ่งมีค่าน้อยจะถือว่า feature vectors ระหว่าง A และ B เป็น feature vectors ที่ใกล้เคียงกันที่สุด โดยที่ w_k เป็นสัมประสิทธิน้ำหนัก (weight cofficient) ซึ่งจะทำให้การวัดมีความยืดหยุ่นขึ้น โดยที่ตัวหาร $\sum w_k$ จะเป็นตัวชดเชยค่าของ k โดยที่สมการที่ 3.2.3 จะสามารถเปลี่ยนให้อยู่ในรูปที่ง่ายตาม function บน F ภายใต้เงื่อนไขดังต่อไปนี้

ก. เงื่อนไขโมโนโทนิคและความต่อเนื่อง (Monotony and continuity condition)

$$\begin{aligned} 0 \le i_k - i_{k-1} \le 1 \\ 0 \le j_k - j_{k-1} \le 1 \end{aligned} \qquad \text{....3.2.4}$$

ตัวอย่างรูปแบบแสดงดังในรูปที่ 3.2.2 โดยที่สามารถเขียนเส้นทางการเดินทางไปยังจุด (n,m) ได้ 3 เส้นทางตาม local constraints รูปแบบที่ 1 ได้คือ

$$\begin{aligned} P &\rightarrow (1,0)\ (1,1) \\ P &\rightarrow (1,1) \\ P &\rightarrow (0,1)\ (1,1) \end{aligned} \qquad 3.2.5$$

ข. เงื่อนไขขอบเขต (Boundary condition)

$$\begin{aligned} i(1) &= 1,\ j(1) = 1 \\ i(K) &= I,\ j(K) = J \end{aligned} \qquad \text{....3.2.6}$$

ค. เงื่อนไขหน้าต่างการปรับตัว (Adjustment window condition)

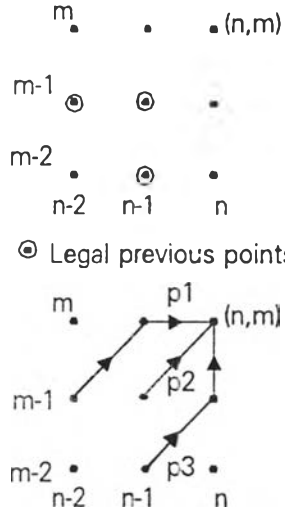
$$|i_k - j_k| \le r \quad , r = \text{ค่าคงที่} \qquad \text{....3.2.7}$$

ถ้า $w_k = (i_k - i_{k-1}) + (j_k - j_{k-1}), (i_0 = j_0 = 0)$ จะได้ว่า

$$\sum_{k=1}^{K} w_k = I + J \qquad \text{....3.2.8}$$

จากสมการที่ 3.2.3 จะได้ว่า

$$D(F) = \frac{I}{I+J} \sum_{k=1}^{K} d(c_k) w_k \qquad \text{....3.2.9}$$

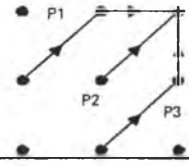
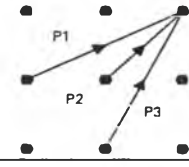
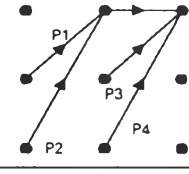
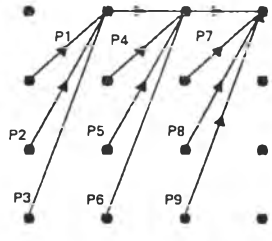
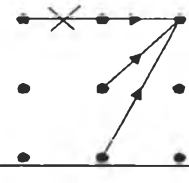


Type 1 local constraints

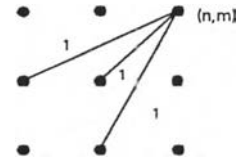
รูป 3.2.2 แสดง local path constraints ที่ไปยังจุด (n,m)

(Myers,Rabiner, and Ronsenberg, 1980)

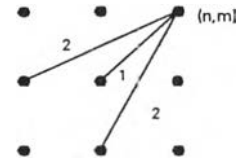
ตารางที่ 3.2.1 ตัวอย่างของชนิดของ local constraints (Myers et al., 1980)

Type	pictorial	productions	E_{max}	E_{min}
1		P1 → (1,0)(1,1) P2 → (1,1) P3 → (0,1)(1,1)	2	1/2
2		P1 → (2,1) P2 → (1,1) P3 → (1,2)	2	1/2
3		P1 → (1,0)(1,1) P2 → (1,0)(1,2) P3 → (1,1) P4 → (1,2)	2	1/2
4		P1 → (1,0)(1,0)(1,1) P2 → (1,0)(1,0)(1,2) P3 → (1,0)(1,0)(1,3) P4 → (1,0)(1,1) P5 → (1,0)(1,2) P6 → (1,0)(1,3) P7 → (1,1) P8 → (1,2) P9 → (1,3)	3	1/3
itakura		no production rule characterization	2	1/2

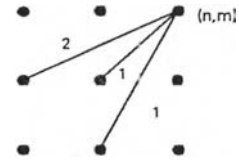
แบบ (a) $w_k = \min(i(k)-i(k-1), j(k)-j(k-1))$



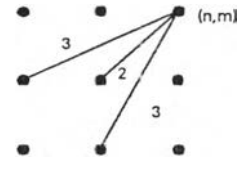
แบบ (b) $w_k = \max(i(k)-i(k-1), j(k)-j(k-1))$



แบบ (c) $w_k = i(k)-i(k-1)$



แบบ (d) $w_k = i(k)-i(k-1)+j(k)-j(k-1)$



รูปที่ 3.2.3 ตัวอย่างของ weighting function of Type 2 constraints

(Myers et al., 1980)

$$
\begin{aligned}
w_k &= \min(i(k)-i(k-1), j(k)-j(k-1)) \\
w_k &= \max(i(k)-i(k-1), j(k)-j(k-1)) \\
w_k &= i(k)-i(k-1) \\
w_k &= i(k)-i(k-1)+j(k)-j(k-1)
\end{aligned}
\qquad \text{....3.2.10}
$$

จากในรูปที่ 3.2.3 จะแสดงการ ให้น้ำหนัก (weighting) 4 แบบที่ใช้กับ Type 2 ในตารางที่ 3.2.1 โดยที่ i(0) = j(0) = 0 การให้น้ำหนักจะเท่ากันหมดในแบบ (a) ส่วนในแบบ (b) นั้น ค่าของความชัน (slope) เป็น 1/2 และ 2 จะมี การให้น้ำหนักมากกว่าที่ความชัน 1 ส่วนแบบ (c) การให้น้ำหนักจะขึ้นกับ distance ที่เคลื่อนที่ไปตามแกน x สำหรับในแบบ (d) นั้นการให้น้ำหนักจะเป็นไปตาม distance ที่เคลื่อนไปตามแกน x และแกน y ส่วนในรูปที่ 3.2.4 จะแสดงการ weight ที่ประยุกต์ใช้กับ Type 1 constraints ดังในรูปที่ 3.2.4 ทางด้านซ้ายมือ ส่วนทางด้านขวามือจะใช้

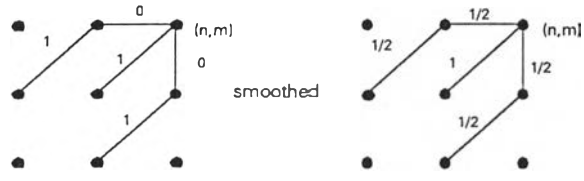
smoothing function กับการ weight ซึ่งเสนอโดย Sakoe และ Chiba สำหรับ Type (c) และ Type (d) ผลการ normalization จะได้ว่า

$$N(w_c) = \sum_{k=1}^{K} i(k)-i(k-1) = i(K) - i(0) = I$$

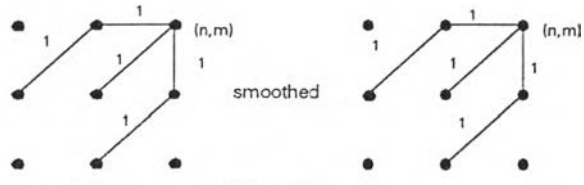
$$N(w_d) = \sum_{k=1}^{K} i(k)-i(k-1)+j(k)-j(k-1)) \quad \text{....3.2.11}$$

$$= i(K)-i(0)+j(K)-j(0) = I+J$$

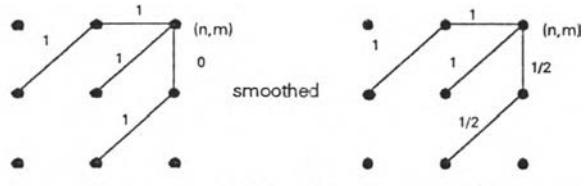
แบบ (a) $w_k = \min(i(k)-i(k-1), j(k)-j(k-1))$



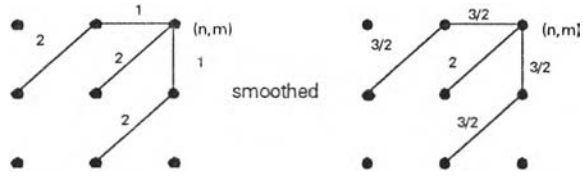
แบบ (b) $w_k = \max(i(k)-i(k-1), j(k)-j(k-1))$



แบบ (c) $w_k = i(k)-i(k-1)$



แบบ (d) $w_k = i(k)-i(k-1)+j(k)-j(k-1)$



รูปที่ 3.2.4 ตัวอย่างการทำ smoothed weighting function ของ Type 1 constraints (Myers et al., 1980)

จากสมการที่ 3.2.9 ในส่วนของ $\sum_{k=1}^{K} d(c_k)w_k$ จะเป็นการหาผลบวกที่มีค่าน้อยที่สุดภายใต้เส้นทางเดิน F ตามสมการที่ 3.2.2 ซึ่งจะสามารถหาผลรวมของ distance ลำดับ $c_1, c_2, \ldots, ck$ (ck

= (i,j)) ได้ดังนี้

$$g(c_k) = g(i,j) = \min_{c_1,\ldots,c_{k-1}} \left[\sum_{m=1}^{k} d(c_m) w_m \right]$$

$$= \min_{c_1,\ldots,c_{k-1}} \left[\sum_{m=1}^{k-1} d(c_m) w_m + d(c_k) w_k \right]$$

$$= \min_{c_{k-1}} \left[\min_{c_1,\ldots,c_{k-2}} \left\{ \sum_{m=1}^{k-1} d(c_m) w_m \right\} + d(c_k) wk \right]$$

$$= \min_{c_{k-1}} \left[g(c_k - 1) + d(c_k) w_k \right] \quad \ldots.3.2.12$$

จากสมการที่ 3.2.12 สามารถเขียนสมการได้เป็น

$$g(i,j) = \min \begin{pmatrix} g(i,j-1) + d(i,j) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{pmatrix} \quad \ldots.3.2.13$$

โดยกำหนดเงื่อนไขเริ่มต้นเป็น

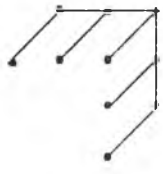
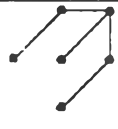
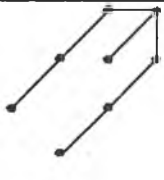
$$g(1,1) = 2d(1,1) \quad \ldots.3.2.14$$

การหาระยะทางของการวัดจะกำหนดให้ j = 1 จากนั้นทำการคำนวณตามค่าของ i ตาม adjustment window ที่กำหนด ทำการคำนวณโดยการเปลี่ยนค่าของ j ไปจนกระทั่ง j มีค่าเท่ากับ J ซึ่งจะให้ค่าของการวัดเป็น

$$D(F) = \frac{1}{I+J} G(I,J) \quad \ldots.3.2.15$$

ตารางที่ 3.2.2 แสดงสมการไดนามิกโปรแกรมมิ่งต่าง ๆ

(ไพศาล ธรรมโพธิทองม, 2533 อ้างถึงใน Sakoe, 1978)

P	แผนภาพ แสดงทางเดิน	สมมาตร / ไม่สมมาตร	สมการไดนามิกโปรแกรมมิ่ง g(i,j) =
0		สมมาตร	$\min \begin{cases} g(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-1, j) + d(i, j) \end{cases}$
		ไม่สมมาตร	$\min \begin{cases} g(i, j-1) \\ g(i-1, j-1) + d(i, j) \\ g(i-1, j) + d(i, j) \end{cases}$
1/2		สมมาตร	$\min \begin{cases} g(i-1, j-3) + 2d(i, j-2) + d(i, j-1) + d(i, j) \\ g(i-1, j-2) + 2d(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-2, j-1) + 2d(i-1, j) + d(i, j) \\ g(i-3, j-1) + 2d(i-2, j) + d(i-1, j) + d(i, j) \end{cases}$
		ไม่สมมาตร	$\min \begin{cases} g(i-1, j-3) + (d(i, j-2) + d(i, j-1) + d(i, j)) / 3 \\ g(i-1, j-2) + (d(i, j-1) + d(i, j)) / 2 \\ g(i-1, j-1) + d(i, j) \\ g(i-2, j-1) + d(i-1, j) + d(i, j) \\ g(i-3, j-1) + d(i-2, j) + d(i-1, j) + d(i, j) \end{cases}$
1		สมมาตร	$\min \begin{cases} g(i-1, j-2) + 2d(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-2, j-1) + 2d(i-1, j) + d(i, j) \end{cases}$
		ไม่สมมาตร	$\min \begin{cases} g(i-1, j-2) + (d(i, j-1) + d(i, j)) / 2 \\ g(i-1, j-1) + d(i, j) \\ g(i-2, j-1) + d(i-1, j) + d(i, j) \end{cases}$
2		สมมาตร	$\min \begin{cases} g(i-2, j-3) + 2d(i-1, j-1) + 2d(i, j-1) + d(i, j) \\ g(i-1, j-1) + 2d(i, j) \\ g(i-3, j-2) + 2d(i-2, j-1) + 2d(i-1, j) + d(i, j) \end{cases}$
		ไม่สมมาตร	$\min \begin{cases} g(i-2, j-3) + 2(d(i-1, j-1) + 2d(i, j-1) + d(i, j)) / 3 \\ g(i-1, j-1) + d(i, j) \\ g(i-3, j-2) + d(i-2, j-1) + d(i-1, j) + d(i, j) \end{cases}$

ในตารางที่ 3.2.2 แสดงตัวอย่างสมการของรูปแบบเส้นทางของ dynamic time warping แบบต่าง ๆ ส่วนของการทำ DTW ของแบบ P เท่ากับ 0 ดังในตารางที่ 3.3 สามารถแสดงได้ดังในรูปที่ 3.2.5

ในรูปที่ 3.2.6 จะเป็นการทำ normalize/warp DTW algorithm เพื่อปรับความยาวของแบบทดสอบและแบบอ้างอิง โดยให้อัตราส่วนของจำนวนเฟรมของแบบทดสอบและแบบอ้างอิงที่ผ่านการปรับ ($\tilde{N} / \tilde{M}$) มีค่าเท่ากับ 1 โดยที่การ normalize แบบ อ้างอิง $\tilde{R}(n)$ จะหาได้จาก

$$\tilde{R}(\tilde{n}) = (1-s)R(n) + s(R(n+1)), \quad \tilde{n} = 1,2,\ldots,\tilde{N} \qquad \ldots.3.2.16$$

โดยที่ R(n) แทน parameter vector ที่แทนรูปแบบอ้างอิงเฟรมที่ n

N แทนจำนวนเฟรมของแบบอ้างอิง

$\tilde{R}(n)$ แทน parameter vector ของแบบอ้างอิงที่ normalized

$\tilde{N}$ แทนขนาดของเฟรมของแบบอ้างอิงที่ mormalized

และ จะหาค่าของ n และ s ในสมการที่ 3.2.16 ได้จาก

$$n = \left[(\tilde{n}-1)\frac{(N-1)}{(\tilde{N}-1)} + 1 \right] \qquad \ldots.3.2.17a$$

$$s = (\tilde{n}-1)\frac{(N-1)}{(\tilde{N}-1)} + 1 - n \qquad \ldots.3.2.17b$$

ในลักษณะเดียวกัน จะสามารถทำการ normalize แบบทดสอบได้จาก

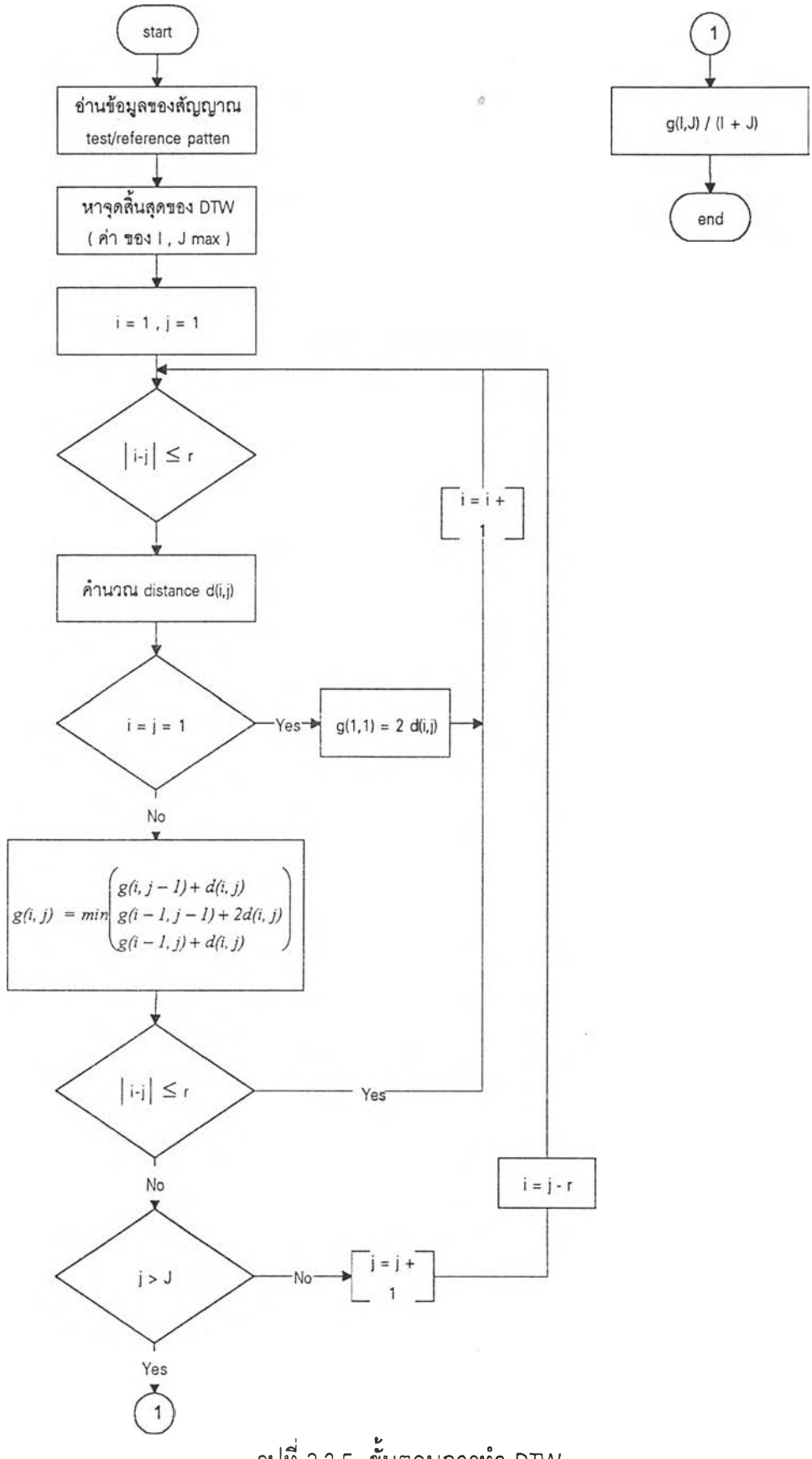
$$\tilde{T}(\tilde{m}) = (1-s)T(m) + s(T(m+1)), \quad \tilde{m} = 1,2,\ldots,\tilde{M} \qquad \ldots.3.2.18$$

โดยที่ T(m) แทน parameter vector ที่แทนรูปแบบทดสอบเฟรมที่ n

M แทนจำนวนเฟรมของแบบอ้างอิง

$\tilde{T}(\tilde{m})$ แทน parameter vector ของแบบทดสอบที่ normalized

$\tilde{M}$ แทนขนาดของเฟรมของแบบทดสอบที่ mormalized



รูปที่ 3.2.5 ขั้นตอนการทำ DTW

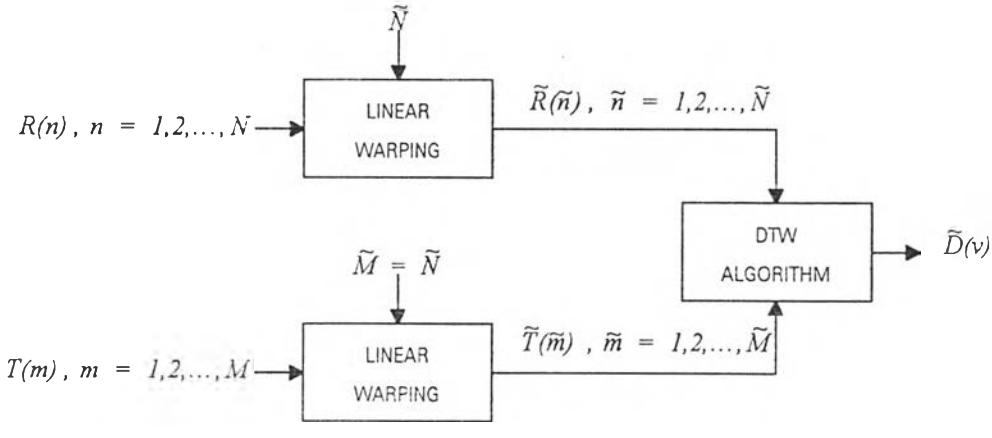
และ จะหาค่าของ n และ s ในสมการที่ 3.2.16 ได้จาก

$$m = \left[(\tilde{m} - 1)\frac{(M - 1)}{(\tilde{M} - 1)} + 1 \right] \quad \text{....3.2.19a}$$

$$s = (\tilde{m} - 1)\frac{(M - 1)}{(\tilde{M} - 1)} + 1 - m \quad \text{....3.2.19b}$$

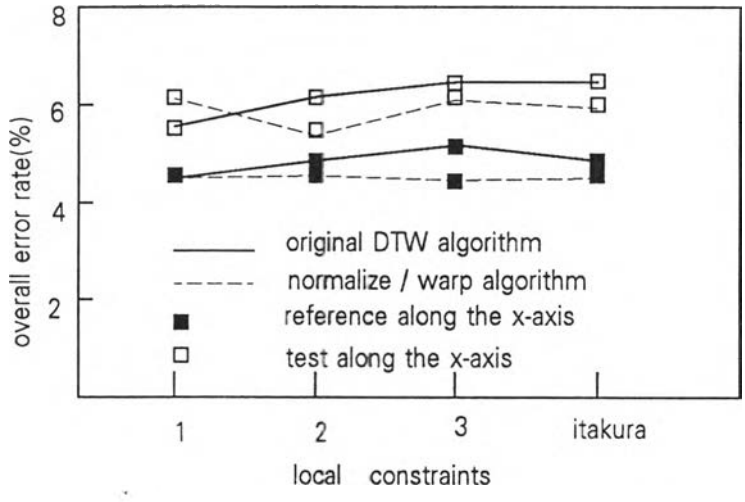
ส่วน $\tilde{D}(v)$ จะเป็นผลลัพธ์จากการทำ DTW ระหว่าง $\tilde{R}(n)$ และ $\tilde{T}(\tilde{m})$

จากสมการที่ 2.2.17a และ 2.2.19a ในเทอมของ [x] จะเป็นค่าของเลขจำนวนเต็มที่มีค่าไม่เกิน x ในส่วนของแบบอ้างอิงและแบบทดสอบตามลำดับ ซึ่งผลการทดสอบที่ผ่านมาแสดงดังในรูปที่ 3.2.7 โดยใช้ค่าของ $\tilde{N}$ เท่ากับ 40



รูปที่ 3.2.6 แสดงการใช้ normalize/warp DTW algorithm

(Myers et al., 1980)



รูปที่ 3.2.7 แสดงการเปรียบเทียบระหว่าง standard DTW และ normalize/warp DTW (Myers et al., 1980)

3.3 <u>การกำหนดพารามิเตอร์และวิธีการวัดในการวิเคราะห์เสียง</u>

เนื่องการการวิเคราะห์เสียงพูดนี้จะใช้วิธีในการตัดส่วนของสัญญาณเสียงออกเป็นช่วง ๆ (frame) แล้วทำการแปลงข้อมูลเสียงโดยใช้ดีสครีตฮาร์ตเลย์ทรานส์ฟอร์ม ซึ่งพารามิเตอร์ที่ได้ดังสมการที่ 3.3.1

$$P_i[k] = |H_i[k]| \qquad \text{....3.3.1}$$

เมื่อ i แทนหมายเลขเฟรม $i = 0,1,2,\ldots,I-1$

k จะแทนลำดับจะอยู่ในช่วง $0-(N_p-1), N_p = 256$

ค่าของ $P_i[k]$ นี้จะเป็นค่าสมบูรณ์ (abslute value) ของดีสครีตฮาร์ตเลย์ทรานส์ฟอร์ม ซึ่งในแต่ละเฟรมของการวิเคราะห์จะมีจำนวน 256 ค่า ซึ่งจะแทนข้อมูลของความถี่เสียงในช่วง 0-4 kHz ข้อมูลเสียงที่ได้นี้ จะทำการ normalize ด้วยค่าของค่าเฉลี่ยของสัญญาณดังในสมการที่ 3.3.2 เพื่อที่จะปรับสัญญาณให้เหมาะสมต่อการนำมาเปรียบเทียบ

$$P_{av} = \frac{\sum_{i=0}^{I-1}\sum_{k=0}^{N_p} P_i[k]}{I.N_p} \qquad \text{....3.3.2}$$

ค่าของ P_{av} จะเป็นค่าเฉลี่ยของสัญญาณที่ผ่านการแปลงข้อมูล ซึ่งจำนำค่าของ P_{av} ไปใช้ในการหาค่าที่ normalize ของสัญญาณดังแสดงในสมการที่ 3.3.3

$$\widetilde{P}_i[k] = \frac{P_i[k]}{P_{av}} \qquad 3.3.3$$

โดยที่ $\widetilde{P}_i[k]$ ที่ได้จากสมการที่ 3.3.3 นี้ยังคงมีค่ามากและมีการจัดเก็บเป็น floating point ซึ่งจะเปลืองเนื้อที่ในการจัดเก็บมาก ดังนั้นเราจะทำการตัดระดับแอมปลิจูดของสัญญาณนี้ออกเป็น 16 ระดับเพื่อสร้างเป็นรูปแบบของสัญญาณเสียง คือจะเก็บเป็นตัวเลข 0-15 แทน โดยจะพิจารณาจากค่าของ P_{av} ซึ่งจะทำการจัดเก็บลดลง 1 ใน 4 ของข้อมูลที่ได้จากการแปลง ซึ่งค่าดังกล่าวนี้จะนำมาใช้เป็นพารามิเตอร์ของเสียง ดังแสดงในตารางที่ 3.3.1

ตารางที่ 3.3.1 แสดงการกำหนดค่าของพารามิเตอร์เพื่อการรู้จำ

ค่าที่จัดเก็บ	ระดับของสัญญาณ $\widetilde{P}_i[k]$ (เป็นจำนวนเท่าของ P_{av})	
	แบบที่ 1	แบบที่ 2
0	0-1	0.0-0.5
1	1-2	0.5-1.0
2	2-3	1.0-1.5
3	3-4	1.5-2.0
4	4-5	2.0-2.5
5	5-6	2.5-3.0
6	6-7	3.0-3.5
7	7-8	3.5-4.0
8	8-9	4.0-4.5
9	9-10	4.5-5.0
10	10-11	5.0-5.5
11	11-12	5.5-6.0
12	12-13	6.0-6.5
13	13-14	6.5-7.0
14	14-15	7.0-7.5
15	15 ขึ้นไป	7.5 ขึ้นไป

การวัด distance ของพารามิเตอร์ที่ได้จะเป็นดังสมการที่ 3.3.4

$$d(i,j)=\sum_{n=0}^{K-1}(a_{in}-b_{jn})^2 \qquad \text{....3.3.4}$$

โดยที่ $d(i,j)$ จะเป็น distance ของเฟรมที่ i และ j

a_{in} เป็นพารามิเตอร์ของเสียงทดสอบที่เฟรมที่ i

b_{jn} เป็นพารามิเตอร์ของเสียงอ้างอิงที่เฟรมที่ j

K เป็นจำนวนพารามิเตอร์ที่ใช้ในการเปรียบเทียบ

3.4 การสร้างแบบอ้างอิง

ในการสร้างแบบอ้างอิงนี้เราจะใช้ค่าของพารามิเตอร์จากตารางที่ 3.3.1 มาสร้างเป็นแบบอ้างอิง โดยเราจะทำการหาค่าเฉลี่ยของแบบอ้างอิงแต่ละแบบ ดังแสดงในสมการที่ 3.4.1

$$R^i=\frac{\sum_{j=0}^{J-1}R_j^i}{J} \qquad \text{....3.4.1}$$

โดยที่ j จะแทนจำนวนบุคคลที่จะนำมาสร้างแบบอ้างอิง, $j=0,1,2,\ldots,J-1$

i แทนหมายเลขแบบอ้างอิง

R_j^i แทนรูปแบบของเสียงที่ i ของคนที่ j

R^i เป็นแบบอ้างอิงที่ i