

พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ระบบการจดจำเสียง Speech Recognition



โดย

นาย รุ่ง

กนกพัชรกุล

นาย อติสรณ์

เลิศสินทร์พยัทธิ

นางสาว เอมอร

อิทธิพลกุล

เลขหมู่.....
เลขทะเบียน..... **62784**
วัน,เดือน,ปี... **22 ส.ค. 2549**

b. **111-30223**
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาดมหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาเผยแพร่

ผ่านการตรวจค้นแล้ว
อภินันท์ อธิพงษ์
(ลงชื่อ).....ผู้ตรวจ
ผ่านการตรวจรูปเล่มแล้ว
HN
(ลงชื่อ).....ผู้ตรวจ

ระบบการจดจำเสียง
Speech Recognition

โดย

นายรจ	กนกพัชรกุล	45010652
นายอดิสรณ์	เลิศสินทรัพย์ทวี	45010901
นางสาวเอมอร	อิทธิพลังกุล	45010994

อาจารย์ที่ปรึกษา
ผศ.อัครพล ตวีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2548

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2548

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการจดจำเสียง

Speech Recognition

ผู้จัดทำ

1. นายรุ่ง กนกพัชรกุล 45010652
2. นายอดิสรณ์ เลิศสินทรัพย์ทวี 45010901
3. นางสาวเอมอร อธิธิพงษ์ 45010994


..... อาจารย์ที่ปรึกษา
(ผศ.อัครวัฒน์ ศรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการจดจำเสียง

SPEECH RECOGNITION

โดย นายรุ่ง กนกพัชรกุล 45010652
นายอดิสรณ์ เลิศสินทรัพย์ทวี 45010901
นส.เอมอร อธิพิฬังกุล 45010994

อาจารย์ที่ปรึกษา ผศ.อัศรพล ศรีรัตน์

บทคัดย่อ

ปริญญานิพนธ์นี้ เป็นการศึกษาและออกแบบระบบจดจำเสียง โดยอาศัยโปรแกรม MATLAB เป็นเครื่องมือในการวิเคราะห์และออกแบบ ซึ่งประกอบด้วย 3 ส่วนสำคัญ คือ ส่วนของการวิเคราะห์สัญญาณเบื้องต้นเพื่อหาคุณลักษณะเด่นของเสียง ส่วนของฐานข้อมูลเพื่อใช้ในการเปรียบเทียบ และ ส่วนของการจำแนกเสียงที่นำมาประมวลผลเพื่อตัดสินใจจำแนกเสียงให้เกิดความถูกต้อง

Abstract

This project studies and designs the speech recognition system. We will use MATLAB for analysis and design. This system consists of 3 majors parts which include feature extraction to find specification of speech, database for compare as well as recognize part to determine the speech accuracy.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สามารถสำเร็จได้ ด้วยการรับความช่วยเหลือจากบุคคลหลายท่าน คณะผู้จัดทำ ขอขอบพระคุณท่านอาจารย์ที่ปรึกษา ผศ.อักรพล ตริรัตน์ ที่ให้คำปรึกษาและความช่วยเหลือด้านอุปกรณ์ในการทำปริญญาานิพนธ์นี้เป็นอย่างดี ขอขอบพระคุณ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ที่เอื้อเพื่อให้ยืมอุปกรณ์ คือ บอร์ดประมวลผล TMS320C6713 ที่ใช้ในการทำปริญญาานิพนธ์ ขอขอบพระคุณ นายปานวิทย์ ฐะนุติ ที่ให้คำปรึกษาด้านการใช้บอร์ดประมวลผล TMS320C6713 และ ขอขอบคุณ นายธีรช พัทธ์ธีร์รัตน์ รวมทั้งเพื่อนๆทุกคน ที่คอยให้ความช่วยเหลือตลอดมา คณะผู้จัดทำรู้สึกไว้เสมอว่า หากไม่ได้รับความช่วยเหลือจากบุคคลที่กล่าวมานั้น ปริญญาานิพนธ์นี้จะไม่สามารถสำเร็จลุล่วงไปด้วยดี จึงขอขอบพระคุณมา ณ ที่นี้

นายรุจ กนกพัชรกุล
นายอดิสรณ์ เลิศสินทรัพย์ทวี
นางสาวเอมอร อธิธิพลังกุล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	
สารบัญ	
สารบัญรูปภาพ	
สารบัญตาราง	
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	3
2.1 เสียง	3
2.1.1 อวัยวะที่ใช้ในการออกเสียง	3
2.1.2 การเกิดของเสียง	3
2.2 การเข้ารหัสสัญญาณเสียง	4
2.3 หลักการวิเคราะห์เสียง	6
2.3.1 Front-end processing	6
2.3.1.1 หลักการพื้นฐานของการประมาณเชิงเส้น หรือ LPC	6
2.3.1.2 การจัดระดับเวกเตอร์ (Vector Quantization)	13
2.3.1.3 แบบจำลองมาร์คอฟ (Hidden Markov Models)	18
2.3.1.3.1 ส่วนประกอบของแบบจำลอง HMM	18
2.3.1.3.2 โครงสร้างของแบบจำลอง HMM	19
2.3.1.3.3 การคำนวณเพื่อแก้ไขปัญหาของ HMM	21
2.3.1.3.4 การปรับค่าพารามิเตอร์ของ HMM	28
2.3.1.3.5 ลำดับของค่าปรากฏหลายเหตุการณ์ (Multiple Observation Sequence)	31
2.3.1.3.6 ขั้นตอนการสร้างโมเดล	33
2.3.1.4 Decision rule	34
2.3.1.4.1 Viterbi Algorithm	34
บทที่ 3 การคำนวณและการออกแบบ	37
3.1 การวิเคราะห์เสียงเบื้องต้น	37
3.1.1 การพรีเอมฟาสิส (Pre-emphasis)	39
3.1.2 การแบ่งช่วงสัญญาณ (Frame blocking)	39
3.1.3 การวินโดว์	40
3.1.4 การวิเคราะห์ห่ออัตโนมัติ (autocorrelation)	41
3.1.5 การวิเคราะห์หาสัมประสิทธิ์ LPC	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6	การเปลี่ยนพารามิเตอร์ LPC เป็น สัมประสิทธิ์เซปสตรีม	44
3.1.7	การเวทค่าพารามิเตอร์ (Parameter weighting)	45
3.2	การสร้างโค้ดบุค	45
3.3	การหาค่าดัชนีโค้ดบุค	46
3.4	การออกแบบ Hidden Markov Model (HMM)	48
3.4.1	การสร้างแบบจำลอง HMM ของเสียง	48
3.5	ขั้นตอนการตัดสินใจ	49
3.6	การออกแบบระบบ เพื่อหาคุณลักษณะเด่นของเสียง โดย บอร์ด TMS320C6713	52
3.6.1	Pre-emphasis	52
3.6.2	Segmentation	53
3.6.3	Windowing	55
3.6.4	Auto Correlation	56
3.6.5	Levinson Durbin	57
บทที่ 4	ผลการทดลอง	59
4.1	ผลการทดลองในส่วนของการ Simulate ระบบการหาค่าคุณลักษณะเด่นของเสียง	59
4.1.1	ผลจากการทดลองหาค่า Autocorrelation	59
4.1.2	ผลจากการทดลองหาค่าสัมประสิทธิ์ LPC	61
4.1.3	ผลการทดลองหาค่า สัมประสิทธิ์ Cepstrum และ Weight พารามิเตอร์	65
4.2	ผลจากโปรแกรมสร้างโค้ดบุค	67
4.3	ผลการทดลองระบบการรู้จำเสียง	68
บทที่ 5	สรุปและวิจารณ์ผลการทดลอง	72
หนังสืออ้างอิง		74
ภาคผนวก		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมการรู้จำเสียงพูด	2
รูปที่ 2.1 แสดง องค์ประกอบของเสียง	4
รูปที่ 2.2 แสดง การกำเนิดเสียงและอวัยวะในการกำเนิดเสียง	4
รูปที่ 2.3 แสดง Basic isolated-word recognition system	6
รูปที่ 2.4 แสดง โครงสร้าง Speech synthesis โดยใช้หลักการ LPC	7
รูปที่ 2.5 การกระจายเฟรมของเสียงพูด แต่ละจุดแทนเฟรมของเสียง	14
รูปที่ 2.6 การรวมกลุ่มของเฟรมเสียงเพื่อนำไปสร้าง Codebook	14
รูปที่ 2.7 แสดงตัวอย่างการทำ Codebook	15
รูปที่ 2.8 บล็อกไดอะแกรมของเวกเตอร์ควอนไทซ์เซชัน	16
รูปที่ 2.9 เวกเตอร์ควอนไทซ์เซชันที่ใช้ในระบบการรับรู้เสียงพูด	18
รูปที่ 2.10 แสดง แบบจำลองต่างๆ ของ HMM	20
รูปที่ 2.11 แสดงกระบวนการไปข้างหน้า	23
รูปที่ 2.12 แสดง กระบวนการไปข้างหลัง	23
รูปที่ 2.13 แสดง ค่าปรากฏที่อยู่สถานะ i ที่เวลา t โดยคำนึงถึงลำดับค่าปรากฏจากเวลา $t+1$ ซึ่งต้องพิจารณา สถานะ j ที่จะเป็นไปได้ทั้งหมด ณ เวลาที่ $t+1$ โดยจะขึ้นอยู่กับค่า a_{ij} และ $b_j(o_{t+1})$	26
รูปที่ 2.14 แสดง ขั้นตอนการสร้าง โมเดล	33
รูปที่ 2.15 แสดง โมเดลของเสียงๆ หนึ่ง	34
รูปที่ 2.16 แสดง ขั้นตอนการตัดสินใจ	35
รูปที่ 2.17 แสดง การเปรียบเทียบ โมเดล	36
รูปที่ 3.1 แสดง Block diagram ของการวิเคราะห์เสียงเบื้องต้น	37
รูปที่ 3.2 แสดง ขั้นตอนการเตรียมสัญญาณในการวิเคราะห์	38
รูปที่ 3.3 แสดง วงจรกรองความถี่สูงผ่าน	39
รูปที่ 3.4 แสดง การแบ่งช่วงของสัญญาณ	40
รูปที่ 3.5 แสดงองค์ประกอบทางเวลา และ ทางความถี่ของฟังก์ชันวิน โคว์ แบบแฮมมิง	41
รูปที่ 3.6 แสดง ฟังก์ชันความสัมพันธ์	42
รูปที่ 3.7 Flow Chart แสดงการทำงานของโปรแกรมสร้าง Codebook	47
รูปที่ 3.8 Flow Chart แสดงการทำงานของโปรแกรมหาค่าดัชนีไค้คุณ	48
รูปที่ 3.9 แสดง Flow chart การทำงานของการวิเคราะห์เสียง	50
รูปที่ 3.10 แสดง Flow chart การทำงานของการเรียนรู้เสียง และ การสร้างโมเดลเสียง	51
รูปที่ 3.11 แสดง block diagram การหา feature attraction	52

ยกสารนี้เป็นเอกสารทูลงวันเวลาหรับการเพ่ง ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.12	แสดง Flow Chart ของ โปรแกรม Pre-emphasis	53
รูปที่ 3.13	แสดง Flow Chart ของ โปรแกรม Segmentation	54
รูปที่ 3.14	แสดง Flow Chart ของ โปรแกรม Windowing	55
รูปที่ 3.15	Flow Chart ในการเขียนโปรแกรม Auto correlation	56
รูปที่ 3.16	Flow Chart ในการเขียนโปรแกรม Levinson Durbin	57
รูปที่ 4.1	แสดง ค่า Autocorrelation เฟรมแรกของเสียงผู้ชายพูด "0" จากการเขียนด้วย MATLABเปรียบเทียบกับผลที่ได้ จากบอร์ด TMS	60
รูปที่ 4.2	แสดง ค่า Autocorrelation เฟรมแรกของเสียงผู้หญิงพูด "0" จากการเขียนด้วย MATLABเปรียบเทียบกับผลที่ได้ จากบอร์ด TMS	61
รูปที่ 4.3	แสดง ค่าสัมประสิทธิ์ LPC เฟรมแรกของเสียงผู้ชายพูด "0" จากการเขียนด้วย MATLABเปรียบเทียบกับผลที่ได้ จากบอร์ด TMS	62
รูปที่ 4.4	แสดง ค่าสัมประสิทธิ์ LPC เฟรมแรกของเสียงผู้หญิงพูด "0" จากการเขียนด้วย MATLABเปรียบเทียบกับผลที่ได้ จากบอร์ด TMS	63
รูปที่ 4.5	แสดง ค่าสัมประสิทธิ์ LPC ของเสียง "0-9" ที่พูดโดยผู้ชาย	64
รูปที่ 4.6	แสดง ค่าสัมประสิทธิ์ LPC ของเสียง "0-9" ที่พูดโดยผู้หญิง	64
รูปที่ 4.7	ค่าสัมประสิทธิ์ Cepstrum ของเสียงผู้ชายและผู้หญิงพูด "0" แสดงผลการทดลอง ของโปรแกรมย่อยที่ทำการหาสัมประสิทธิ์ Cepstrum	65
รูปที่ 4.8	ค่าสัมประสิทธิ์ Weight ของเสียงผู้ชายและผู้หญิงพูด "0" แสดงผลการทดลอง ของโปรแกรมย่อยที่ทำการหาพารามิเตอร์ Weight	65
รูปที่ 4.9	ทำการเปรียบเทียบพารามิเตอร์ LPC , Cepstrum, Weight ของเสียงผู้ชายพูด "0" ในเฟรมแรก	66
รูปที่ 4.10	ทำการเปรียบเทียบพารามิเตอร์ LPC, Cepstrum, Weight ของเสียงผู้หญิงพูด "0" ในเฟรมแรก	66
รูปที่ 4.11	เวกเตอร์พารามิเตอร์ของเฟรมต่างๆกับโค้ดบุคที่สุ่มได้ใน 2 มิติ ดังนั้นค่าจุดศูนย์กลาง ของกลุ่มจึงยังรวมตัวกันอยู่และกระจายครอบคลุม อินพุท ไม่ทั่วถึง	67
รูปที่ 4.12	เวกเตอร์พารามิเตอร์ของเฟรมต่างๆกับโค้ดบุคที่คำนวณได้ใน 2 มิติ ได้ผ่านกระบวนการ K-mean algorithm แล้วทำให้ codebook ที่ได้ กระจายและสามารถครอบคลุม อินพุท ที่เข้ามาได้ทั่วถึงมากขึ้น	67
รูปที่ 4.13	หน้าต่างโปรแกรม Speech Recognition	68
รูปที่ 4.14	หน้าต่างโปรแกรม Speech Recognition หลังจากทำการพูดเสียง "0" โดยผู้ชายคนหนึ่ง และ ทำการกดปุ่ม Recognize แล้ว ซึ่งได้ผลดังแสดง	68

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 4.1 แสดงการเปรียบเทียบค่า Autocorrelation ของเสียงผู้ชายพูด “0” (เฟรมแรก) จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS	59
ตารางที่ 4.2 แสดงการเปรียบเทียบค่า Autocorrelation ของเสียงผู้ชายพูด “0” (เฟรมแรก) จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS	60
ตารางที่ 4.3 แสดงการเปรียบเทียบค่าสัมประสิทธิ์ LPC และ Gain ของเสียงผู้ชายพูด “0” (เฟรมแรก) จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS	61
ตารางที่ 4.4 แสดงการเปรียบเทียบค่าสัมประสิทธิ์ LPC และ Gain ของเสียงผู้ชายพูด “0” (เฟรมแรก) จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS	62
ตารางที่ 4.5 ผลการทดสอบการรู้จำเสียงโดยแสดงค่าความถูกต้องเปรียบเทียบระหว่าง เสียงพูดแต่ละเสียง	69
ตารางที่ 4.6 ผลการทดสอบการรู้จำเสียง โดยแสดงค่าความถูกต้องเปรียบเทียบเมื่อผู้ทดสอบเป็นเพศชาย	70
ตารางที่ 4.7 ผลการทดสอบการรู้จำเสียง โดยแสดงค่าความถูกต้องเปรียบเทียบเมื่อผู้ทดสอบเป็นเพศหญิง	71

บทที่ 1

บทนำ

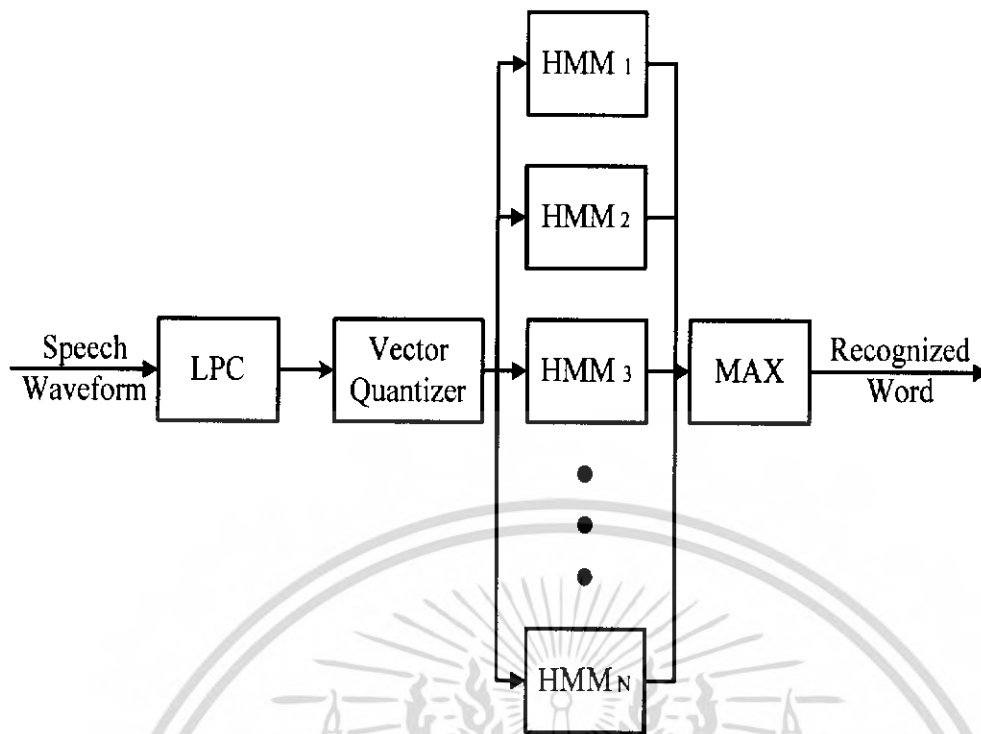
ในปัจจุบันนี้เทคโนโลยีทางด้านคอมพิวเตอร์ ได้เข้ามามีบทบาทในชีวิตประจำวันของมนุษย์เป็นอย่างมาก ทำให้การติดต่อสื่อสารเป็นไปได้อย่างสะดวกรวดเร็วและประหยัดค่าใช้จ่ายลงอย่างมาก ทำให้อุปกรณ์อำนวยความสะดวกมีราคาถูกลงตามไปด้วย ดังนั้นเราจึงสามารถนำเอาเทคโนโลยีใหม่ๆเข้ามาช่วยอำนวยความสะดวกในชีวิตประจำวันของเราได้

ในอดีตมีการวิเคราะห์ว่าการสอนให้คอมพิวเตอร์สามารถรู้จำเสียงพูดของมนุษย์นั้นไม่สามารถทำได้ เนื่องจากการพูดของมนุษย์นั้นมีความซับซ้อนและมีปัจจัยที่แตกต่างกันในแต่ละบุคคล เช่น ความเร็วในการพูด, ความสูงต่ำของเสียงพูด, ความดังความค่อยของเสียงพูด และสภาพแวดล้อมที่พูด เป็นต้น แต่ก็ได้มีการวิจัยเพื่อให้นักคอมพิวเตอร์สามารถรู้จำเสียงพูดของมนุษย์เรื่อยมา จนกระทั่งในปัจจุบันการรู้จำเสียงพูดสามารถใช้งานได้ดี และมีการนำไปใช้กับอุปกรณ์ต่างๆ เช่น บ้านอัจฉริยะ(Smart Home), เครื่องตอบรับโทรศัพท์อัตโนมัติ และนำมาประยุกต์ใช้ในโทรศัพท์เคลื่อนที่ในการดาวน์โหลดริงโทน เป็นต้น ถึงแม้ว่าคอมพิวเตอร์จะไม่สามารถรู้จำเสียงพูดได้เท่ากับมนุษย์ แต่เราสามารถสอนให้คอมพิวเตอร์สามารถรู้จำเสียงพูดได้ในขีดจำกัดระดับหนึ่ง โดยจะมีการกำหนดขอบเขตของการรู้จำเสียงพูด เช่นการจำกัดคำที่สามารถรับรู้ได้ และในปัจจุบันนี้เทคโนโลยีทางด้าน DSP (Digital Signal Processing) ถือว่าเป็นเทคโนโลยีที่น่าสนใจและมีบทบาทกับเทคโนโลยีทางด้านคอมพิวเตอร์อย่างมาก ดังนั้นเราจึงสามารถนำเทคโนโลยีทางด้าน DSP นี้มาประยุกต์ใช้กับเทคโนโลยีด้านการรู้จำเสียงพูดของมนุษย์ได้ ซึ่งจะมีส่วนช่วยในเรื่องของการประมวลผลข้อมูลให้มีประสิทธิภาพมากยิ่งขึ้น

วัตถุประสงค์ของปริญญาานิพนธ์ มีดังนี้

1. ศึกษาการวิเคราะห์เสียงพูดแบบไม่ขึ้นกับผู้พูด
2. ศึกษาขั้นตอนการประมาณเชิงเส้นของสัญญาณเสียงพูด โดยวิธีออดิโอดอร์รีเลชั่น เพื่อนำมาใช้ในการวิเคราะห์เสียงพูดของมนุษย์
3. สร้างแบบอ้างอิง(Code Book) โดยใช้วิธีการจัดระดับเวกเตอร์(Vector Quantization) ของพารามิเตอร์ที่ได้จากขั้นตอนของการประมาณเชิงเส้น
4. สร้างแบบจำลองของมาร์คอฟ(Hidden Markov Model) เพื่อใช้ในการรู้จำเสียงพูด
5. ศึกษาการใช้งานบอร์ด TMS320C6713

ปริญญาานิพนธ์นี้มีจุดประสงค์หลักคือ การรู้จำเสียงพูด 0-9 ซึ่งเป็นการเริ่มต้นศึกษาหลักการ ดังนั้น จึงเป็นเพียงการเขียนโปรแกรมเพื่อจำลองส่วนต่างๆ ตามบล็อกไดอะแกรม ดังรูปที่ 1.1



รูปที่ 1.1 บล็อกไดอะแกรมการรู้จำเสียงพูด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2 ทฤษฎีและหลักการ

2.1 เสียง

คนเราเปล่งเสียงพูดด้วยอวัยวะที่ใช้ในการออกเสียง ทำเสียงที่มีตามในภาษาของตน แม้ว่าคนที่อยู่ในสังคมเดียวกันจะใช้ภาษาเดียวกันแต่ถ้าพิจารณาเสียงที่เปล่งออกมาแล้ว แต่ครั้งก็อาจจะสังเกตเห็นลักษณะที่ต่างกันได้ เราจึงสามารถจำเสียง จำวิธีพูดของคนที่เราคุ้นเคยได้ เสียงพูดจะอธิบายด้วยหลักเกณฑ์ทางวิทยาศาสตร์ แม้ว่าในภาษาหนึ่งๆจะมีเสียงต่างกันมากบ้างน้อยบ้าง แต่เสียงก็สามารถนำมาพิจารณาและเรียนรู้ลักษณะการออกเสียงและตำแหน่งที่เกิดเสียงได้ คำอธิบายนี้จะทำให้เข้าใจลักษณะเสียงทุกเสียง

2.1.1 อวัยวะที่ใช้ในการออกเสียง

อวัยวะที่ใช้ในการออกเสียงมีอยู่หลายส่วน แต่ละส่วนสามารถทำให้เกิดเสียงต่างกันออกไปได้ อวัยวะเหล่านี้มีปาก และส่วนต่างๆในปาก ช่องคอ กล่องเสียง ช่องว่างในปาก และช่องว่างในจมูก อวัยวะที่ใช้ในการเปล่งเสียงพูดแบ่งออกเป็น 2 ประเภท คือ

1.) อวัยวะที่ใช้ในการทำอาการ (Articulator) คือ อวัยวะที่เคลื่อนไหว เพื่อผลัดลมไปยังส่วนต่างๆ อวัยวะที่สำคัญคือ ลิ้น ซึ่งเป็นส่วนที่เคลื่อนไหวมากที่สุด

2.) อวัยวะซึ่งเป็นตำแหน่งซึ่งเกิดเสียงต่างๆ (Point of Articulator) คือ ตำแหน่งที่เกิดเสียงต่างๆ เช่น ริมฝีปาก ฟัน เพดาน เป็นต้น

อวัยวะส่วนที่มีหน้าที่ออกเสียงโดยตรงมีดังนี้ ริมฝีปาก ฟัน ปุ่มเหงือก เพดานแข็ง เพดานอ่อน ลิ้น ใต้ลิ้น แผ่นเนื้อปากหลอดลม ช่องคอ เส้นเสียง ช่องจมูก

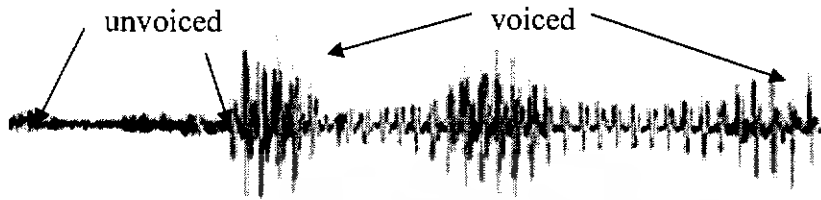
2.1.2 การเกิดของเสียง

การเกิดของเสียงแบ่งเป็น 3 ขั้นตอน คือ

- 1.) จุดเริ่มต้น เป็นขั้นตอนที่ลมเริ่มถูกขับออกจากปอด ผ่านเข้าไปสู่ขั้นตอนที่ 2
- 2.) การคัดแปลงลมที่อวัยวะเส้นเสียง ที่ใช้ในขั้นตอนนี้คือส่วนที่ต่อจากปอดขึ้นมาถึงกล่องเสียง และที่กล่องเสียง เส้นเสียงจะทำหน้าที่ปิดเปิดทำให้เกิดเสียง 2 ชนิด คือ

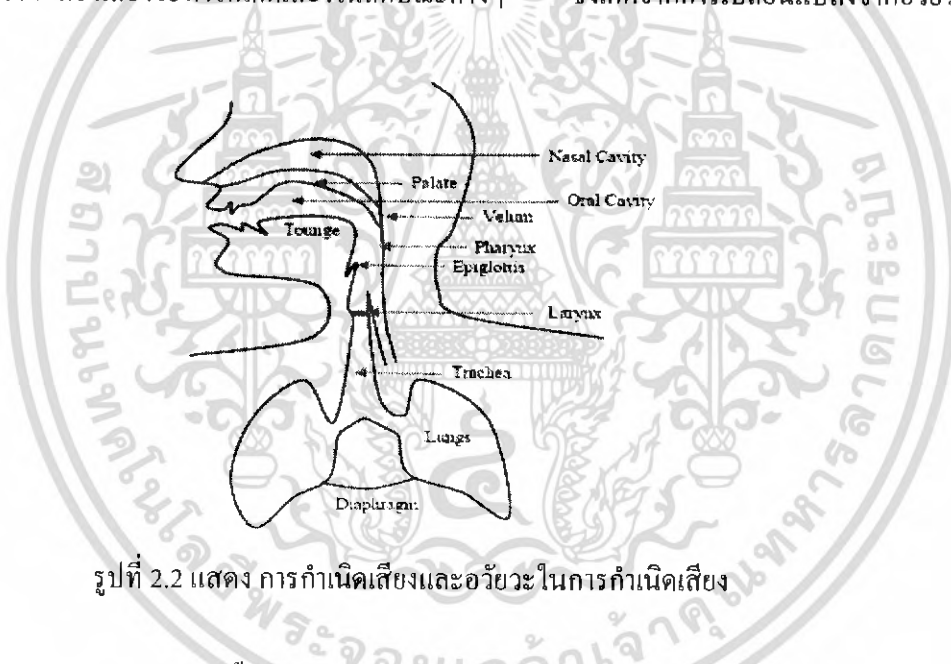
- **Voiced** เกิดจากเส้นเสียงปิดกันลมไว้ ลมที่ผ่านมาจะเพิ่มแรงดันมากขึ้นจนเส้นเสียงปิดเปิดสลับกัน ทำให้เกิดเสียงก้องขึ้นมา ซึ่งเรียกความถี่ในการปิด เปิด เส้นเสียงว่า “ความถี่มูลฐาน” มีลักษณะเป็นคาบ (periodic) และมีคาบของเสียง (pitch period) เป็นส่วนประกอบสำคัญ
- **Unvoiced** เสียงชนิดนี้เส้นเสียงจะเปิดตลอดเวลาที่ลมผ่าน ลมจึงผ่านออกมาได้ สะดวกทำให้เกิดเสียงไม่ก้องขึ้นมา มีลักษณะไม่เป็นคาบ แต่จะมีลักษณะเป็นสัญญาณของเสียงรบกวน (noise)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 แสดง องค์ประกอบของเสียง

3.) การเปลี่ยนแปลงลักษณะเส้นเสียง อวัยวะที่ใช้คือส่วนที่ต่อจากกล่องเสียงจนถึงริมฝีปาก โดยลมที่ผ่านออกจากกล่องเสียงจะทำให้เกิดเสียงในลักษณะต่างๆ ซึ่งเกิดจากการเปลี่ยนแปลงจากอวัยวะต่างๆ



รูปที่ 2.2 แสดง การกำเนิดเสียงและอวัยวะในการกำเนิดเสียง

ในการสร้างสัญญาณเสียงนั้นจะเลียนแบบรูปแบบทางกายภาพของมนุษย์ คือ ลม เปรียบได้กับสัญญาณกระตุ้น(Excitation Signal) จะเป็นตัวกระตุ้นให้เกิดสัญญาณเสียง

สัญญาณ Excitation ซึ่งถูกจำลองโดยใช้รูปแบบของฟิลเตอร์ (Filter) โดยมีค่า pitch เป็นความถี่ของเสียงที่ชัดที่สุด แล้ว นำสัญญาณกระตุ้นที่ได้มาปรับแต่งให้เหมือนกับเสียงที่ออกจากช่องปากจะได้สัญญาณเสียงที่สังเคราะห์ออกมา

2.2 การเข้ารหัสสัญญาณเสียง

การเข้ารหัสสัญญาณเสียงเกี่ยวกับการสุ่มข้อมูล (sampling) และการควอนไทซ์แอมพลิจูด (amplitude quantization) โดยขนาดอัตราสุ่มข้อมูลของเสียงพูดมักจะคงที่ ประมาณ 2 เท่าของขนาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แถบความถี่ของสัญญาณเสียงแบบอนาล็อก ดังนั้น การควอนไทซ์จะเป็นวิธีการสำคัญในการกำหนดรูปแบบการเข้ารหัสสัญญาณเสียงพูด ซึ่งแบ่งได้เป็น 2 แบบใหญ่ๆ คือ

1) การควอนไทซ์โดยตรง หรือ แบบนอนพารามตริก (Non parametric) คือ การนำสัญญาณโบนารีมาแสดงสัญญาณเสียงโดยตรง

2.) การควอนไทซ์แบบพารามตริก คือ การนำสัญญาณโบนารีมาแสดงแทนโมเดลของเสียงหรือพารามิเตอร์ทางความถี่ของเสียง

โดยวิธีการควอนไทซ์ที่สำคัญจะกล่าวถึงในหัวข้อถัดไป เสียงพูดโดยปกติของมนุษย์มีความถี่ไม่เกิน 4 กิโลเฮิร์ต และจะถูกสุ่มข้อมูลที่อัตรา 8 กิโลเฮิร์ต การเข้ารหัสเสียงโดยการควอนไทซ์โดยตรงที่ง่ายที่สุดคือ พัลส์โค้ดมอดูเลชัน (Pulse-Code Modulation : PCM) เสียงพูดที่เข้ารหัสที่อัตรา 64 กิโลบิตต่อวินาที ที่ใช้หลักการทิมพีซีเอ็ม (logarithm PCM) ถูกเรียกว่า การเข้ารหัสเสียงแบบไม่บีบอัดข้อมูล (non-compressed) ใช้เป็นการเข้ารหัสแบบอ้างอิงเทียบกับการเข้ารหัสแบบอื่นๆ ซึ่งอาจจะเข้ารหัสเสียงพูดได้ตามอัตราข้อมูลดังนี้

- 1.) การเข้ารหัสอัตราข้อมูลสูง ใช้อัตราข้อมูลมากกว่า 16 กิโลบิตต่อวินาที
- 2.) การเข้ารหัสอัตราข้อมูลขนาดกลาง ใช้อัตราข้อมูลระหว่าง 8-16 กิโลบิตต่อวินาที
- 3.) การเข้ารหัสอัตราข้อมูลต่ำ ใช้อัตราข้อมูลระหว่าง 2.4-8 กิโลบิตต่อวินาที
- 4.) การเข้ารหัสอัตราข้อมูลต่ำมาก ใช้อัตราข้อมูลน้อยกว่า 2.4 กิโลบิตต่อวินาที

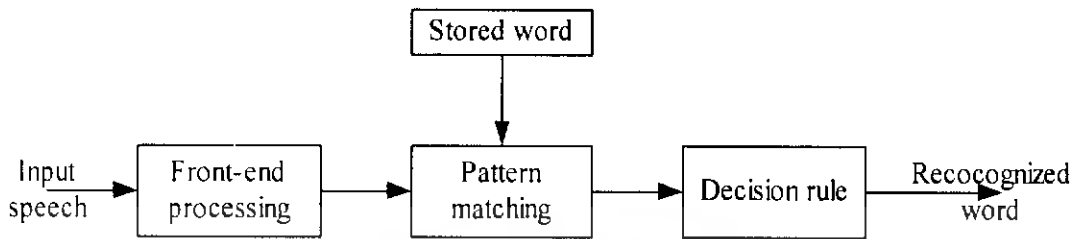
การเข้ารหัสเสียงพูดที่อัตราตั้งแต่ขนาดกลางลงมานั้นต้องใช้กระบวนการวิเคราะห์-สังเคราะห์ (analysis-synthesis) ในขั้นของการวิเคราะห์จะหาชุดของพารามิเตอร์ที่ใช้แทนสัญญาณเสียงที่ถูกเข้ารหัสได้อย่างมีประสิทธิภาพ และในขั้นของการสังเคราะห์ค่าพารามิเตอร์เหล่านี้จะถูกถอดรหัสและสร้างเสียงพูดกลับมา การวิเคราะห์อาจเป็นไปได้ทั้งแบบวงปิด (close loop) และแบบวงเปิด (open loop) ในแบบวงปิดค่าพารามิเตอร์จะถูกค้นหาจากความแตกต่างระหว่างเสียงต้นฉบับกับเสียงที่ถูกสร้างขึ้นมา นั่นคือในส่วนวงจรปิดจะต้องมีส่วนสังเคราะห์อยู่ภายใน กระบวนการแบบนี้ เรียกว่าการวิเคราะห์จากการสังเคราะห์ (analysis by synthesis)

การเข้ารหัสเสียงพูดแบบพารามตริกอาจเรียกอีกอย่างหนึ่งว่า “การเข้ารหัสเสียงตามลักษณะของเสียงพูด หรือ โวลโค้ดเดอร์ (speech-specific or voice coder : vocoder)” เป็นการเข้ารหัสที่เน้นในเรื่องคุณภาพของการรับฟังของเสียงพูดโดยไม่จำเป็นต้องได้สัญญาณที่เหมือนเดิมทุกประการ โวลโค้ดเดอร์สามารถทำงานที่อัตราข้อมูลต่ำมากๆ โดยให้คุณภาพของเสียงที่ระดับเสียงสังเคราะห์ โดยที่ในอัตราข้อมูลที่สูงขึ้นก็จะให้คุณภาพเสียงที่ดีขึ้น

ซึ่งการเข้ารหัสสัญญาณเสียงแบ่งได้เป็น 2 พวกใหญ่ๆ คือ การเข้ารหัสเสียงตามรูปคลื่น (waveform coding) และ การเข้ารหัสเสียงตามลักษณะของเสียงพูด หรือ โวลโค้ดเดอร์ การเข้ารหัสเสียงตามรูปคลื่น เช่น PCM , DPCM , ADPCM ส่วนในโครงการนี้จะใช้การเข้ารหัสเสียงตามลักษณะของเสียงพูดหรือโวลโค้ดเดอร์ ได้แก่ LPC-10 เพื่อหาลักษณะของเสียงพูดของแต่ละคน

2.3 หลักการวิเคราะห์เสียง

จากการศึกษารูปแบบการรู้จำเสียงพูดแบบคำเดียว (Isolated word recognition) มีรูปแบบและหลักการพื้นฐานดังรูปที่ 2.3



รูปที่ 2.3 แสดง Basic isolated-word recognition system

สัญญาณที่เข้ามาในโมเดล จะเป็นคลื่นเสียง ส่วนผลลัพธ์ที่ได้ จะเป็นเสียงที่ได้จากการวิเคราะห์จากสัญญาณอินพุต โมเดลนี้มีการใช้อย่างแพร่หลาย โดยแบ่งเป็นส่วนต่างๆ ดังนี้

- Front-end processing เป็นขั้นตอนของการวิเคราะห์เสียงที่ได้รับมา ให้อยู่ในรูปแบบที่สามารถนำไปวิเคราะห์ได้อย่างเหมาะสม
- Pattern matching เป็นขั้นตอนของการสร้างโมเดลเสียง
- Decision rule เป็นขั้นตอนของการตัดสินใจในการพิจารณาเลือกโมเดลเสียงที่ใกล้เคียงกับเสียงที่นำมาทดสอบมากที่สุด

2.3.1 Front-end processing

Front-end processing เป็นขั้นตอนที่จะทำการวิเคราะห์หาคุณลักษณะของเสียง (Feature extraction) คือ การแปลงข้อมูลที่มีอยู่มากมายให้เป็นส่วนเล็กๆ และสามารถแสดงคุณสมบัติของคลื่นเสียงนั้นๆออกมาได้ ซึ่งมีเทคนิคในการแปลงข้อมูลอยู่มากมาย โดยผ่านขั้นตอน ดังนี้

2.3.1.1 การประมาณเชิงเส้น (Linear Prediction Coding:LPC)

2.3.1.2 การจัดระดับเวกเตอร์ (Vector Quantization:VQ)

2.3.1.1 หลักการพื้นฐานของการประมาณเชิงเส้น หรือ LPC

จากแนวความคิดของ LPC Model ที่ซึ่ง ได้นำ Speech sample ที่เวลา n หรือ $s(n)$ ที่สามารถประมาณค่าด้วยการรวมแบบเชิงเส้น ที่มี p speech sample ด้วยสมการที่ 2.1

$$s(n) \approx a_1s(n-1) + a_2s(n-2) + \dots + a_p s(n-p) \quad (2.1)$$

เมื่อ สัมประสิทธิ์ a_1, a_2, \dots, a_p จะเป็น ค่าคงที่จากการวิเคราะห์ speech frame โดยเราสามารถเขียนสมการใหม่ได้ โดย เราจะเพิ่ม Excitation term $Gu(n)$ เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$s(n) = \sum_{i=1}^p a_i s(n-i) + Gu(n) \quad (2.2)$$

เมื่อ $u(n)$ คือ Normalized excitation และ G คือ เกนของ Excitation จากสมการที่ 2.2 สามารถแปลงให้อยู่ใน Z-domain ได้ ดังสมการ

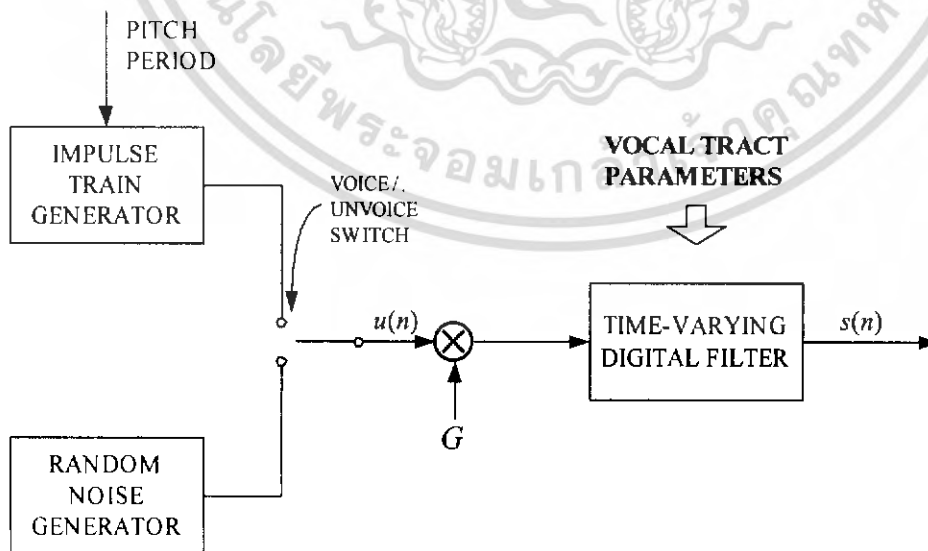
$$S(z) = \sum_{i=1}^p a_i z^{-1} S(z) + GU(z) \quad (2.3)$$

ดังนั้น เราสามารถหา Transfer function เป็น

$$H(z) = \frac{S(z)}{GU(z)} = \frac{1}{1 - \sum_{i=1}^p a_i z^{-1}} = \frac{1}{A(z)} \quad (2.4)$$

จาก Transfer function เราจะเห็นว่า ระบบนี้เป็น All pole system ซึ่งในการที่จะสร้างสัญญาณ Speech signal นั้น เราจะต้องคำนึงถึงการกำหนด Excitation function โดยต้องขึ้นอยู่กับสัญญาณที่เข้ามาว่าเป็น voice หรือ unvoice โดยถ้าเป็น voice เราจะได้ว่า Excitation function จะเป็น pulse train แต่ ถ้าเป็น unvoice จะได้ Excitation function เป็น random noise โดยวิธีการที่เหมาะสมนำมาสังเคราะห์โมเดลของ speech ก็คือ วิธีการของ LPC

จากหลักการพื้นฐานของการประมาณเชิงเส้น และ แบบจำลองระบบสร้างสัญญาณเสียง เราสามารถเขียน Block diagram ของการสังเคราะห์สัญญาณ speech ด้วยการทำการประมาณเชิงเส้น ที่นำมาสร้างสัญญาณเสียงพูด ได้ดังรูปที่ 2.4



รูปที่ 2.4 แสดง โครงสร้าง Speech synthesis โดยใช้หลักการ LPC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.4 สามารถเขียนสมการความสัมพันธ์ ระหว่าง $s(n)$ และ $u(n)$ จะ ได้เป็น

$$s(n) = \sum_{k=1}^p a_k s(n-k) + Gu(n) \quad (2.5)$$

การประมาณเชิงเส้น โดยใช้สัมประสิทธิ์ a_k คือ

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

ดังนั้น ค่าความคลาดเคลื่อน คือ

$$e(n) = s(n) - s'(n) = s(n) - \sum_{k=1}^p a_k s(n-k) \quad (2.7)$$

ดังนั้น จะมี Transfer function ระหว่าง $e(n)$ และ $s(n)$ คือ

$$A(z) = \frac{E(z)}{S(z)} = 1 - \sum_{k=1}^p a_k z^{-k} \quad (2.8)$$

จากการตั้งสมการที่ต้องแก้ เพื่อหาค่าสัมประสิทธิ์ Filter นั้น เราจะกำหนดให้ใช้ speech ที่สั้นๆ และ error ที่เป็นส่วนๆ ที่เวลา n คือ

$$s_n(m) = s(n+m)$$

$$e_n(m) = e(n+m)$$

เราสามารถหาค่าผลรวมของกำลังสองของความคลาดเคลื่อนของสัญญาณได้ คือ

$$E_n = \sum_m e_n^2(m) \quad (2.9)$$

$$E_n = \sum_m [s_n(m) - s'_n(m)]^2 \quad (2.10)$$

$$E_n = \sum_m \left[s_n(m) - \sum_{k=1}^p a_k s_n(m-k) \right]^2 \quad (2.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราต้องการค่า E_n ต่ำที่สุด จะต้องมีการเงินไว้ว่า

$$\frac{\partial E_n}{\partial a_k} = 0 \quad k = 1, 2, \dots, p \quad (2.12)$$

จากสมการที่ 2.11 จะได้

$$\frac{\partial E_n}{\partial a_k} = 2 \sum_m s_n(m) s_n(m-k) - \sum_{k=1}^p a_k \sum_m s_n(m-i) s_n(m-k) \quad (2.13)$$

เราต้องการให้ $\frac{\partial E_n}{\partial a_k} = 0$ ดังนั้นเราจะได้ว่า

$$\sum_m s_n(m) s_n(m-k) = \sum_{k=1}^p a_k \sum_m s_n(m-i) s_n(m-k) \quad (2.14)$$

เมื่อ $i = 1, 2, \dots, p$

ถ้าเรากำหนดให้ $\phi_n(i, k) = \sum_m s_n(m-i) s_n(m-k)$

จากสมการที่ 2.14 เราจะได้ว่า

$$\phi_n(i, 0) = \sum_{k=1}^p a_k \phi_n(i, k) \quad (2.15)$$

และจากสมการที่ 2.14 จะได้ว่า

$$\begin{aligned} E_n &= \sum_m s_n^2(m) - \sum_{k=1}^p a_k \sum_m s_n(m) s_n(m-k) \\ &= \phi_n(0, 0) - \sum_{k=1}^p a_k \phi_n(0, k) \end{aligned} \quad (2.16)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสมมติให้ 1 เฟรม ของสัญญาณ speech ที่ตัดมาเพื่อคำนวณ มีความยาว N โดย ในที่นี้ เรา กำหนดให้ $s_n(m) = 0$ เมื่อ $m < 0$ หรือ $m > N-1$ ดังนั้น เราจะได้ว่าค่าผลรวมของกำลังสองของ ความคลาดเคลื่อนของสัญญาณ จะได้เป็น

$$E_n = \sum_{m=0}^{N-1+p} e_n^2(m) \quad (2.17)$$

ถ้าแสดงในเทอมของ $\phi_n(i, k)$

$$\phi_n(i, k) = \sum_{m=0}^{N-1+p} s_n(m-i)s_n(m-k) \quad (2.18)$$

หรือ
$$\phi_n(i, k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k)$$

เมื่อ $1 \leq i \leq p$ และ $0 \leq k \leq p$

จะลดรูปให้เป็นฟังก์ชันของ ออโตคอร์รีเลชัน จะได้เป็น

$$\phi_n(i, k) = r_n(i-k) = \sum_{m=0}^{N-1-(i-k)} s_n(m)s_n(m+i-k) \quad (2.19)$$

เมื่อ ฟังก์ชันของออโตคอร์รีเลชันจะสมมาตรกัน คือ $R_n(-k) = R_n(k)$ เราจะสามารถแสดง สมการของ LPC ได้ ดังสมการ

$$\sum_{k=1}^p r_n(|i-k|) a_k = R_n(i) \quad (2.20)$$

เมื่อ $1 \leq i \leq p$

และ สามารถ แสดงให้อยู่ในรูป Matrix form ได้ ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \vdots \\ R_n(p) \end{bmatrix}$$

หรือ

$$R_n \cdot a = r_n \quad (2.21)$$

ซึ่งจะเป็น Matrix ขนาด $p \times p$ ของค่าออโตคอร์รีเลชัน เรียก Toeplitz Matrix (เป็น Matrix ที่สมมาตรในแนวทแยงมุม)

เมื่อ

$$R_n = \begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix}, a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix}$$

และ $r_n = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \vdots \\ R_n(p) \end{bmatrix}$

จากสมการที่ 2.21 เราสามารถหาค่าสัมประสิทธิ์ LPC (a_k) ได้จาก

$$a = R_n^{-1} \cdot r_n \quad (2.22)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การวิเคราะห์หาค่าออโตคอร์รีเลชัน (autocorrelation)

สมมุติว่า สัญญาณเดิมเป็น $s(n)$ การประมาณค่า สัญญาณเป็น $s'(n)$ ดังนั้น สามารถอธิบายการประมาณเชิงเส้นด้วยสมการ ต่อไปนี้

$$s'(n) = \sum_{k=1}^p a_k s(n-k) \quad (2.23)$$

เมื่อ a_k เป็น ค่าคงที่ เรียกวิธีการนี้ว่า การประมาณเชิงเส้น อันดับที่ p โดยมีเงื่อนไขว่า ค่า a_k ที่ใช้ในการประมาณจะต้องทำให้ ผลรวมของกำลังสองของความคลาดเคลื่อน คือ $(s(n) - s'(n))^2$ มีค่าน้อยที่สุด นั่นคือ $\sum e^2(n) = \sum (s(n) - s'(n))^2$ มีค่าที่ต่ำที่สุด ซึ่งจะใช้การประมาณเชิงเส้น แบบวิธีออโตคอร์รีเลชัน (Autocorrelation Method) หรือ วิธีอิตตสัมพันธ์

การคำนวณออโตคอร์รีเลชัน เป็นวิธีการ หาค่าสัมประสิทธิ์ LPC โดยฟังชั่นออโตคอร์รีเลชัน ซึ่งเป็นการเปรียบเทียบสัญญาณกับสัญญาณของตัวเองที่ถูกเลื่อนออกไปตามแกนเวลา การที่เราเลือกใช้วิธีนี้ เนื่องจาก เป็นการคำนวณที่มีความยุ่งยากในการคำนวณน้อยกว่าวิธีอื่นๆ และ มีความแน่นอนในด้านเสถียรภาพ อีกทั้งยังมีการเก็บข้อมูลที่น้อยกว่า

โดย คุณสมบัติของออโตคอร์รีเลชัน คือ

- เป็นฟังก์ชันคู่ $R(k) = R(-k)$
- จะเป็นค่าสูงสุด เมื่อ เปรียบเทียบสัญญาณของตัวเองที่ตำแหน่งทางแกนเวลาเดียวกัน คือ $R(0) = \max$

- การหาค่าอัตราขยาย (G)

เราสามารถหาค่าอัตราขยายได้จากสมการ

$$G^2 = \frac{R_n(0) - \sum_{k=1}^p a_k R_n(k)}{\sum_{m=0}^{N-1} u^2(m)} \quad (2.24)$$

ดังนั้น หลักการ LPC ก็คือ การประมาณเสียงจากผลรวมเชิงเส้นของสัญญาณเสียงในอดีต โดยอาศัย หลักการของค่ากำลังสองของสัญญาณความคลาดเคลื่อนให้มีค่าที่ต่ำที่สุด ในการหาสัมประสิทธิ์การประมาณเชิงเส้น กล่าวคือ จะทำการหาค่าออโตคอร์รีเลชัน เพื่อ นำไปทำการหา สัมประสิทธิ์ LPC และ อัตราการขยาย ที่เหมาะสม

จากสมการการหาค่าสัมประสิทธิ์ LPC (2.21) จะสามารถหาได้ โดยการแก้สมการหา Inverse Matrix ซึ่งในที่นี้ เราต้องการหาค่าสัมประสิทธิ์ LPC (Order 10) ซึ่งจะยุ่งยากมากในการสร้างโปรแกรมแก้หา Inverse Matrix โดยตรง เราจึงนำ Algorithm ของ Levinson-Durbin มาช่วยในการหาค่าสัมประสิทธิ์ LPC แทนการแก้สมการ Inverse Matrix

ซึ่งมีสมการ ดังนี้

$$k_p = \frac{r_{p-1}^* \tilde{a}_{p-1}}{E_{p-1}} \quad (2.25)$$

$$a_p = \begin{bmatrix} a_{p-1} \\ 0 \end{bmatrix} - k_p \begin{bmatrix} 0 \\ \tilde{a}_{p-1}^* \end{bmatrix} \quad (2.26)$$

$$E_p = (1 - |k_p|^2) E_{p-1} \quad (2.27)$$

สำหรับคำตอบที่เราต้องการจะเท่ากับ a_p โดยที่ $p=0,1,\dots,10$

2.3.1.2 การจัดระดับเวกเตอร์ (Vector Quantization)

การรับรู้เสียงพูดนั้นเป็นลักษณะหนึ่งของการรับรู้รูปแบบ (Pattern Recognition) คือจะเป็นการเปรียบเทียบระหว่างแบบทดสอบ (Test Pattern) กับแบบอ้างอิง (Reference Pattern) ซึ่งเป็นรูปแบบที่เราทราบและเก็บไว้ล่วงหน้า

ขั้นตอนการรับรู้แบ่งเป็น 2 ขั้นตอน ดังนี้คือ

1. ขั้นเรียนรู้ (Learning) จะเป็นการสร้างกลุ่มของแบบอ้างอิงในการรับรู้เสียงพูด ในขั้นนี้จะทำการวิเคราะห์เสียงพูดก่อนแล้วเก็บลักษณะของเสียงพูดในรูปของพารามิเตอร์ ดังได้กล่าวมาแล้วในข้างต้น และใช้เวกเตอร์ควอนไทซ์เซชันในการสร้างแบบอ้างอิง เพื่อใช้เปรียบเทียบในขั้นตอนต่อไป

2. ขั้นรับรู้ (Recognition) จะเป็นการทดสอบการรับรู้ระหว่างแบบอ้างอิงกับแบบทดสอบ โดยจะทำการเปรียบเทียบพารามิเตอร์ของแบบทดสอบกับแบบอ้างอิงทั้งหมด แบบอ้างอิงที่เลือกคือแบบอ้างอิงที่มีพารามิเตอร์ใกล้เคียงกับแบบทดสอบมากที่สุด

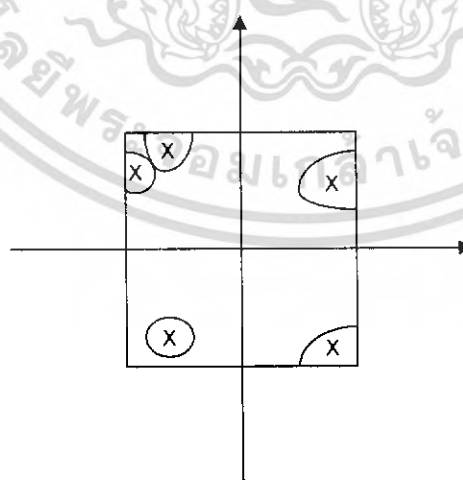
ในทฤษฎีของการสื่อสารนั้น แหล่งเข้ารหัสจะใช้เทคนิคการเปลี่ยนค่าสัญญาณเอาต์พุตไปเป็นรหัสแบบไบนารี ซึ่งเป้าหมายของเครื่องเข้ารหัสก็คือ การเป็นไปได้ที่จะมีอัตราการผิดพลาดของข้อมูลน้อยที่สุด ด้วยอัตราความเร็วที่กำหนดไว้

เวกเตอร์ควอนไทซ์เซชันเป็นเทคนิคการเข้ารหัสเทคนิคหนึ่งที่ค่อนข้างมีคุณภาพ โดยวิธีการเข้ารหัสชนิดนี้จะเป็นการลดโดเมนชั้น (Dimension) หรือจำนวนของข้อมูลเวกเตอร์อินพุตหรือเซตเทรน

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิ่ง (Training set) หรือพารามิเตอร์ที่ได้จากขั้นตอน LPC โดยจะถูกเลือกมากลุ่มหนึ่งซึ่งจะใช้เป็นตัวแทนของข้อมูลจำนวนหนึ่งหรือที่เรียกว่าการหา Codebook อินพุทที่เข้ามาจะถูกนำมาทำการเปรียบเทียบกับ Codebook ที่มีอยู่ โดยจะทำการพิจารณาว่าอินพุทที่เข้ามานั้นห่างจาก Codebook ใดน้อยที่สุด อินพุทดังกล่าวจะถูกแทนด้วยเวกเตอร์ไค้ด (index) นั้น อินพุททุกตัวที่เป็นสมาชิกของเวกเตอร์ไค้ดใดๆจะถูกนำมาหาจุดศูนย์กลางร่วมใหม่ และนำจุดศูนย์กลางที่ได้นี้ไปทำการหาค่าความคลาดเคลื่อนกับสมาชิกทุกตัว ถ้าค่าความคลาดเคลื่อนที่ได้มีค่ามากกว่าค่าที่กำหนดไว้ค่าหนึ่งหรือค่าที่ยอมรับได้ ก็จะนำจุดศูนย์กลางใหม่นั้นไปเป็น Codebook แทน และจะทำการจัดกลุ่มอินพุทเข้ากับ Codebook ใหม่ที่ได้และหาค่าความคลาดเคลื่อนอีกครั้ง ทำอย่างนี้ซ้ำจนกระทั่งค่าความคลาดเคลื่อนมีค่าน้อยถึงค่าที่ยอมรับได้ก็จะถือว่าได้ Codebook ที่ดีที่สุดที่จะเป็นตัวแทนของอินพุททั้งหมด จะสังเกตได้ว่าทุกครั้งที่มีการหา Codebook ใหม่ นั้นค่าความคลาดเคลื่อนที่ได้จะมีค่าลดลงทุกครั้งด้วย

รูปที่ 2.5 การกระจายเฟรมของเสียงพูด แต่ละจุดแทนเฟรมของเสียง

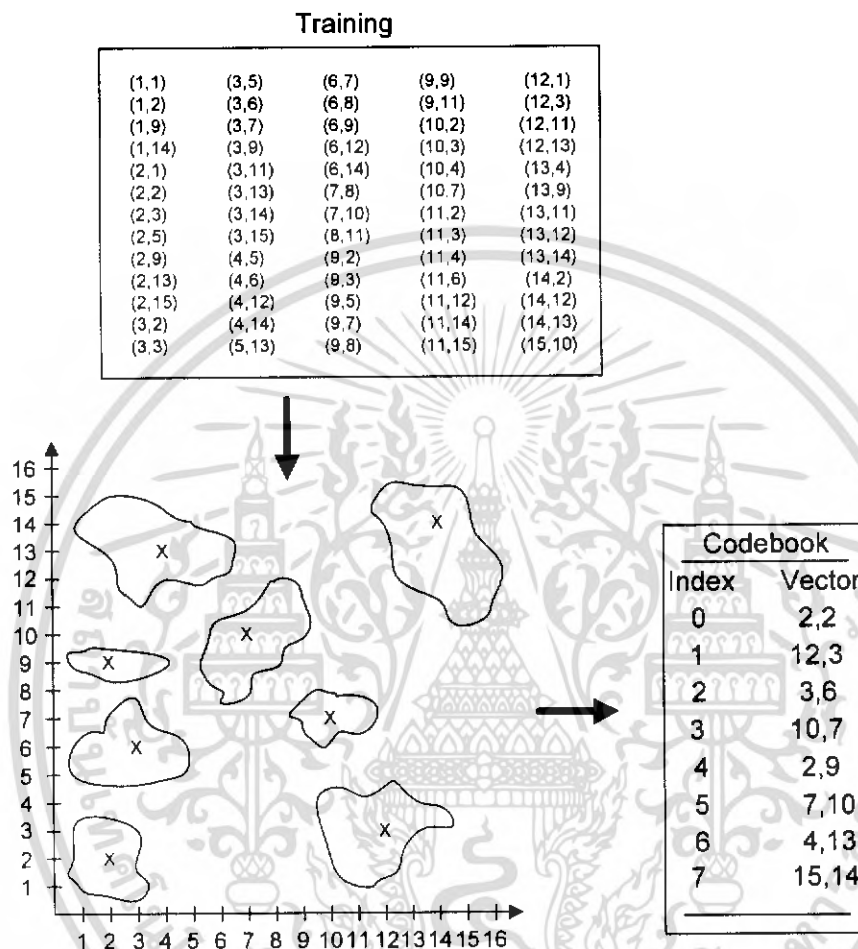


รูปที่ 2.6 การรวมกลุ่มของเฟรมเสียงเพื่อนำไปสร้าง Codebook

โดย X แทนเวกเตอร์ศูนย์กลางเพื่อแบ่งแยกเวกเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างของเวกเตอร์ควอนไทซ์เซชัน สมมติให้เวกเตอร์แต่ละตัวมี 2 มิติ และทำการหา Codebook ขนาด 8 เวกเตอร์ เวกเตอร์อินพุตทั้งหมดจะถูกจัดเข้ากลุ่มไคลด์บุดต่างๆ แล้วทำการหาจุดศูนย์กลางใหม่โดยการเฉลี่ยค่าเวกเตอร์สมาชิกทุกตัวที่อยู่ในกลุ่มเดียวกัน ผลที่ได้คือ Codebook 8 ตัวเป็นตัวแทนของเวกเตอร์อินพุตทั้งหมด

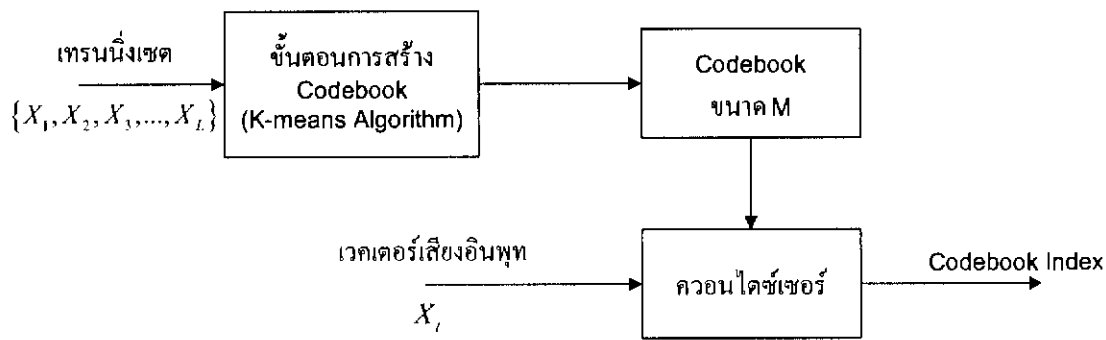


รูปที่ 2.7 แสดงตัวอย่างการหา Codebook

ข้อดีของวิธีการ เวกเตอร์ควอนไทซ์เซชันคือ

- ลดความจุในการเก็บข้อมูลที่ได้จากการวิเคราะห์หลัง
- ลดระยะเวลาในการคำนวณเปรียบเทียบความเหมือนของเวกเตอร์อินพุตที่ได้จากการวิเคราะห์ลง
- กระบวนการการเลือก Codebook ที่เหมาะสมที่สุดเพื่อนำมาแสดงแทนเวกเตอร์อินพุตที่ได้จากการวิเคราะห์เสียงพูดที่เข้ามานั้นเป็นไปด้วยความรอบคอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 บล็อกไดอะแกรมของเวกเตอร์ควอนไทซ์เซชัน

การทำงานของควอนไทซ์แบบเวกเตอร์ สามารถแบ่งเป็น 2 ขั้นตอนได้ดังนี้

1. การสร้าง Codebook โดยใช้วิธี K-mean

จากขั้นตอนการประมาณเชิงเส้น ของเสียงตัวอย่างจำนวนมาก จะได้เซตเวกเตอร์ ซึ่งประกอบด้วยเวกเตอร์สเปกตรัมจำนวน L เฟรม ; $X = \{X_i; 1 \leq i \leq L\}$ เฟรมละ P มิติ ; $X = \{X_1, X_2, \dots, X_p\}$ แล้วนำข้อมูลที่ได้นำมาทำการสร้างกลุ่มของแบบอ้างอิง

เซตเวกเตอร์นี้จะแปรตามสถานะต่างๆดังนี้

- ผู้พูด รวมทั้งช่วงของอายุ เพศ ความเร็วช้าในการพูด ระดับเสียง ฯลฯ
- สถานะแวดล้อมขณะพูด เช่น ในห้องเงียบ หรือในที่ที่มีเสียงรบกวน
- ระบบการบันทึกเสียง รวมถึงไมโครโฟนด้วย

ดังนั้นในระบบการรับรู้เสียงพูดแบบต่างบุคคล จึงจำเป็นต้องใช้แบบอ้างอิงของคำหนึ่งๆจากผู้พูดจำนวนมาก เพื่อที่จะได้ครอบคลุมถึงความแปรปรวนต่างๆ ที่เกิดขึ้นระหว่างผู้พูดแต่ละคน เนื่องจากถ้ามีการใช้แบบอ้างอิงจำนวนมาก เวลาที่ใช้ในการตอบสนองและเนื้อที่หน่วยความจำสำรองที่ใช้เก็บแบบอ้างอิงก็จะเพิ่มมากขึ้นไปด้วย และเมื่อเพิ่มแบบอ้างอิงไปจนถึงระดับหนึ่ง ความถูกต้องในการรับรู้จะเริ่มคงที่ ดังนั้นจึงมีการจัดกลุ่มของแบบอ้างอิงใหม่เพื่อให้ได้แบบอ้างอิงที่เหมาะสมที่สุด และสามารถใช้เป็นตัวแทนของแบบอ้างอิงที่มีอยู่ทั้งหมดได้ อัลกอริทึมที่ใช้ได้แก่ K-means Algorithm โดยขั้นตอนในการสร้าง Codebook มีดังนี้

1.1) นำเซตเวกเตอร์มาใช้ในการสร้าง Codebook

เซตของเวกเตอร์สเปกตรัมที่ได้จากการวิเคราะห์ $X_1, X_2, X_3, \dots, X_L$ ซึ่งเป็นเซตขนาดใหญ่จะถูกนำมาใช้ในการสร้างเซตของ Codebook เพื่อใช้เป็นตัวแทนเวกเตอร์สเปกตรัมของเสียงพูดที่สังเกตได้ ขนาด Codebook ของเวกเตอร์ควอนไทซ์เซชันคือ $M = 2^B$ เวกเตอร์ (เราเรียกว่า B-bit Codebook) ซึ่งเราต้องการให้ $L \gg M$ เพื่อที่จะหาเซตของ M-Codebook ที่ดีที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2) การสุ่มค่าเริ่มต้น

การสุ่มค่าเริ่มต้น เป็นวิธีหนึ่งในการออกแบบ Codebook ซึ่งคือการเลือกค่าเริ่มต้นของ Codebook เรียก Codebook ที่ได้จากการสุ่มค่าเริ่มต้นนี้ว่า Random Codebook ถึงแม้ว่าวิธีการนี้จะไม่ใช่วิธีที่ดีที่สุดแต่ Codebook ที่ได้จากการสุ่มก็ได้ผลเป็นที่ยอมรับ

1.3) การหาค่าความคลาดเคลื่อน

การหาค่าความคลาดเคลื่อน เป็นส่วนที่จำเป็นและเป็นประโยชน์ต่อการออกแบบ Codebook สมการทางพีชคณิตที่ใช้ในการคำนวณหาระยะทางมีหลายวิธีแต่วิธีที่นำมาใช้ก็คือการหาค่าความคลาดเคลื่อนของกำลังสองรวม (Total Square Error) ซึ่งเป็นวิธีที่ง่ายและรวดเร็ว

ถ้าสัญญาณมี P มิติ เราสามารถหาระยะห่างระหว่างสัญญาณอินพุต (x) กับเวกเตอร์โค้ด (y) โดยสมการ

$$d(v_1, v_2) = \|v_1 - v_2\|^2 = \sum_{i=0}^{k-1} (x_i - y_i)^2 \quad (2.28)$$

1.4) การจัดกลุ่ม (Classification) และการหาจุดศูนย์กลางของกลุ่ม (Center Cluster)

การจัดกลุ่มเป็นการแบ่งเวกเตอร์อินพุตเข้าไปตามกลุ่มต่างๆของ Random Codebook โดยพิจารณาระยะทางหรือความคลาดเคลื่อนน้อยที่สุดของแต่ละเวกเตอร์อินพุต x กับเวกเตอร์โค้ด y ซึ่งเป็น Codebook จากนั้นจะทำการหาค่าเฉลี่ยของแต่ละกลุ่ม เพื่อเป็นค่าศูนย์กลางของกลุ่มนั้นๆจะได้

$$\bar{Y} = \frac{1}{L} \sum_{i=1}^L x_i \quad (2.29)$$

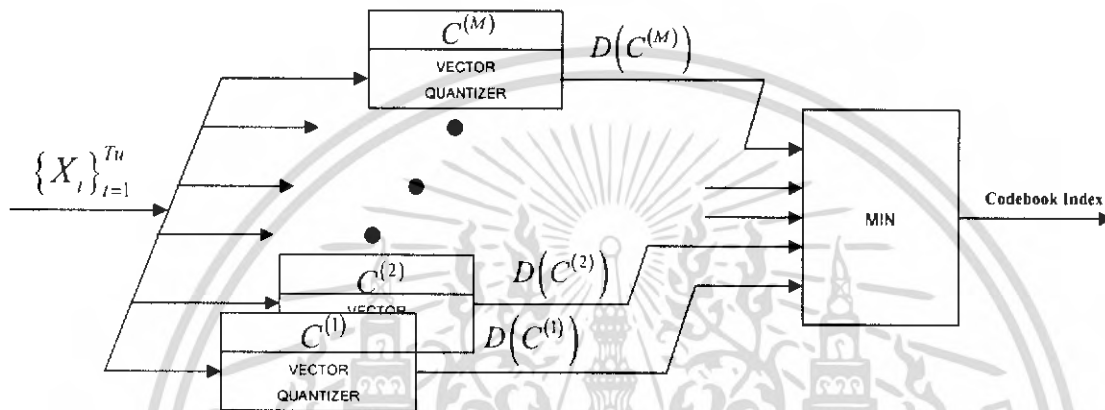
\bar{Y} เป็นจุดศูนย์กลางซึ่งเป็นเวกเตอร์ที่อยู่ตรงกลางของ $\{x_i\}_{i=1}^L$ ซึ่งแต่ละมิติจะไม่ขึ้นแก่กัน หมายความว่าแต่ละ y_k เป็นค่ากลางของ $\{x_i\}_{i=1}^L$ ทำ 2 ขั้นตอนนี้ซ้ำ จะเกิดการลู่เข้า (Convergent) โดยความคลาดเคลื่อนรวมจะต่ำกว่าค่าๆหนึ่ง ซึ่งค่าความคลาดเคลื่อนรวมจะลดลงทุกครั้งที่มีการคำนวณซ้ำใหม่ จึงขึ้นกับค่าที่กำหนดว่าจะต้องการให้ความคลาดเคลื่อนรวมน้อยเท่าไร ค่าดังกล่าวของแต่ละกลุ่มจะถูกเก็บเป็นเวกเตอร์โค้ดจะได้ว่า y เป็นควอนไทซ์ของค่า x

โดย $q(\cdot)$ เป็นโอเปอเรเตอร์ของควอนไทซ์ y ถูกเรียกว่าเอาท์พุทเวกเตอร์ของค่า x โดย y เป็นค่าใดค่าหนึ่งใน $y = \{y_i; 1 \leq i \leq M\}$ โดย $y_i = [y_{i1} \ y_{i2} \ \dots \ y_{ip}]$ Y เป็นเซตของโค้ดบุค M เป็นขนาดของโค้ดบุค และ $\{y_i\}$ เป็นเซตของเวกเตอร์โค้ด y_i อาจเรียกได้ว่าโค้ดอ้างอิง และ M อาจเรียกว่าจำนวนระดับขั้น จะทำการแบ่งเวกเตอร์ x ไปใน M เซล $\{C_i; 1 \leq i \leq M\}$ เมื่อ x อยู่ในเซล C_i

$$q(x) = y_i \text{ ถ้า } x \in C_i \quad (2.30)$$

2. ขั้นตอนการเปรียบเทียบ

เวกเตอร์ควอนไทซ์เซชันที่ใช้ในการออกแบบการรับรู้เสียงพูดนั้น จะมีจำนวนควอนไทซ์เซอร์ M ตัว นั่นหมายถึงมี M ระดับของเสียงเพื่อการรับรู้ แต่ละระดับเสียงสามารถพิจารณาจากเซตของข้อมูลเทรนนิ่ง $\{X_i^{(i)}\}$ เมื่อ $i = 1, 2, 3, \dots, M$ ซึ่ง M เป็นดัชนีระดับ โดยในแต่ละระดับจะเก็บเสียงที่อยู่ในระดับเดียวกัน เมื่อมีเสียงที่เราต้องการพิจารณา $\{X_i\}^{Tu}$ เข้ามา จะเป็นอินพุตไปยังทุกๆควอนไทซ์เซอร์ ค่าดัชนีระดับ (index) ที่เราเลือกนั้น จะเป็นระดับที่มีอัตราความผิดพลาดที่น้อยที่สุด $D(c^i)$ เมื่อ $i = 1, 2, 3, \dots, M$



รูปที่ 2.9 เวกเตอร์ควอนไทซ์เซชันที่ใช้ในระบบการรับรู้เสียงพูด

2.3.1.3 แบบจำลองมาร์คอฟ (Hidden Markov Models)

แบบจำลองมาร์คอฟ เป็นแบบจำลอง (Model) ทางสถิติ ซึ่งพัฒนาเพื่อแบ่งกลุ่มของอนุกรมทางเวลา หรือ สัญญาณที่ไม่คงที่ นั่นคือ ใช้สำหรับจัดกลุ่มของสัญญาณที่ไม่รู้จัก (Unknown signal) ให้ไปอยู่ในกลุ่มใดกลุ่มหนึ่งของสัญญาณ ซึ่งแบบจำลองมาร์คอฟ ได้ถูกนำมาประยุกต์ใช้ในการรู้จำเสียงพูด

แบบจำลองมาร์คอฟ แบ่งออกเป็น 2 ประเภท คือ แบบต่อเนื่อง (Continuous) และแบบไม่ต่อเนื่อง (Discrete-time) ซึ่งในที่นี้ จะเลือกใช้แบบไม่ต่อเนื่อง เพราะ เป็นวิธีการที่ซับซ้อนน้อยกว่า และใช้ได้ดีกับคำพูดสั้นๆ

2.3.1.3.1 ส่วนประกอบของแบบจำลอง HMM

- 1) N คือ จำนวนสเตตในแบบจำลอง โดยสามารถย้ายจากสเตตหนึ่งไปยังอีกสเตตหนึ่งได้ เราให้เซตของสเตตเป็น $\{1, 2, \dots, N\}$ และ สเตตที่เวลา t ใดๆ เป็น q_t
- 2) M คือ จำนวนของค่าปรากฏต่างๆ ต่อหนึ่งสเตต แทนด้วยสัญลักษณ์ $V = \{v_1, v_2, \dots, v_M\}$
- 3) $A = \{a_{ij}\}$ คือ ความน่าจะเป็นในการย้ายสเตต โดย $a_{ij} = P[q_t = j | q_{t-1} = i]$ เมื่อ

$$1 \leq i, j \leq N$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 4) $B = \{b_j(k)\}$ คือ ความน่าจะเป็น ของการเกิดค่าปรากฏ โดย
 $b_j(k) = P[o_t = v_k | q_t = j]$ เมื่อ $j = 1, 2, \dots, N$ และ $1 \leq k \leq M$
- 5) $\pi = \{\pi_i\}$ คือ ความน่าจะเป็นที่แต่ละสเตตจะเป็นสเตตเริ่มต้น เมื่อ $\pi_i = P[q_1 = i]$ เมื่อ $1 \leq i \leq N$

จะเห็นได้ว่า แบบจำลอง HMM ต้องการพารามิเตอร์ของแบบจำลอง 2 ตัว คือ N และ M และกลุ่มของความน่าจะเป็น A, B, π และ เพื่อความสะดวกจะใช้สัญลักษณ์เป็น (A, B, π) เพื่อแสดงเซตของพารามิเตอร์ที่สมบูรณ์ของแบบจำลอง

2.3.1.3.2 โครงสร้างของแบบจำลอง HMM

แบ่งตามลักษณะการเปลี่ยนสเตต (Transition) ของเมตริกซ์ A

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

1) HMM แบบ Egordic Model หรือ Fully Connected Model แบบจำลองนี้ทุกสเตต สามารถเปลี่ยนสเตตไปยังสเตตอื่นๆ ได้ทุกๆสเตต แสดงดังรูปที่ 2.10(a)

2) HMM แบบ Left-Right Model หรือ Bakis Model แบบจำลองนี้ การเปลี่ยนสเตตจะเปลี่ยนจากซ้ายไปขวา แสดงดังรูปที่ 2.10(b) ซึ่งมีคุณสมบัติการเปลี่ยนสเตต ดังนี้

- $a_{ij} = 0, j < i$ หมายความว่า จะไม่มีการย้ายจากสเตตที่ต่ำกว่า ไปที่สเตตปัจจุบัน
- $\pi_i = \begin{cases} 0 & ; i \neq 1 \\ 1 & ; i = 1 \end{cases}$ หมายความว่า ลำดับของสเตตต้องเริ่มต้นที่สเตต 1 ดังนั้น สเตตที่เหลือ

จึงมีความน่าจะเป็น ที่จะเป็นสเตตเริ่มต้นเท่ากับศูนย์

และ Left-Right Model นี้ จะมีกฎข้อบังคับการเปลี่ยนสเตต เพื่อไม่ให้มีการเปลี่ยนแปลงดัชนีของสเตตมากนัก นั่นคือ

$$a_{ij} = 0 \quad \text{เมื่อ } j > i + \Delta i$$

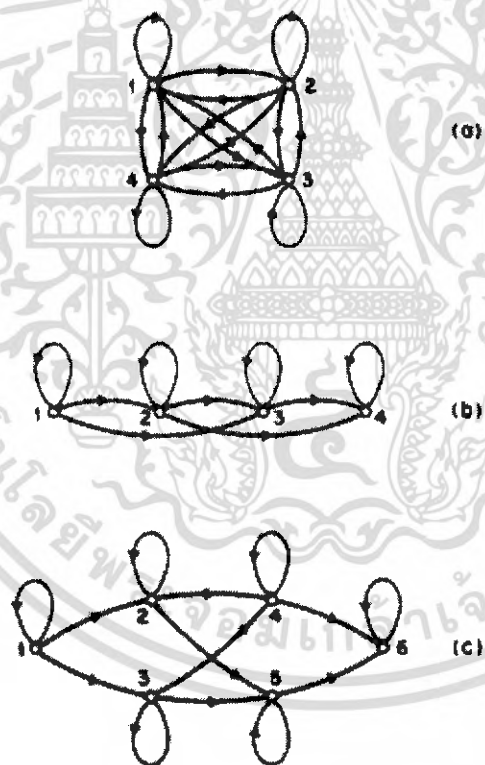
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย ค่าของ $\Delta i = 2$ หมายความว่า การเปลี่ยนสเตตจะสามารถเปลี่ยนได้เกิน 2 สเตต จะได้เมตริกซ์การเปลี่ยนสเตตเป็น

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix}$$

จะเห็นว่า สเตตสุดท้าย จะมีสัมประสิทธิ์การเปลี่ยนสเตตเป็น $a_{NN} = 1$, $a_{Ni} = 0$ เมื่อ $i < N$ แบบจำลองนี้ จึงเหมาะกับสัญญาณที่มีลักษณะการเปลี่ยนแปลงอย่างต่อเนื่อง เช่น เสียงพูด

3) HMM แบบ Parallel Left-Right Model มีคุณสมบัติการเปลี่ยนสเตตคล้ายกับแบบที่ 2 แต่มีความยืดหยุ่นมากกว่า แสดงดังรูปที่ 2.10(c)



รูปที่ 2.10 แสดง แบบจำลองต่างๆ ของ HMM

ปัญหาของแบบจำลอง HMM

ปัญหาของ HMM มี 3 ข้อ ซึ่งต้องใช้วิธีการที่วิธีต่างๆ ในการคำนวณเพื่อแก้ปัญหาคือจะกล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัญหาที่ 1

เมื่อลำดับของค่าปรากฏ $O = \{o_1, o_2, \dots, o_T\}$ และ มีแบบจำลอง $\lambda = (A, B, \pi)$ เราจะคำนวณ หาค่า $P(O|\lambda)$ ของลำดับของค่าปรากฏได้อย่างไร

การแก้ปัญหานี้ จะทำให้ เราสามารถเลือกแบบจำลอง ที่สอดคล้อง กับเหตุการณ์ ที่ดีที่สุดได้ จากแบบจำลอง หลายอันที่มีอยู่

ปัญหาที่ 2

เมื่อมีลำดับของค่าปรากฏ $O = \{o_1, o_2, \dots, o_T\}$ และ แบบจำลอง $\lambda = (A, B, \pi)$ เราจะหาลำดับ สเตท $q = \{q_1, q_2, \dots, q_T\}$ ที่เหมาะสมในการให้ค่าปรากฏนั้นได้อย่างไร

การแก้ปัญหานี้ จะช่วยในการหาลำดับของสเตทที่ถูกต้อง

ปัญหาที่ 3

จะปรับค่าพารามิเตอร์เพื่อหาแบบจำลอง $\lambda = (A, B, \pi)$ ที่ให้ค่า $P(O|\lambda)$ มากที่สุด ได้

การแก้ปัญหานี้ จะช่วยให้ได้ λ ที่เหมาะสมที่สุด เพื่ออธิบายการได้มาซึ่งลำดับเหตุการณ์ที่กำหนดให้เป็นอย่างดีที่สุด

ลำดับของค่าปรากฏที่ใช้ปรับค่าพารามิเตอร์ A, B และ π เพื่อให้ได้แบบจำลองที่ดีที่สุดนั้น เรียกว่า ลำดับเทรนนิ่ง (Training sequence)

2.3.1.3.3 การคำนวณเพื่อแก้ไขปัญหของ HMM

1) การแก้ปัญหานี้ 1 เป็นการคำนวณว่าแบบจำลอง λ ให้ความน่าจะเป็นที่จะได้ลำดับค่าปรากฏ มากน้อยเพียงใด

เมื่อเราต้องการคำนวณหาความน่าจะเป็นของลำดับค่าปรากฏ $O = \{o_1, o_2, \dots, o_T\}$ และ พิจารณาลำดับที่กำหนดไว้ $q = \{q_1, q_2, \dots, q_T\}$ เมื่อ q_1 เป็น สเตทเริ่มต้น ความน่าจะเป็นของลำดับค่าปรากฏ O ที่กำหนดไว้คือ

$$P(O|q, \lambda) = \prod_{i=1}^T P(o_i | q_i, \lambda) \quad (2.31)$$

ความน่าจะเป็นในการเกิดค่าปรากฏ คือ

$$P(O|q, \lambda) = b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots \cdot b_{q_T}(o_T) \quad (2.32)$$

ความน่าจะเป็นในการย้ายสเตท q จะเป็น

$$P(q|\lambda) = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \dots \cdot a_{q_{T-1}, q_T} \quad (2.33)$$

ดังนั้น ความน่าจะเป็นที่ O และ q จะเกิดขึ้นพร้อมกัน จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P(O|q, \lambda) = \sum_{\text{all } q} P(O|q, \lambda) P(q|\lambda) \quad (2.34)$$

แต่เนื่องจากสมการดังกล่าว มีการคำนวณที่ยุ่งยาก ดังนั้น เราจึง มีวิธีการเพื่อช่วยแก้ปัญหาให้ง่ายขึ้น โดยใช้กระบวนการต่อไปนี้

1.1) กระบวนการไปข้างหน้า (Forward Procedure)

เมื่อกำหนดให้ Forward variable เป็น $\alpha_t(i) = P(o_1, o_2, \dots, o_t, q_t = i | \lambda)$ หมายถึง ความน่าจะเป็นของการเกิดลำดับค่าปรากฏ o_1, o_2, \dots, o_t ที่จะอยู่ที่สแตต i ณ เวลา t โดยมีแบบจำลอง เป็น λ โดยสามารถหา $\alpha_t(i)$ ได้ดังนี้

- การเริ่มต้น (Initialization)

$$\alpha_t(i) = \pi_i b_i(o_1) \quad \text{เมื่อ } 1 \leq i \leq N \quad (2.35)$$

เมื่อกำหนด ที่เวลาเริ่มต้น $t=1$ และ เหตุการณ์เริ่มต้น o_1

- การเหนี่ยวนำ (Induction)

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^N \alpha_t(i) a_{ij} \right] b_j(o_{t+1}) \quad \text{เมื่อ } 1 \leq t \leq T-1 \quad \text{และ } 1 \leq j \leq N$$

(2.36)

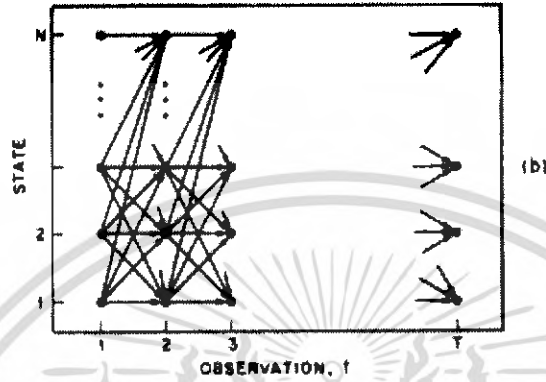
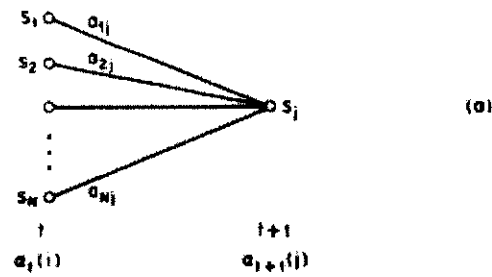
หมายถึง ความน่าจะเป็นของสแตต j ที่เวลา $t+1$ ที่มาจากสแตต i ที่เป็นไปได้ถึง N สแตต ที่เวลา t ดังรูปที่ 2.11

- การสิ้นสุด (Termination)

ความน่าจะเป็นของลำดับค่าปรากฏ O ได้จากผลรวมของ $\alpha_t(i)$ จากทุกๆ สแตต

$$P(O|\lambda) = \sum_{i=1}^N \alpha_t(i) \quad \text{เมื่อ } 1 \leq j \leq N \quad (2.37)$$

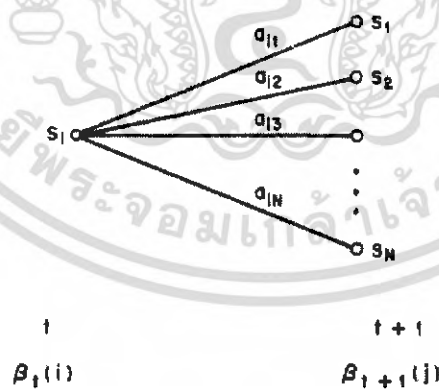
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 แสดงกระบวนการ ไปข้างหน้า

1.2) กระบวนการย้อนกลับ (Backward Procedure)

เมื่อกำหนดให้ Backward variable $\beta_T(i) = P(o_{t+1}, o_{t+2}, \dots, o_T, q_t = i | \lambda)$ หมายถึง ความน่าจะเป็นของลำดับค่าปรากฏส่วนหลัง จากเวลา $t + 1$ ไปจนจบ โดยการกำหนดว่าต้องอยู่ที่สถานะ i ที่เวลา t และมีแบบจำลองเป็น λ เราคำนวณหา $\beta_T(i)$ ได้ดังนี้



รูปที่ 2.12 แสดง กระบวนการ ไปข้างหลัง

- การเริ่มต้น (Initialization)

$$\beta_T(i) = 1 \text{ เมื่อ } 1 \leq i \leq N \tag{2.38}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การเหนี่ยวนำ (Induction)

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

เมื่อ $t = T-1, T-2, \dots, 1$ และ $1 \leq i \leq N$

2) การแก้ปัญหาที่ 2 เพื่อหาลำดับสแตทที่เหมาะสม

เราจะใช้วิธีวิเทอริบี อัลกอริทึม (Viterbi Algorithm) เพื่อหาลำดับสแตทที่ดีที่สุด ณ เวลา t หนึ่งๆ เมื่อ กำหนดลำดับเหตุการณ์ $O = (o_1, o_2, \dots, o_t)$ โดยนิยามให้

$$\delta_t(i) = \max_{q_1, q_2, \dots, q_{t-1}} P[q_1, q_2, \dots, q_{t-1}, q_t = i, o_1, o_2, \dots, o_t | \lambda] \quad (2.40)$$

$\delta_t(i)$ หมายถึง ความน่าจะเป็นสูงสุดของเส้นทาง (Path) ณ เวลา t ซึ่งเริ่มนับจากเหตุการณ์ที่เวลาเริ่มต้น จนถึงเวลา t ที่สแตท i และ โดยอาศัยการคูณสมบัติการเหนี่ยวนำ (Induction) เราจะได้

$$\delta_{t+1}(i) = \left[\max_j \delta_t(j) a_{ij} \right] b_j(o_{t+1}) \quad (2.41)$$

เราสามารถหาลำดับสแตทที่ดีที่สุดได้ โดยใช้กระบวนการต่อไปนี้ เมื่อกำหนดให้ $\psi_t(i)$ เป็นอาร์เรย์ (Array)

- การเริ่มต้น (Initialization)

$$\delta_t(i) = \pi_i b_i(o_1) \quad \text{เมื่อ } 1 \leq i \leq N \quad (2.42)$$

$$\psi_t(i) = 0 \quad (2.43)$$

- การย้อนกลับ (Recursion)

$$\delta_t(j) = \left[\max_{1 \leq i \leq N} \delta_{t-1}(i) a_{ij} \right] b_j(o_t) \quad (2.44)$$

เมื่อ $2 \leq t \leq T$ และ $1 \leq j \leq N$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{ij}] \quad (2.45)$$

เมื่อ $2 \leq t \leq T$ และ $1 \leq j \leq N$

- การสิ้นสุด (Termination)

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.46)$$

$$q_T^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.47)$$

- เส้นทางเดินย้อนกลับ (Path backtracking)

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \text{ เมื่อ } t = T-1, T-2, \dots, 1 \quad (2.48)$$

3) การแก้ปัญหาที่ 3 เพื่อหาโมเดลที่จะให้ผลตามลำดับค่าปรากฏหนึ่งๆ โดยเลือกค่าพารามิเตอร์ A, B, π ที่ดีที่สุด โดยใช้ กระบวนการทำซ้ำ (Iterative) วิธีที่เราเลือกใช้คือ วิธี บาม-เวลช์ (Baum-Welch) หรือ EM (Expectation-Maximization)

เมื่อนิยามให้

$$3.1) \gamma_t(i) = P(q_t = i | O, \lambda) \quad (2.49)$$

หมายถึง ความน่าจะเป็นที่จะอยู่ที่สแตท i ณ เวลา t โดยกำหนดลำดับเหตุการณ์ O และแบบจำลอง λ ให้สามารถแสดงค่า $\gamma_t(i)$ ได้ดังนี้

$$\begin{aligned} \gamma_t(i) &= \frac{P(O, q_t = i | \lambda)}{P(O | \lambda)} \\ &= \frac{P(O, q_t = i | \lambda)}{\sum_{i=1}^N P(O, q_t = i | \lambda)} \end{aligned} \quad (2.50)$$

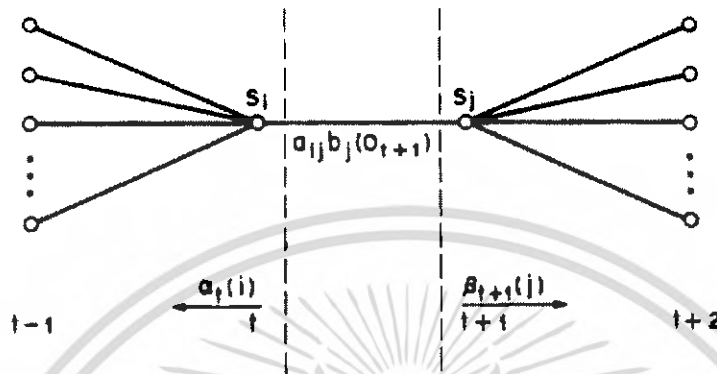
เนื่องจาก $P(O, q_t = i | \lambda)$ มีค่าเท่ากับ $\alpha_t(i)\beta_t(i)$ จึงได้

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (2.51)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$3.2) \mathcal{E}_t(i, j) = P(q_t = i, q_{t+1} = j | O, \lambda) \quad (2.52)$$

หมายถึง ความน่าจะเป็นที่จะอยู่ที่สแตต i ที่เวลา t และ สแตต j ที่เวลา $t+1$ เมื่อ กำหนดแบบจำลอง และลำดับค่าปรากฏให้ ดังแสดงดังรูปที่ 2.13



รูปที่ 2.13 แสดง ค่าปรากฏที่จะอยู่ที่สแตต i ที่เวลา t โดยคำนึงถึงลำดับค่าปรากฏจากเวลา $t+1$ ซึ่งต้องพิจารณา สแตต j ที่จะเป็นไปได้ทั้งหมด ณ เวลาที่ $t+1$ โดยจะขึ้นอยู่กับค่า a_{ij} และ $b_j(o_{t+1})$

ซึ่งจากนิยามของตัวแปรไปข้างหน้า และ ตัวแปรย้อนกลับ สามารถนำมาสัมพันธ์กับ $\mathcal{E}_t(i, j)$ ได้ดังนี้

$$\begin{aligned} \mathcal{E}_t(i, j) &= \frac{P(q_t = i, q_{t+1} = j, O | \lambda)}{P(O | \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O | \lambda)} \quad (2.53) \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned}$$

และ จะได้ความสัมพันธ์ของ $\gamma_t(i)$ และ $\mathcal{E}_t(i, j)$ ดังนี้

$$\gamma_t(i) = \sum_{j=1}^N \mathcal{E}_t(i, j) \quad (2.54)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ

$\sum_{i=1}^{T-1} \gamma_i(i)$ คือ จำนวนของการเปลี่ยนสเตตออกจากสเตต i ในลำดับค่าปรากฏ O

$\sum_{i=1}^{T-1} \mathcal{E}_i(i, j)$ คือ จำนวนของการเปลี่ยนสเตตจากสเตต i ไปสเตต j ในลำดับค่าปรากฏ O

ดังนั้น สามารถหาค่าพารามิเตอร์ได้ ดังนี้

$$\pi'_i = \gamma_i(i) \quad \text{เมื่อ } 1 \leq i \leq N \quad (2.55)$$

โดย π'_i คือ จำนวนครั้งในการอยู่ที่สเตต i ไปสเตต j

$$a'_{ij} = \frac{\sum_{i=1, o_i=v_k}^T \gamma_i(j)}{\sum_{i=1}^{T-1} \gamma_i(i)} \quad (2.56)$$

โดย a'_{ij} คือ จำนวนครั้งที่คาดไว้ของการย้ายจากสเตต i ไปสเตต j

จำนวนครั้งที่คาดว่าจะย้ายจากสเตต i

$$b'_j(k) = \frac{\sum_{i=1, o_i=v_k}^T \gamma_i(j)}{\sum_{i=1}^T \gamma_i(i)} \quad (2.57)$$

โดย $b'_j(k)$ คือ จำนวนครั้งที่คาดว่าจะอยู่ในสเตต j และเกิดเหตุการณ์ v_k

จำนวนครั้งที่คาดว่าจะอยู่ที่สเตต j

จากกระบวนการข้างต้น ถ้าเราจะคำนวณซ้ำๆ โดยให้ $\lambda = (A, B, \pi)$ เป็น แบบจำลองปัจจุบัน จะได้ $\lambda' = (A', B', \pi')$ จะถูกแทนที่ $\lambda = (A, B, \pi)$ ซึ่งเป็นแบบจำลองเริ่มต้นแล้ว จะทำให้ความน่าจะเป็นของการเกิดลำดับค่าปรากฏ O ดีขึ้น จนกระทั่งถึงจุดวิกฤต จึงหยุด ซึ่งเราจะได้จุดวิกฤตของฟังก์ชันความน่าจะเป็นในกรณีที่ $\lambda' = \lambda$ หรือ ถ้า λ' มีความน่าจะเป็นมากกว่าแบบจำลอง λ ในลักษณะที่ $P(O|\lambda') > P(O|\lambda)$ นั่นก็คือ เราจะได้ แบบจำลอง λ' ใหม่ ที่น่าจะ ทำให้เกิดลำดับค่าปรากฏ O ได้ดีกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.3.4 การปรับค่าพารามิเตอร์ของ HMM

1) การสเกลลิง (Scaling)

เนื่องจาก $\alpha_t(i)$ จะประกอบด้วยผลรวมของเทอมจำนวนมาก ซึ่งก็คือ

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

และเนื่องจากแต่ละเทอมของ a และ b มีค่าน้อยกว่า 1 อยู่แล้ว เมื่อพิจารณาผลรวมของการคูณค่าที่น้อยลงเรื่อยๆ แสดงว่า เมื่อ t มากขึ้น แต่ละเทอมของ $\alpha_t(i)$ จะเข้าสู่ศูนย์ ทำให้ Dynamic Range ของการคำนวณ $\alpha_t(i)$ เกิน Range ของคอมพิวเตอร์ ทำให้ค่าที่ได้ไม่ถูกต้อง จึงได้มีการสเกลลิงขึ้น เพื่อให้ $\alpha_t(i)$ อยู่ภายใน Dynamic Range ของคอมพิวเตอร์ ($1 \leq t \leq T$)

การสเกลลิงทำได้โดยการคูณ $\alpha_t(i)$ ด้วยสัมประสิทธิ์การสเกลลิง ซึ่งค่าสัมประสิทธิ์นี้ ไม่ขึ้นกับ i การสเกลลิง $\beta_t(i)$ ก็เช่นเดียวกัน หลังการคำนวณค่าการสเกลลิง ก็จะตัดกันหมดไปเอง พิจารณาจากสมการ

$$\hat{a}_{ij} = \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{t=1}^T \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad (2.58)$$

เมื่อเราให้ $\alpha_t(i)$ แทน α ที่ยังไม่ได้สเกลลิง

$\hat{\alpha}_t(i)$ แทน α ที่สเกลลิงแล้ว

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

เมื่อ $t=1$ จะได้ $\hat{\alpha}_1(i) = c_1 \alpha_1(i)$

$$\text{เมื่อ} \quad c_1 = \frac{1}{\sum_{i=1}^N \alpha_1(i)} \quad (2.59)$$

เมื่อ $2 \leq t \leq T$ คำนวณ $\hat{\alpha}_t(i)$ จาก สมการ(2.58) ในเทอมของ $\hat{\alpha}_t(i)$ ค่าก่อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\hat{\alpha}_i(i) = \sum_{j=1}^N \hat{\alpha}_{i-1}(j) a_{ji} b_i(O_i) \quad (2.60)$$

เมื่อ สัมประสิทธิ์การสเกลลิง (c_i) เป็น

$$c_i = \frac{1}{\sum_{i=1}^N \hat{\alpha}_i(i)} \quad (2.61)$$

เมื่อให้ $\hat{\alpha}_i(i) = c_i \alpha_i(i)$ ดังนั้น เราจะได้ว่า

$$\hat{\alpha}_i(i) = \frac{\sum_{j=1}^N \hat{\alpha}_{i-1}(j) a_{ji} b_j(O_i)}{\sum_{i=1}^N \sum_{j=1}^N \hat{\alpha}_{i-1}(j) a_{ji} b_j(O_i)} \quad (2.62)$$

และ โดยการเหนี่ยวนำ จะได้

$$\hat{\alpha}_{i-1}(j) = \prod_{\tau=1}^{i-1} c_\tau \alpha_{i-1}(j) \quad (2.63)$$

จะได้ว่า

$$\begin{aligned} \hat{\alpha}_i(i) &= \frac{\sum_{j=1}^N \alpha_{i-1}(j) \prod_{\tau=1}^{i-1} c_\tau a_{ji} b_i(O_i)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_{i-1}(j) \prod_{\tau=1}^{i-1} c_\tau a_{ji} b_i(O_i)} \\ &= \frac{\alpha_i(i)}{\sum_{i=1}^N \alpha_i(i)} \end{aligned} \quad (2.64)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นคือ จะสเกล $\alpha_t(i)$ ได้ โดยหารด้วยผลรวมของ $\alpha_t(i)$ ทั้งหมด และสเกล $\beta_t(i)$ ด้วยค่าเดียวกันนี้ ในเทอมของการสเกลนี้ สมการ 2.58 จะได้เป็น

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(o_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N \hat{\alpha}_t(i) a_{ij} b_j(o_{t+1}) \hat{\beta}_{t+1}(j)} \quad (2.65)$$

โดยแต่ละ $\hat{\alpha}_t(i)$ และ $\hat{\beta}_{t+1}(j)$ จะได้เป็น

$$\hat{\alpha}_t(i) = \left[\prod_{s=1}^t c_s \right] \alpha_t(i) = c_t \alpha_t(i) \quad (2.66)$$

$$\hat{\beta}_{t+1}(j) = \left[\prod_{s=t+1}^T c_s \right] \beta_{t+1}(j) = d_{t+1} \beta_{t+1}(j) \quad (2.67)$$

ดังนั้น สมการที่ (2.65) จะเขียนได้เป็น

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} c_t \alpha_t(i) a_{ij} b_j(o_{t+1}) d_{t+1} \beta_{t+1}(j)}{\sum_{t=1}^{T-1} \sum_{j=1}^N c_t \alpha_t(i) a_{ij} b_j(o_{t+1}) d_{t+1} \beta_{t+1}(j)} \quad (2.68)$$

ซึ่งเทอม $C_t D_{t+1}$ จะเขียนได้ในเทอม

$$\begin{aligned} C_t D_{t+1} &= \prod_{s=1}^t c_s \prod_{s=t+1}^T c_s \\ &= \prod_{s=1}^T c_s \\ &= C_T \end{aligned} \quad (2.69)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งไม่ขึ้นกับเวลา t ดังนั้น $C_t D_{t+1}$ จะถูกตัดทิ้ง ทั้งเศษและส่วนของสมการ (2.68) ซึ่งทำให้ได้
สูตรการคำนวณซ้ำๆ (Reestimate) เดิมกลับคืนมา

กระบวนการสเกลลิง ดังกล่าวนี้ สามารถนำไปใช้ได้กับสัมประสิทธิ์ β และ π ในการสเกล
ลิงนี้ จะทำให้การคำนวณค่า $P(O|\lambda)$ เปลี่ยนไป เราจะไม่สามารถหาได้จากการรวมกับเทอม $\hat{\alpha}_T(i)$
แต่ จะหาจากคุณสมบัติ

$$\prod_{t=1}^T c_t \sum_{i=1}^N \alpha_T(i) = c_T \sum_{i=1}^N \alpha_T(i) = 1 \quad (2.70)$$

ดังนั้นจะได้

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

$$P(O|\lambda) = \frac{1}{\prod_{t=1}^T c_t} \quad (2.71)$$

ทำให้อยู่ในรูป \log ของ P เพื่อไม่ให้เกิน Dynamic Range ของคอมพิวเตอร์

$$\log[P(O|\lambda)] = \sum_{t=1}^T \log c_t \quad (2.72)$$

2.3.1.3.5 ลำดับของค่าปรากฏหลายเหตุการณ์ (Multiple Observation Sequence)

ในการใช้แบบจำลองแบบ Left-Right นั้น การแทนแบบจำลองต้องใช้หลายๆ เหตุการณ์ของ
ลำดับค่าปรากฏเข้ามาแทน เพื่อให้ได้ค่าพารามิเตอร์ที่ถูกต้องมากขึ้น

ถ้าให้เซตของ v ลำดับค่าปรากฏเป็น

$$O = [O^{(1)}, O^{(2)}, \dots, O^{(k)}]$$

เมื่อ $O^{(v)} = O_1^{(v)}, O_2^{(v)}, \dots, O_T^{(v)}$ เป็นลำดับค่าปรากฏของเหตุการณ์ที่ v โดยให้แต่ละ
เหตุการณ์ เป็นอิสระต่อกัน เราจะปรับค่าพารามิเตอร์ของ λ ให้ดีที่สุด เราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 P(O|\lambda) &= \prod_{v=1}^V P(O^{(v)}|\lambda) \\
 &= \prod_{v=1}^V P_v
 \end{aligned}
 \tag{2.73}$$

เนื่องจากสูตรการคำนวณซ้ำๆ มาจากพื้นฐานของความน่าจะเป็นของการเกิดเหตุการณ์ต่างๆ สูตรการคำนวณซ้ำของลำดับเหตุการณ์หลายๆ ลำดับ ก็จะสามารถนำเอาความถี่ของการเกิดเหตุการณ์ของการเกิดค่าปรากฏต่างๆ แต่ละเหตุการณ์มารวมกัน จะได้สูตรการหา a_{ij} และ $b_j(k)$ เป็น

$$\bar{a}_{ij} = \frac{\sum_{v=1}^V \frac{1}{P_v} \sum_{t=1}^{T_v-1} \alpha_t^v(i) a_{ij} b_j(O_{t+1}^v) \beta_{t+1}^v(j)}{\sum_{v=1}^V \frac{1}{P_v} \sum_{t=1}^{T_v-1} \alpha_t^v(i) \beta_t^v(i)}
 \tag{2.74}$$

$$\bar{b}_i(k) = \frac{\sum_{v=1}^V \frac{1}{P_v} \sum_{t=1, O_t=k}^{T_v-1} \alpha_t^v(i) \beta_t^v(i)}{\sum_{v=1}^V \frac{1}{P_v} \sum_{t=1}^{T_v-1} \alpha_t^v(i) \beta_t^v(i)}
 \tag{2.75}$$

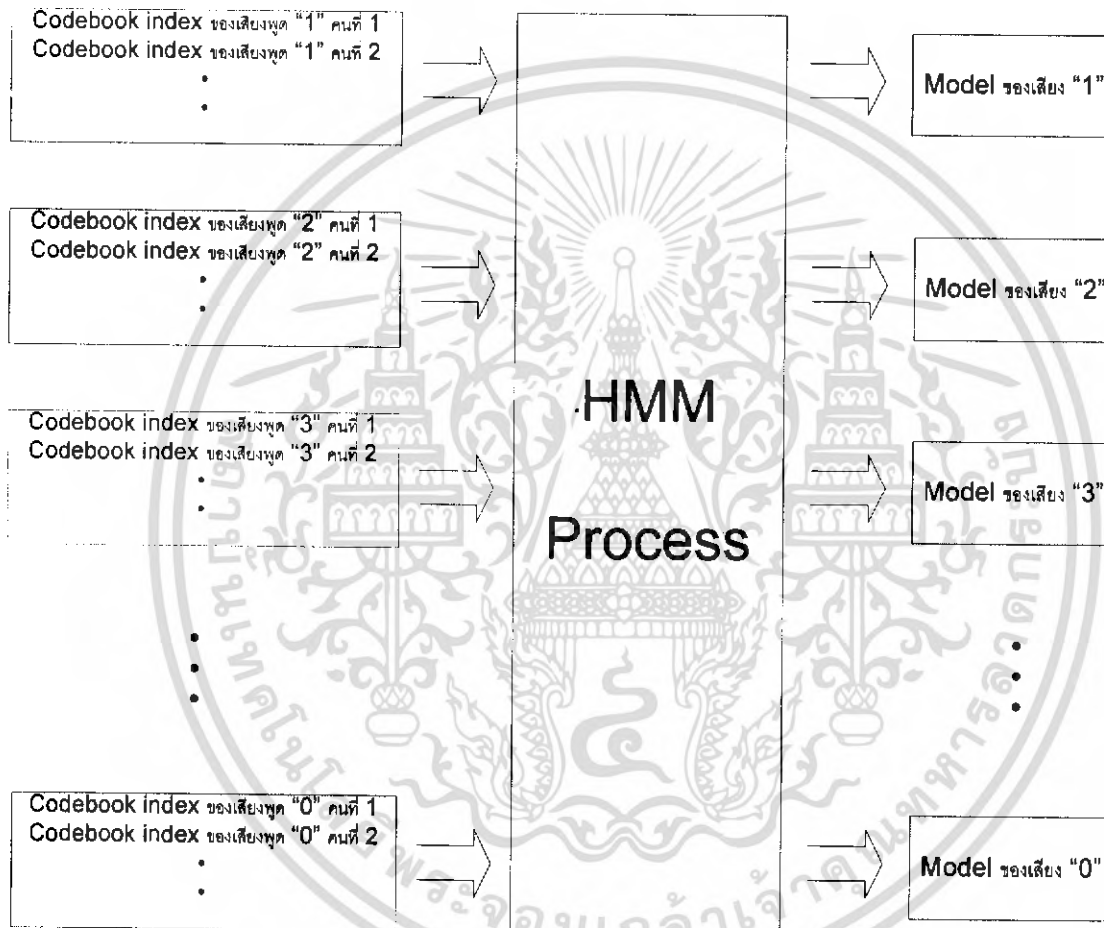
ส่วน π_i ไม่ต้องคำนวณ เนื่องจาก $\pi_1 = 1$ และ $\pi_i = 0$ เมื่อ $i \neq 1$
การสเกลลิ่งที่เหมาะสมของสมการ (2.74)-(2.75)

$$\bar{a}_{ij} = \frac{\sum_{v=1}^V \sum_{t=1}^{T_v-1} \hat{\alpha}_t^v a_{ij} b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)}{\sum_{v=1}^V \sum_{t=1}^{T_v-1} \sum_{i=1}^N \hat{\alpha}_t^v(i) a_{ij} b_j(O_{t+1}^v) \hat{\beta}_{t+1}^v(j)}
 \tag{2.76}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\bar{b}_i(k) = \frac{\sum_{v=1}^V \sum_{t=1, O_t=k}^{T_v-1} \hat{\alpha}^v a_{ij} b_j(o_{t+1}^v) \beta_t^v(j)}{\sum_{v=1}^V \sum_{t=1}^{T_v-1} \sum_{i=1}^N \hat{\alpha}^v(i) a_{ij} b_j(o_{t+1}^v) \beta_t^v(j)} \quad (2.77)$$

2.3.1.3.6 ขั้นตอนการสร้างโมเดล



รูปที่ 2.14 แสดง ขั้นตอนการสร้างโมเดล

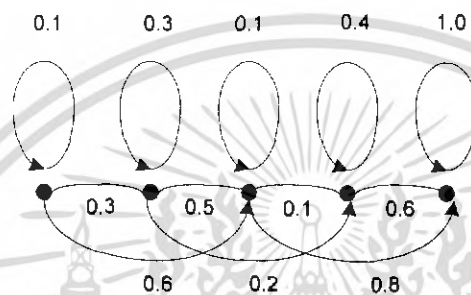
การสร้างโมเดลโดยวิธี HMM นั้นเป็นการนำทฤษฎีความน่าจะเป็นมาอธิบายการเกิดของตัวแปร 2 ตัว คือ สถานะ หรือ State ซึ่งจะเก็บข้อมูลการเปล่งเสียงส่วนหนึ่งของพยางค์ที่เวลาต่างๆ ไว้ และ ค่าปรากฏ คือ Sequence ของค่า Index ของเสียงที่ได้จากขั้นตอน VQ ที่เวลาต่างๆ ผู้สังเกตจะเห็นเพียงค่าปรากฏของแต่ละสแตท แต่จะไม่ทราบแน่ชัดว่าอยู่ที่สแตทใด จึงเรียกว่า Hidden Markov Models อาจเปรียบเทียบกับ การโยนเหรียญ สมมุติว่าโยนเหรียญ 3 เหรียญ แล้วออก "ก้อย" 1 เหรียญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในที่นี้เหรียญ 3 เหรียญ ก็คือ จำนวนสแตท 3 สแตท และ ค่าปรากฏก็คือ “ก้อย” เราทราบเพียงว่า ออก “ก้อย” แต่ไม่ทราบแน่ชัดว่าเหรียญใดแน่ที่ออก “ก้อย”

การใช้วิธี HMM สามารถบอกได้ว่าค่าปรากฏนั้นอยู่ที่สแตทใด โดยการใช้เทคนิคความน่าจะเป็นเข้ามาช่วย โมเดลที่สร้างจาก HMM จะประกอบด้วยข้อมูลของสถานะต่างๆ ที่เชื่อมเข้าหากัน โดยใช้เส้นที่หมายถึงการเปลี่ยนแปลงสถานะของข้อมูลความน่าจะเป็น เพื่อใช้บอกว่าสถานะใดจะเกิดต่อจากสถานะปัจจุบัน

ชนิดของโมเดลที่ใช้ เป็นแบบ Left-Right Model ซึ่งมีลักษณะการย้ายสแตทจากซ้ายไปขวา และที่สแตทใดๆ จะสามารถย้ายไปสแตทถัดไปได้มากที่สุด 2 สแตท ดังแสดงในรูปที่ 2.15



รูปที่ 2.15 แสดงโมเดลของเสียงๆ หนึ่ง

จากรูปตัวอย่างโมเดลของเสียงนี้ จะประกอบด้วยสแตท 5 สแตท แต่ละเส้นทางของการย้ายสแตท จะกำหนดด้วยค่าความน่าจะเป็นค่าหนึ่ง ซึ่งแต่ละโมเดลจะมีค่าต่างกัน ที่สแตทใดๆ ค่าความน่าจะเป็นรวมของการย้ายสแตทจะเท่ากับ 1

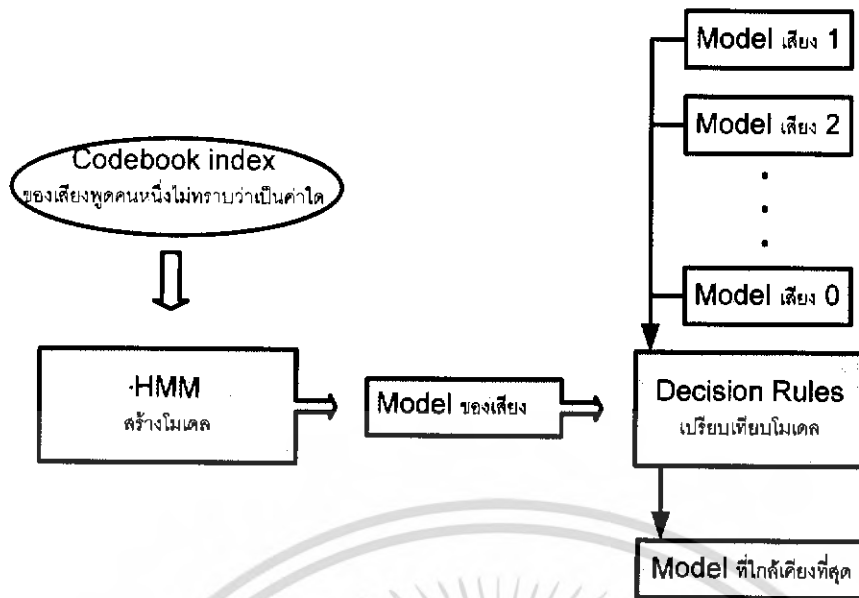
2.3.1.4 Decision rule

สิ่งที่ได้กล่าวไว้ในส่วนการแก้ปัญหาข้อที่ 2 ของ HMM โดยใช้วิธีตาม Viterbi Algorithm

2.3.1.4.1 Viterbi Algorithm

ใช้ในการหาพยางค์ที่สั้นที่สุด จากระยะทางที่เป็นไปได้ ของ โมเดลที่มีอยู่ กับ โมเดลของเสียงที่เข้ามา โดยเลือกโมเดลที่มีความเป็นไปได้ในการเกิดเหตุการณ์ ถ้าโมเดลของคำใดมีความน่าจะเป็นสูงกว่าโมเดลของคำอื่นๆ จะแสดงว่าคำที่ไม่ทราบว่าเป็นเสียงใด ก็คือ คำนั้นนั่นเอง

ขั้นการเปรียบเทียบโมเดลเป็นขั้นที่น่าเสียง Input ที่ไม่ทราบว่าเป็นคำใด เข้ามาผ่านกระบวนการสร้างโมเดลเฉพาะของเสียงนั้น แล้วนำโมเดลที่ได้นี้ ไปเปรียบเทียบกับโมเดลของเสียงทุกเสียงที่ได้เก็บไว้แล้ว

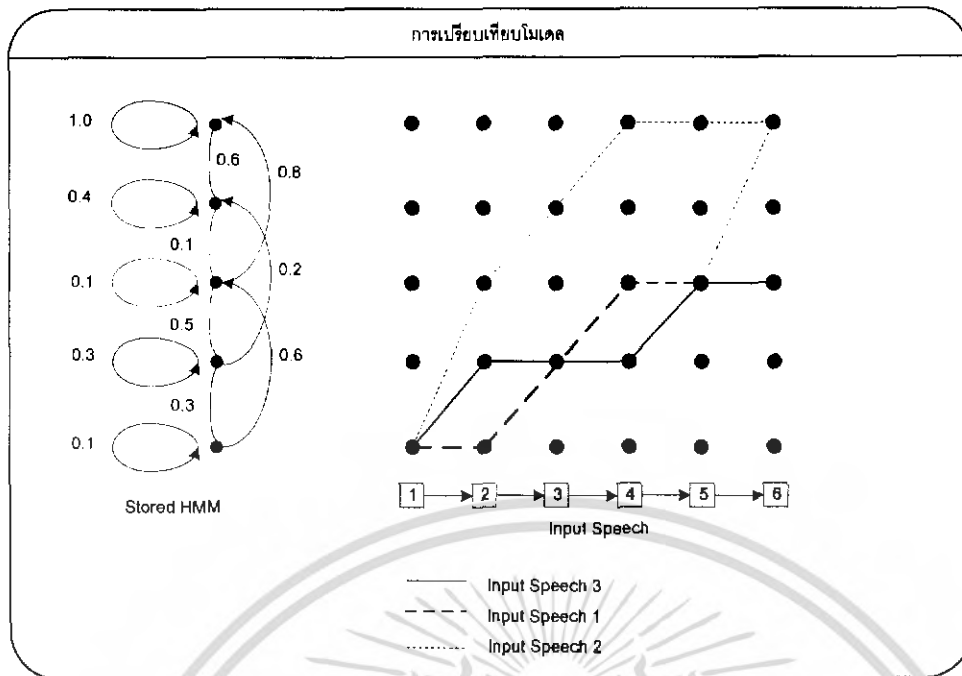


รูปที่ 2.16 แสดง ขั้นตอนการตัดสินใจ

การเปรียบเทียบโมเดลนั้น จะใช้วิธี Viterbi Algorithm (VA) ซึ่งจะเลือกเส้นทางที่สั้นที่สุดของโมเดลเสียงที่ไม่ทราบ กับโมเดลที่เก็บไว้ทุกตัว ผลของการเลือกจะทำให้ทราบว่าเสียงนั้นเป็นเสียงใด ความสั้นยาวของเส้นทางที่กล่าวถึง ก็คือ ความน่าจะเป็นของการเปลี่ยนสแตตนั้นเอง ค่าปรากฏที่เกิดจากโมเดลของเสียงที่ไม่ทราบค่าจะเป็นตัวกำหนดสแตตที่เกิดขึ้นในแต่ละช่วงเวลา การตัดสินใจว่าเสียงนั้นตรงกับโมเดลใด ได้มาจากการเปรียบเทียบค่าที่น้อยที่สุดของ “ความยาวรวมของเส้นทางที่ได้จากสแตตทั้งหมด” ที่เปรียบเทียบกับแต่ละโมเดล

ตัวอย่าง

มี Input Speech 3 ตัว เทียบกับแบบจำลอง HMM 1 ตัว Input ตัวที่มีระยะทางห่างจากแบบจำลองนี้ที่น้อยที่สุด ก็จะถือว่าเป็นคำนี้



รูปที่ 2.17 แสดง การเปรียบเทียบ โมเดล

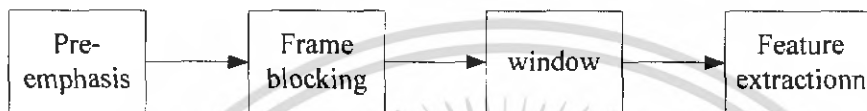
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การคำนวณและการออกแบบ

3.1 การวิเคราะห์เสียงเบื้องต้น

ในการวิเคราะห์เสียงเพื่อนำไปทำแบบจำลอง (Model) จำเป็นต้องทำการวิเคราะห์เสียงเบื้องต้นก่อน เพื่อให้ได้ค่าตัวอย่างของแต่ละเสียง ที่สามารถเป็นตัวแทนของเสียงนั้นได้ ทำให้การวิเคราะห์ขั้นต่อไปมีความสะดวก ซึ่งขั้นตอนในการวิเคราะห์เสียงเบื้องต้น สามารถเขียนเป็น Block diagram ได้ ดังนี้

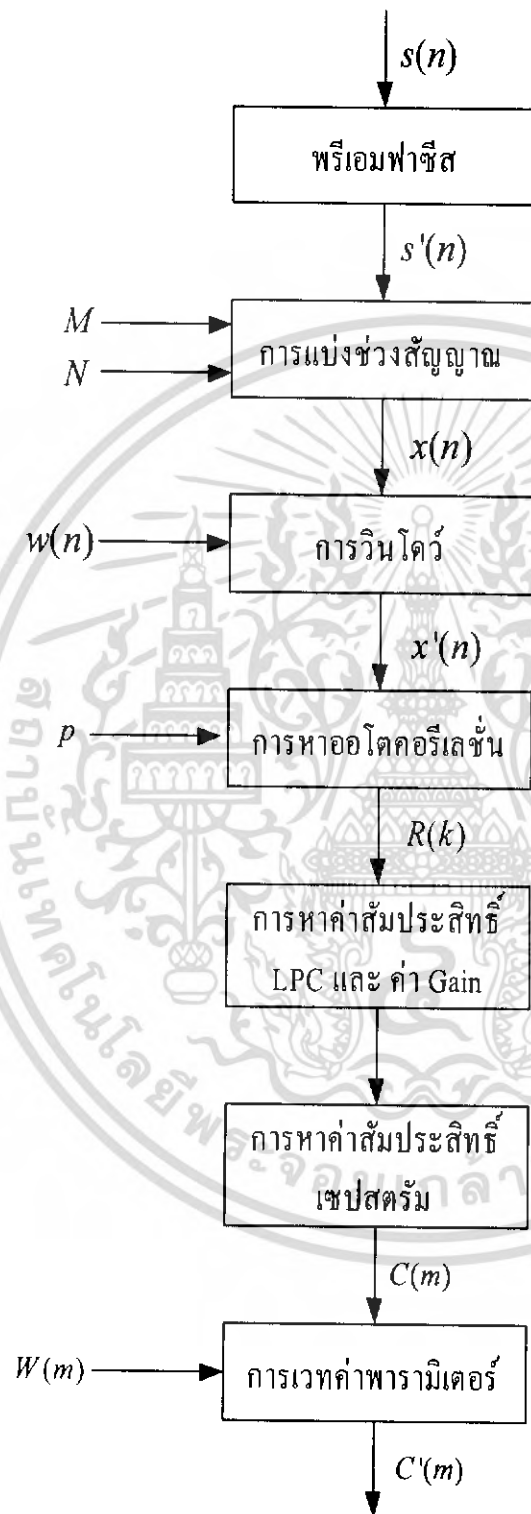


รูปที่ 3.1 แสดง Block diagram ของการวิเคราะห์เสียงเบื้องต้น

หรือ อาจจะแบ่งเป็นขั้นตอนในการเตรียมสัญญาณเข้าได้เป็นดังรูป 3.2

การวิเคราะห์สามารถแบ่งเป็นขั้นตอนต่างๆ ดังนี้

- 3.1 การพรีเอมฟาสซิส (Pre-emphasis)
- 3.2 การแบ่งช่วงสัญญาณ (Frame blocking)
- 3.3 การวินโดว์ (Windowing)
- 3.4 การหาค่าออโตคอร์เรเลชัน (Autocorrelation analysis)
- 3.5 การหาค่าสัมประสิทธิ์ LPC
- 3.6 การหาค่าสัมประสิทธิ์เซปสตรัม (Cepstrum)
- 3.7 การเวทค่าพารามิเตอร์ (Parameter weighting)



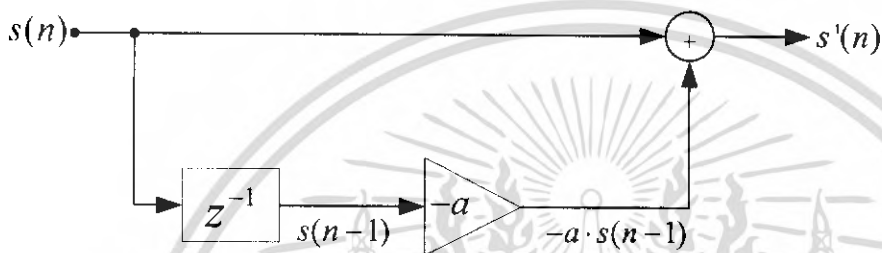
รูปที่ 3.2 แสดง ขั้นตอนการเตรียมสัญญาณในการวิเคราะห์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การพรีเอมฟาซิส (Pre-emphasis)

เนื่องจากสัญญาณเสียงพูดของมนุษย์ จะมีองค์ประกอบส่วนใหญ่อยู่บริเวณความถี่ต่ำ เมื่อเทียบกับแถบความถี่ที่ปฏิบัติงาน (bandwidth) ไม่เกิน 5 กิโลเฮิร์ตซ์ เพื่อให้อัตราส่วนสัญญาณเสียงต่อสัญญาณรบกวน (signal to noise ratio:SNR) มีค่าค่อนข้างคงที่ ตลอดช่วงความถี่ที่ปฏิบัติงานนี้ เราจึงต้องมีการพรีเอมฟาซิส โดยเน้นที่ความถี่สูงให้มีขนาดสูงขึ้น นั่นคือ การพรีเอมฟาซิส ก็คือ การกรองสัญญาณด้วยวงจรกรองความถี่สูงผ่าน (high pass filter) ซึ่งมักนิยมใช้วงจรกรองอันดับหนึ่ง มีฟังก์ชันถ่ายโอนเป็น

$$H(z) = 1 - a \cdot z^{-1} \quad (3.1)$$



รูปที่ 3.3 แสดง วงจรกรองความถี่สูงผ่าน

โดยที่ $0.9 < a < 1$ เมื่อ เทียบกับรูป 3.3 เราจะได้ว่า

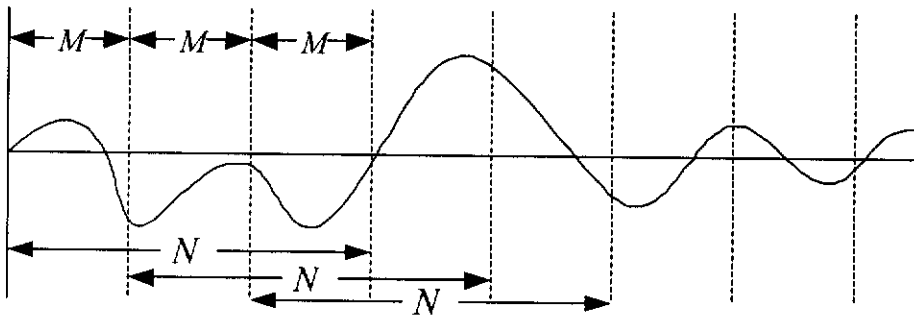
$$s'(n) = s(n) - a \cdot s(n-1) \quad (3.2)$$

เมื่อ สมมุติว่า สัญญาณเดิมเป็น $s(n)$ และ เมื่อ ประมวลผลสัญญาณแล้ว จะเป็น $s'(n)$

ยิ่งค่า a มีค่าเข้าใกล้ 1 เท่าใด สัญญาณที่ความถี่สูงก็จะถูกขยายมากขึ้นเท่านั้น ค่า a ที่นิยมใช้สำหรับการหาค่าสัมประสิทธิ์ของ LPC คือ $\frac{15}{16} = 0.9375$

3.1.2 การแบ่งช่วงสัญญาณ (Frame blocking)

สัญญาณที่ผ่านการพรีเอมฟาซิสแล้ว $s'(n)$ จะถูกตัดแบ่งออกเป็นช่วงๆ หรือ เป็นเฟรมๆ เฟรมละ N ตัวอย่างสัญญาณ ในการวิเคราะห์หาค่าสัมประสิทธิ์ เราจะวิเคราะห์ทีละช่วงของแต่ละ N ตัวอย่างสัญญาณ ดังรูปที่ 3.4



รูปที่ 3.4 แสดง การแบ่งช่วงของสัญญาณ

โดยช่วงในการวิเคราะห์แต่ละช่วงจะถูกเลื่อนไปเป็นระยะ M ช่วงสัญญาณ จะเห็นได้ว่า ถ้าค่า M ใหญ่กว่าค่า N ในการเลื่อนของช่วงในการวิเคราะห์จะทำให้บางสัญญาณไม่ถูกใช้ในการวิเคราะห์ ก็จะเป็นการสูญเสียส่วนหนึ่งทำให้ผลที่ได้ไม่ถูกต้องเท่าที่ควร ถ้าค่า M เล็กกว่า N ยิ่งค่า M เล็กเท่าไร ก็จะทำให้ตัวอย่างสัญญาณทุกตัวถูกนำมาวิเคราะห์ จะมีความแม่นยำในการวิเคราะห์สูงขึ้นเท่านั้น แต่ก็จะทำให้การคำนวณช้าลง

หลักการกำหนดขนาดของวินโดว์

3.1.2.1 วินโดว์จะต้องสั้นพอที่คุณสมบัติของเสียงที่เราสนใจจะวิเคราะห์นั้น ยังไม่เปลี่ยนแปลงในวินโดว์

3.1.2.2 วินโดว์จะต้องยาวพอที่ จำนวนตัวอย่างสัญญาณในวินโดว์ สามารถนำมาคำนวณหาคุณสมบัติที่ต้องการได้

3.1.2.3 วินโดว์ที่ติดกันไม่ควรจะสั้นจนกระโดดข้ามข้อมูลบางส่วนไป แต่ควรเลื่อนวินโดว์ให้ น้อยกว่าขนาดของเฟรม

เนื่องจาก เราใช้ความถี่ในการสุ่มสัญญาณ 8 กิโลเฮิรตซ์ ในการวิเคราะห์นี้ เราเลือกใช้ค่า $N = 80$ และ ค่า $M = 40$ นั่นคือ ช่วงในการวิเคราะห์ คือ 10 มิลลิวินาที และ ระยะในการเลื่อนเฟรมจะมี ค่าประมาณ 5 มิลลิวินาที

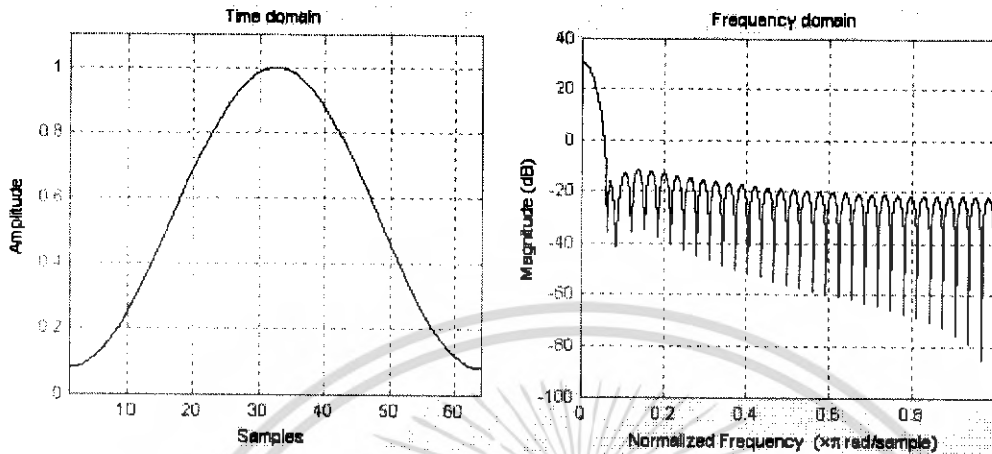
3.1.3 การวินโดว์

พิจารณาช่วงสัญญาณ N ตัวอย่างสัญญาณของช่วงใดๆ ที่ตัดมาวิเคราะห์ จะเห็นว่า ที่ขอบเขตของเฟรมที่ตัดมานี้ มีความไม่ต่อเนื่องของสัญญาณ ถ้าเราพิจารณาในโดเมนของความถี่สูง ก็จะพบว่ามีความถี่สูงเกิดขึ้น ดังนั้น เพื่อที่จะลดองค์ประกอบของความถี่สูงเหล่านี้ เราจะคูณด้วยฟังก์ชันวินโดว์ เพื่อลดความไม่ต่อเนื่องของสัญญาณที่บริเวณขอบ และ ไม่ทำให้สเปกตรัมของสัญญาณในช่วงความถี่ต่ำเปลี่ยนแปลงไปมากนัก โดย ในที่นี้ เราจะใช้ฟังก์ชันวินโดว์แฮมมิง (Hamming window function) ซึ่ง นิยามโดยสมการ ดังนี้ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (3.3)$$

เมื่อ $n = 0, 1, 2, \dots, N-1$



รูปที่ 3.5 แสดงองค์ประกอบทางเวลา และ ทางความถี่ของฟังก์ชันวินโดว์ แบบแฮมมิง

จากรูป 3.5 เราจะได้ว่า

$$x'(n) = x(n) \cdot w(n) \quad (3.4)$$

3.1.4 การวิเคราะห์หัตถ์ออโตคอร์รีเลชัน (autocorrelation)

สมมติว่า สัญญาณเดิมเป็น $s(n)$ การประมาณค่า สัญญาณเป็น $s'(n)$ ดังนั้น สามารถอธิบายการประมาณเชิงเส้นด้วยสมการ ต่อไปนี้

$$s'(n) = \sum_{k=1}^p a_k s(n-k) \quad (3.5)$$

เมื่อ a_k เป็น ค่าคงที่ เรียกวิธีการนี้ว่า การประเมินเชิงเส้น อันดับที่ p โดยมีเงื่อนไขว่า ค่า a_k ที่ใช้ในการประมาณจะต้องทำให้ ผลรวมของกำลังสองของความคลาดเคลื่อน คือ $(s(n) - s'(n))^2$ มีค่าน้อยที่สุด นั่นคือ $\sum e^2(n) = \sum (s(n) - s'(n))^2$ มีค่าต่ำที่สุด ซึ่งจะใช้การประมาณเชิงเส้น แบบวิธีออโตคอร์รีเลชัน (Autocorrelation Method) หรือ วิธีอิตตสัมพันธ์

โดยเราจะทำการหาค่าจาก

$$a = R_n^{-1} \Gamma_n \quad (3.6)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $R_n = \begin{bmatrix} R_n(0) & R_n(1) & R_n(2) & \cdots & R_n(p-1) \\ R_n(1) & R_n(0) & R_n(1) & \cdots & R_n(p-2) \\ R_n(2) & R_n(1) & R_n(0) & \cdots & R_n(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_n(p-1) & R_n(p-2) & R_n(p-3) & \cdots & R_n(0) \end{bmatrix}$, $a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ \vdots \\ a_p \end{bmatrix}$ และ

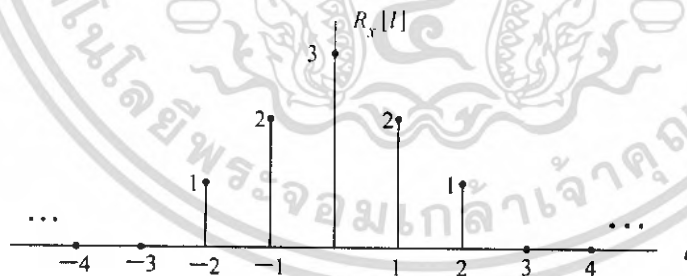
$r_n = \begin{bmatrix} R_n(1) \\ R_n(2) \\ R_n(3) \\ \vdots \\ R_n(p) \end{bmatrix}$

3.1.5 การวิเคราะห์หาค่าสัมประสิทธิ์ LPC

โดยเราสามารถแก้สมการ โดยใช้ Algorithm ของ Levinson-Durbin มาช่วยได้ ดังตัวอย่างต่อไปนี้

ตัวอย่าง

พิจารณาการทำนายเชิงเส้นของ Real Random Process ด้วยฟังก์ชันความสัมพันธ์ที่แสดงดังรูป



รูปที่ 3.6 แสดง ฟังก์ชันความสัมพันธ์

จากรูป เราจะได้ค่าของฟังก์ชันความสัมพันธ์ คือ

$$R_n[0] = 3, R_n[1] = R_n[-1] = 2 \text{ และ } R_n[2] = R_n[-2] = 1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตัวอย่างนี้ เราจะทำการหาค่าสัมประสิทธิ์ LPC (Order 3)

ซึ่งเราจะแก้สมการ โดยใช้สมการ ตามสมการที่ (2.25)-(2.27) โดยค่า $R_n[x]$ คือ ค่าที่ได้จากการ ทำ อัดดสัมพันธ์

เราจะต้องกำหนดค่าเริ่มต้นตามเงื่อนไข

$$a_0 = 1, r_0 = R_n[1] \text{ และ } E_0 = R_n[0] \quad (3.7)$$

ดังนั้น ในตัวอย่างนี้ เราจะกำหนดค่าเริ่มต้น ดังนี้

$$r_0 = R_n[1] = 2, a_0 = 1 \text{ และ } E_0 = R_n[0] = 3$$

ทำการหาค่าพารามิเตอร์ที่ first stage ของ recursion ($p=1$)

$$k_1 = \frac{r_0 a_0}{E_0} = \frac{2 \cdot 1}{3} = \frac{2}{3}$$

$$a_1 = \begin{bmatrix} a_0 \\ 0 \end{bmatrix} - k_1 \begin{bmatrix} 0 \\ \tilde{a}_0^* \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 0 \end{bmatrix} - \frac{2}{3} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{2}{3} \end{bmatrix}$$

$$E_1 = (1 - |k_1|^2) E_0$$

$$= \left(1 - \left|\frac{2}{3}\right|^2\right) \cdot 3 = \frac{5}{3}$$

เมื่อทำการหาค่าพารามิเตอร์ของ first stage แล้ว ก็จะทำทำการหาค่าพารามิเตอร์ของ stage ต่อไป ของกระบวนการ recursion ในรูปแบบเดิม ($p=2$)

$$k_2 = \frac{r_1^* \tilde{a}_1}{E_1}$$

$$= \frac{1}{5/3} \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} -\frac{2}{3} \\ 1 \end{bmatrix} = -\frac{1}{5}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$a_2 = \begin{bmatrix} a_1 \\ 0 \end{bmatrix} - k_2 \begin{bmatrix} 0 \\ \tilde{a}_1^* \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ -\frac{2}{3} \\ 0 \end{bmatrix} - \left(-\frac{1}{5}\right) \begin{bmatrix} 0 \\ -\frac{2}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{4}{5} \\ \frac{1}{5} \end{bmatrix}$$

$$E_2 = (1 - |k_2|^2) E_1$$

$$= \left(1 - \left|-\frac{1}{5}\right|^2\right) \cdot \frac{5}{3} = \frac{8}{5}$$

ดังนั้น ค่าสัมประสิทธิ์ LPC (Order 3) คือ

$$\begin{bmatrix} 1 \\ -\frac{4}{5} \\ \frac{1}{5} \\ 1 \\ 5 \end{bmatrix}$$

3.1.6 การเปลี่ยนพารามิเตอร์ LPC เป็น สัมประสิทธิ์เชปสตรัม

หลังจากที่หาสัมประสิทธิ์ LPC และ Gain ใน 1 เฟรมแล้ว เราจะเปลี่ยนสัมประสิทธิ์ LPC ให้เป็นสัมประสิทธิ์เชปสตรัม เนื่องจาก สัมประสิทธิ์เชปสตรัม จะเป็น พารามิเตอร์ที่มีลักษณะความน่าเชื่อถือได้ดีกว่า สัมประสิทธิ์ LPC ทั้งยังมีความสัมพันธ์ใกล้ชิดกับการรับรู้เสียงตามความรู้สึกของมนุษย์โดยแท้จริง สัมประสิทธิ์เชปสตรัม สามารถหาได้โดยตรงจากสัมประสิทธิ์ LPC ที่แสดงดังสมการ

$$C_0 = \ln G \quad (3.8)$$

$$C_m = a_m + \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \quad \text{เมื่อ } 1 \leq m \leq p \quad (3.9)$$

$$C_m = \sum_{k=1}^{m-1} \left(\frac{k}{m}\right) C_k a_{m-k} \quad \text{เมื่อ } m > p \quad (3.10)$$

$$Q \approx \frac{3}{2} p \quad (3.11)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย $p=10$ ดังนั้น $Q=15$ นำมารวมกับสัมประสิทธิ์ตัวแรก เราจะได้ว่า จำนวนของสัมประสิทธิ์เซปสตรีม ใน 1 เฟรม จะเท่ากับ 16 ตัว

3.1.7 การเวทค่าพารามิเตอร์ (Parameter weighting)

เนื่องจาก สัมประสิทธิ์เซปสตรีมที่ได้ นั้น ช่วงลำดับต้นๆ และ ลำดับท้ายๆ ของเฟรมที่นำมาวิเคราะห์ จะเกิดความคลาดเคลื่อนมากกว่าบริเวณส่วนอื่น เพราะฉะนั้น จึงทำการถ่วงน้ำหนัก เพื่อลดค่าความคลาดเคลื่อนดังกล่าวนี้ ด้วยฟังก์ชันเวทดัง ดังสมการ

$$W_m = \left[1 + \frac{Q}{2} \sin\left(\frac{\pi n}{Q}\right) \right] \quad \text{เมื่อ } 1 \leq m \leq Q \quad (3.12)$$

และ จะได้ว่า $C'_m = C_m * W_m$ (3.13)

จากนั้น ทำการพิจารณาให้ครบทุกเฟรมของข้อมูล เมื่อพิจารณาเรียบร้อยแล้ว ก็จะนำไปจัดกลุ่มเสียง และสร้างแบบจำลองเสียง เพื่อใช้ในการเปรียบเทียบต่อไป

โดยในที่นี้ เราจะนำบอร์ด TMS320c6713 มาประยุกต์ใช้ กับการวิเคราะห์สัญญาณเสียง โดยใช้ LPC Model (Linear Predictive Coding Model) โดยจะกล่าวรายละเอียดในหัวข้อต่อไป

3.2 การสร้างโค้ดบุค

ในโครงการนี้ทำการออกแบบสร้างโค้ดบุคโดยการนำเทรนนิ่งเซตที่ได้จากการประมาณเชิงเส้น มาผ่านกระบวนการที่มีขั้นตอนดังนี้คือ

1. ทำการสุ่มค่าเริ่มต้นของจุดศูนย์กลางของโค้ดบุคขึ้นมา 64 ตัว ตัวละ 16 มิติ
2. หาระยะทางระหว่างโค้ดบุคเริ่มต้นกับเทรนนิ่งเซตแต่ละเฟรม ในที่นี้ใช้วิธีวัดความคลาดเคลื่อนโดยใช้วิธีการคำนวณแบบ square error distortion ในการหาระยะทางเนื่องจากเป็นวิธีที่ค่อนข้างง่ายและรวดเร็ว
3. ทำการจัดกลุ่มของเทรนนิ่งเซตโดยพิจารณาจากระยะทางระหว่างเทรนนิ่งเซตแต่ละเฟรมกับโค้ดบุคเริ่มต้น ว่าน้อยที่สุดที่กลุ่มใดก็จัดให้อยู่ในกลุ่มนั้น
4. คำนวณหาจุดศูนย์กลางของกลุ่มใหม่ โดยคำนวณจากค่าเฉลี่ยของเวกเตอร์ทั้งหมดในกลุ่มนั้น ซึ่งจะได้เป็นค่าของจุดศูนย์กลางใหม่
5. นำค่าของจุดศูนย์กลางใหม่ที่ได้มาทำการหาค่าความคลาดเคลื่อนรวมเทียบกับจุดศูนย์กลางเริ่มต้นที่สุ่มขึ้น การคำนวณค่าความคลาดเคลื่อนรวมใช้การคำนวณแบบ square error distortion
6. หากค่าความคลาดเคลื่อนรวมที่ได้มีค่ามากกว่า 0.01 ก็จะกลับไปทำข้อ 2-5 ใหม่แต่จุดศูนย์กลางที่จะนำไปหาความคลาดเคลื่อนเทียบกับเทรนนิ่งเซตนั้นจะใช้จุดศูนย์กลางใหม่ที่คำนวณมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

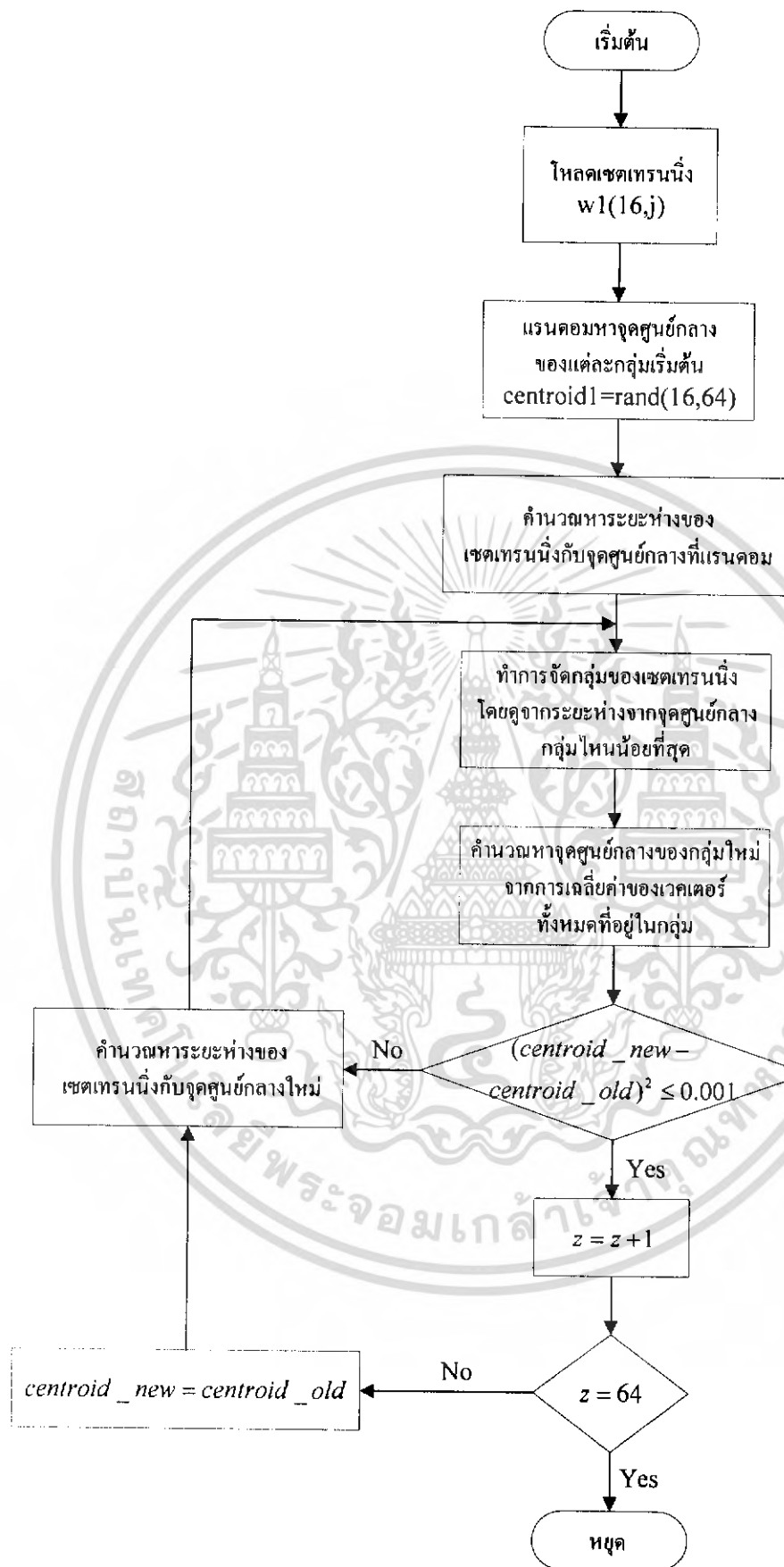
โดยจะทำเช่นนี้จนกว่าค่าความคลาดเคลื่อนรวมจะมีค่าน้อยกว่าหรือเท่ากับ 0.01 ค่าของจุดศูนย์กลางที่ได้จะเป็นค่าของโศคบุคที่ต้องการนั่นเอง

3.3 การหาค่าดัชนีโศคบุค

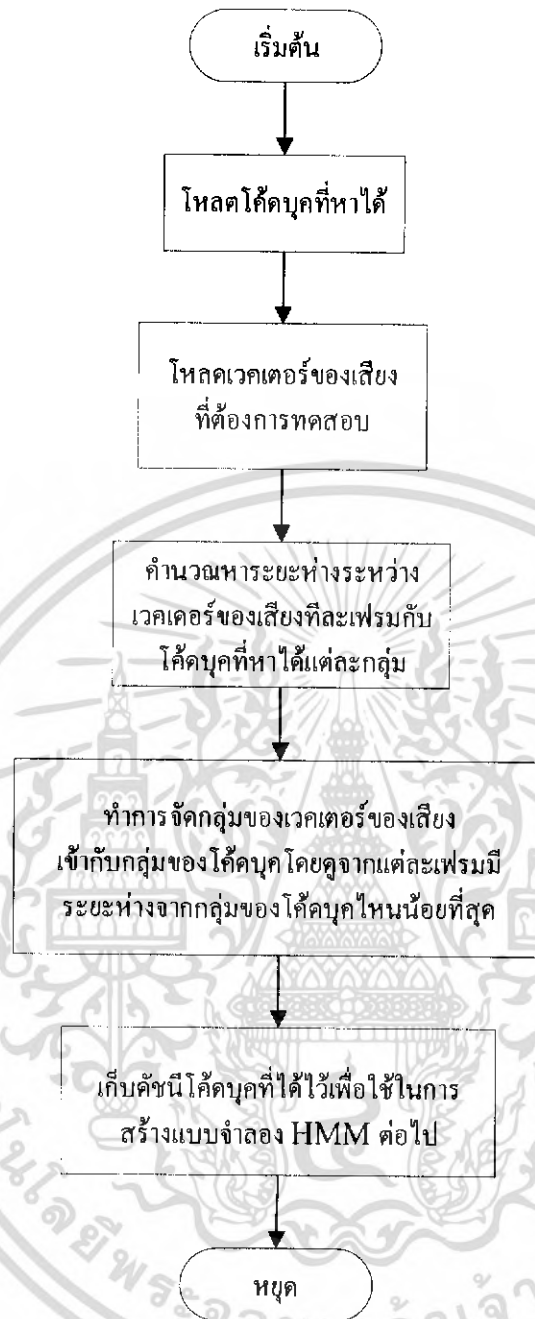
โดยการนำเวกเตอร์ของเสียงที่ได้จากการประมาณเชิงเส้นมาที่ละเสียง นำมาเปรียบเทียบกับโศคบุคที่ได้จากการสร้างในขั้นตอนของการเรียนรู้ที่ละเฟรม โดยใช้วิธีการคำนวณแบบ square error distortion โดยพิจารณาว่าเวกเตอร์ของเสียงแต่ละเฟรมห่างจากโศคบุคใดน้อยที่สุด จะได้เป็นค่าดัชนีโศคบุคของเฟรมเสียงนั้น และจะทำการเก็บดัชนีโศคบุคของเสียงแต่ละเฟรมไว้เป็นลำดับค่าปรากฏ (observation sequence) สำหรับนำไปสร้างเป็นแบบจำลองของเสียงโดยใช้ HMM ต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 Flow Chart แสดงการทำงานของโปรแกรมสร้าง Codebook
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 Flow Chart แสดงการทำงานของโปรแกรมหาค่าดัชนีโศคบุค

3.4 การออกแบบ Hidden Markov Model (HMM)

3.4.1 การสร้างแบบจำลอง HMM ของเสียง

3.4.1.1.) คู่ค่าเริ่มต้น a, b และ กำหนดให้ $\pi = [1 0 0 0 0]$ ตามเงื่อนไขในการใช้แบบจำลองแบบ Left-Right

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.1.2.) หาค่า α, β จากค่า a, b เริ่มต้น และ ลำดับค่าปรากฏ $O = \{o_1, o_2, \dots, o_T\}$ ซึ่งเรียกว่า ลำดับเทรนนิ่ง ตามวิธีของ Forward และ Backward Procedure โดยใช้ลำดับของค่าปรากฏหลายๆ ลำดับ เข้ามาเทรน เพื่อความถูกต้องมากขึ้น

3.4.1.3.) ทำการสเกลลิง α เพื่อให้ค่าอยู่ในช่วงที่คอมพิวเตอร์สามารถคำนวณได้อย่างถูกต้อง

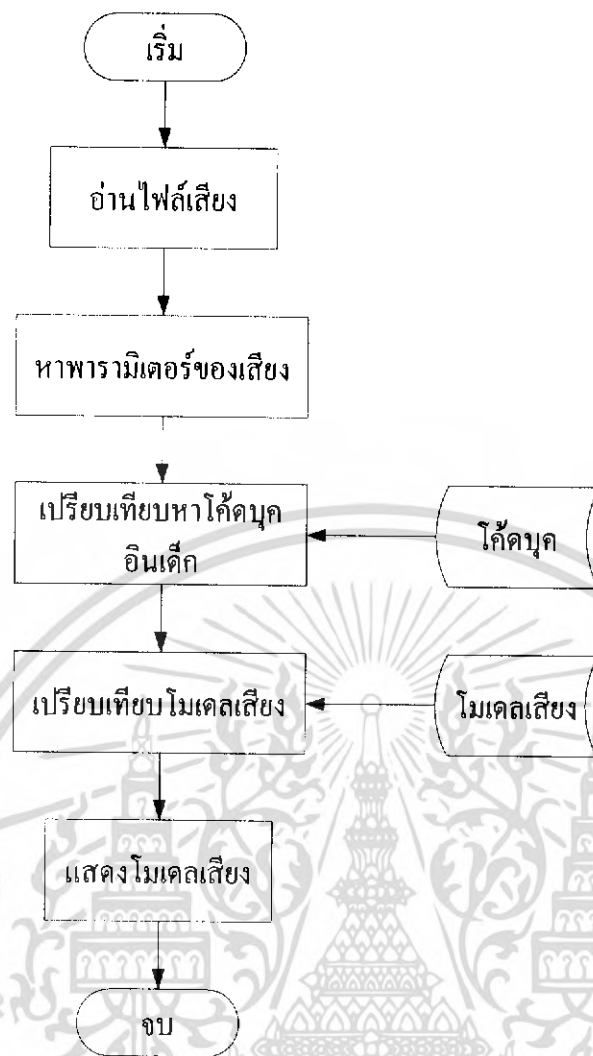
3.4.1.4.) หาค่าพารามิเตอร์ a, b, π ที่ให้ค่าความน่าจะเป็นสูงสุด ที่จะเป็นแบบจำลอง λ ที่เหมาะสมของเสียง

3.4.1.5.) ตรวจสอบค่าพารามิเตอร์ของแบบจำลองที่ได้ ว่าสู้เข้าหรือยัง โดยใช้วิธีการคำนวณค่า a, b ซ้ำประมาณ 50 รอบ เมื่อมีการเปลี่ยนแปลงน้อยมาก จนเป็นที่น่าพอใจ ตามระดับค่าที่ตั้งไว้ในที่นี้ ใช้ค่าเท่ากับ 10^{-5} ก็จะหยุด และได้ค่าพารามิเตอร์ a, b และ π ของแบบจำลองที่ต้องการ

3.4.1.6.) เก็บค่าพารามิเตอร์ a, b, π ที่ได้จากขั้นตอนที่ 5 เป็นพารามิเตอร์ของแบบจำลองไว้

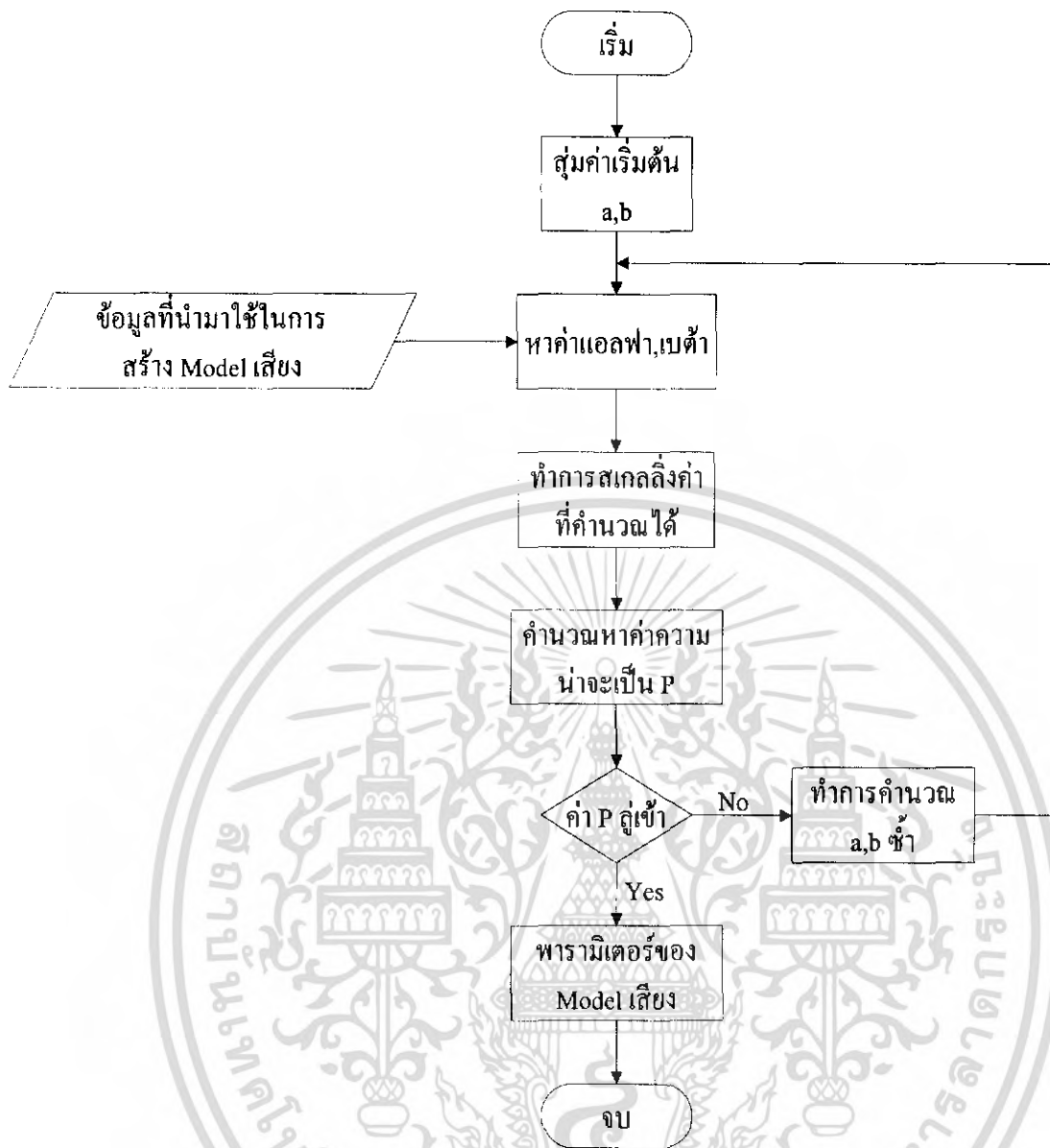
3.5 ขั้นตอนการตัดสินใจ

หลังจากที่ได้แบบจำลอง HMM ของแต่ละเสียงแล้ว เมื่อมีลำดับของค่าปรากฏ $O = \{o_1, o_2, \dots, o_T\}$ ของเสียงที่ unknown ซึ่งเป็นเสียงที่ต้องการทดสอบเข้ามา เราจะทำการคำนวณหาความน่าจะเป็น $P(O|\lambda)$ ทุกแบบจำลองของแต่ละเสียง โดยใช้วิธี Viterbi Algorithm แล้วเลือกเอาค่าที่มีความน่าจะเป็นสูงสุด ซึ่งก็คือ เสียงที่แบบจำลองสามารถจดจำได้นั่นเอง



รูปที่ 3.9 แสดงขั้นตอนการวิเคราะห์และจดจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

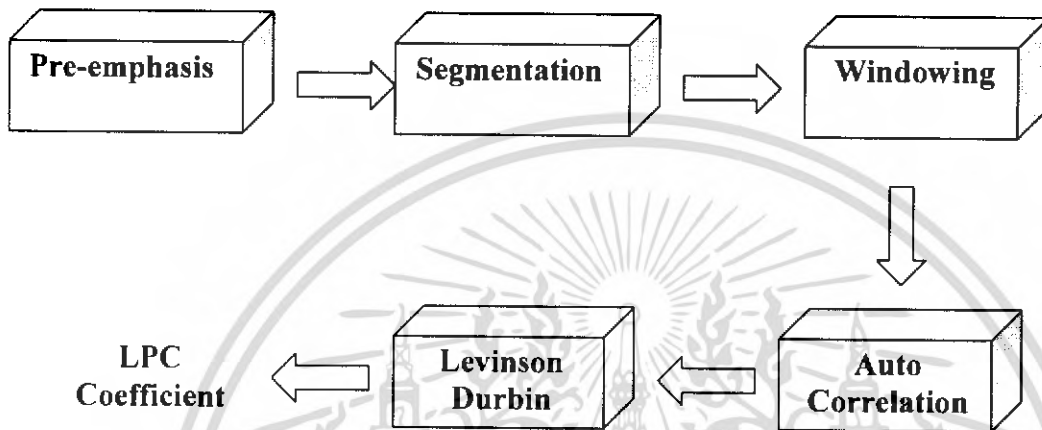


รูปที่ 3.10 แสดง Flow chart การทำงานของการเรียนรู้เสี่ยง และการสร้างโมเดลเสี่ยง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 การออกแบบระบบ เพื่อหาคุณลักษณะเด่นของเสียง โดย บอร์ด TMS320C6713

ในการหาคุณลักษณะเด่นของเสียง (Feature Attraction) จะใช้วิธีการหา สัมประสิทธิ์ LPC ดังที่กล่าวไปแล้วในส่วนของการทฤษฎี ซึ่งขบวนการในการ implement ลงในบอร์ด TMS 320C6713 มีดังนี้



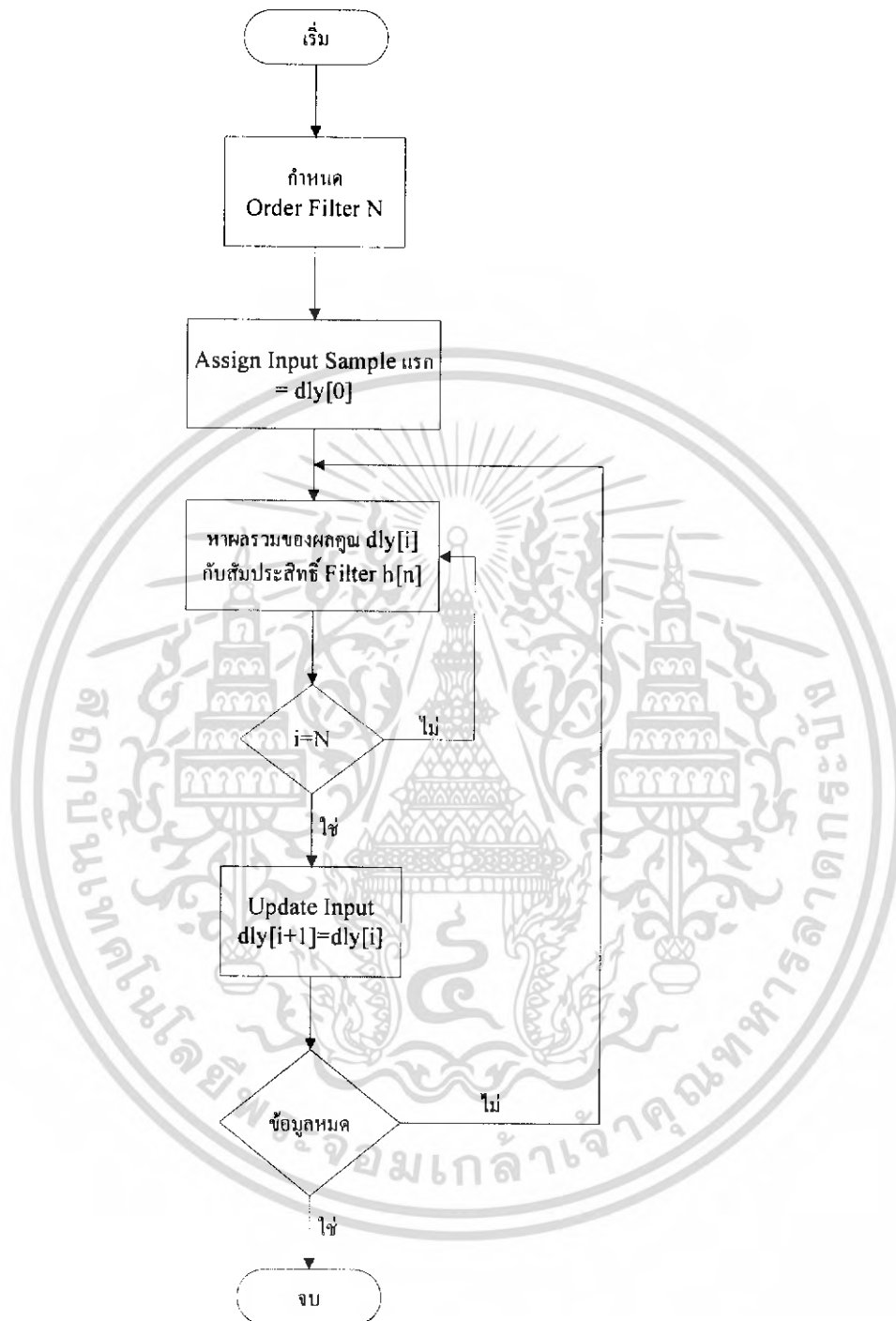
รูปที่ 3.11 แสดง block diagram การหา feature attraction

3.6.1 Pre-emphasis

ทดลองทำการสร้างวงจร Filter ซึ่งจะทำการออกแบบ FIR filter โดยใช้ฟังก์ชัน Sptool ในโปรแกรม MATLAB เพื่อหาสัมประสิทธิ์ $h[n]$ ของ filter หลังจากนั้นจึงนำมาทำการ convolution กับสัญญาณ input $x[n]$ ดังสมการ

$$y(n) = \sum_{k=0}^{\alpha} x(k)h(n-k) \quad (3.14)$$

ซึ่งจะมี Flow chart ในการเขียน โปรแกรมดังนี้



รูปที่ 3.12 แสดง Flow Chart ของ โปรแกรม Pre-emphasis

3.6.2 Segmentation

โปรแกรม Segmentation ใช้ในการตัดเฟรมข้อมูลมาพิจารณา ข้อมูลที่เข้ามาของเสียงหนึ่งเสียงมีประมาณ 4800 samples ซึ่งเป็นข้อมูลที่ยาวเกินไปในการที่จะนำมาหา คุณลักษณะเด่นของเสียง จึงต้องมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแบ่งข้อมูลออกเป็น เฟรมย่อยๆ เฟรมละ 80 samples และจะมีการ overlap กันอยู่ 40 samples เพื่อป้องกันการสูญหายของข้อมูล โดยจะต้องคำนวณหาจำนวนเฟรมข้อมูลทั้งหมด ดังนี้

$$n_{\text{frame}} = \frac{(l - \text{frame} + \text{shift})}{\text{shift}} \quad (3.9)$$

l = ความยาวข้อมูล

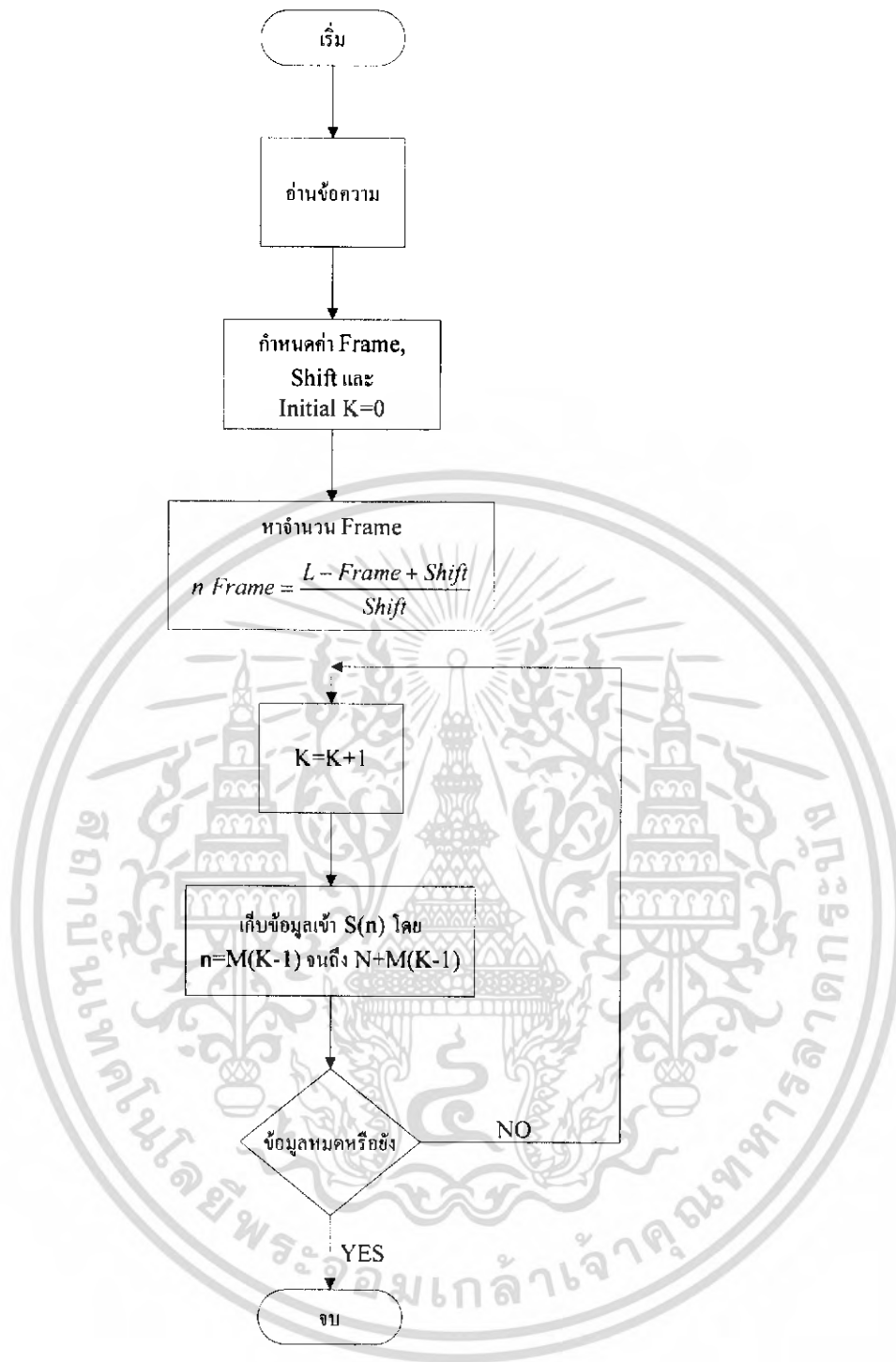
frame = จำนวน sample ในหนึ่งเฟรม

shift = จำนวน samples ที่ overlap กัน

n_{frame} = จำนวน frame ทั้งหมด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



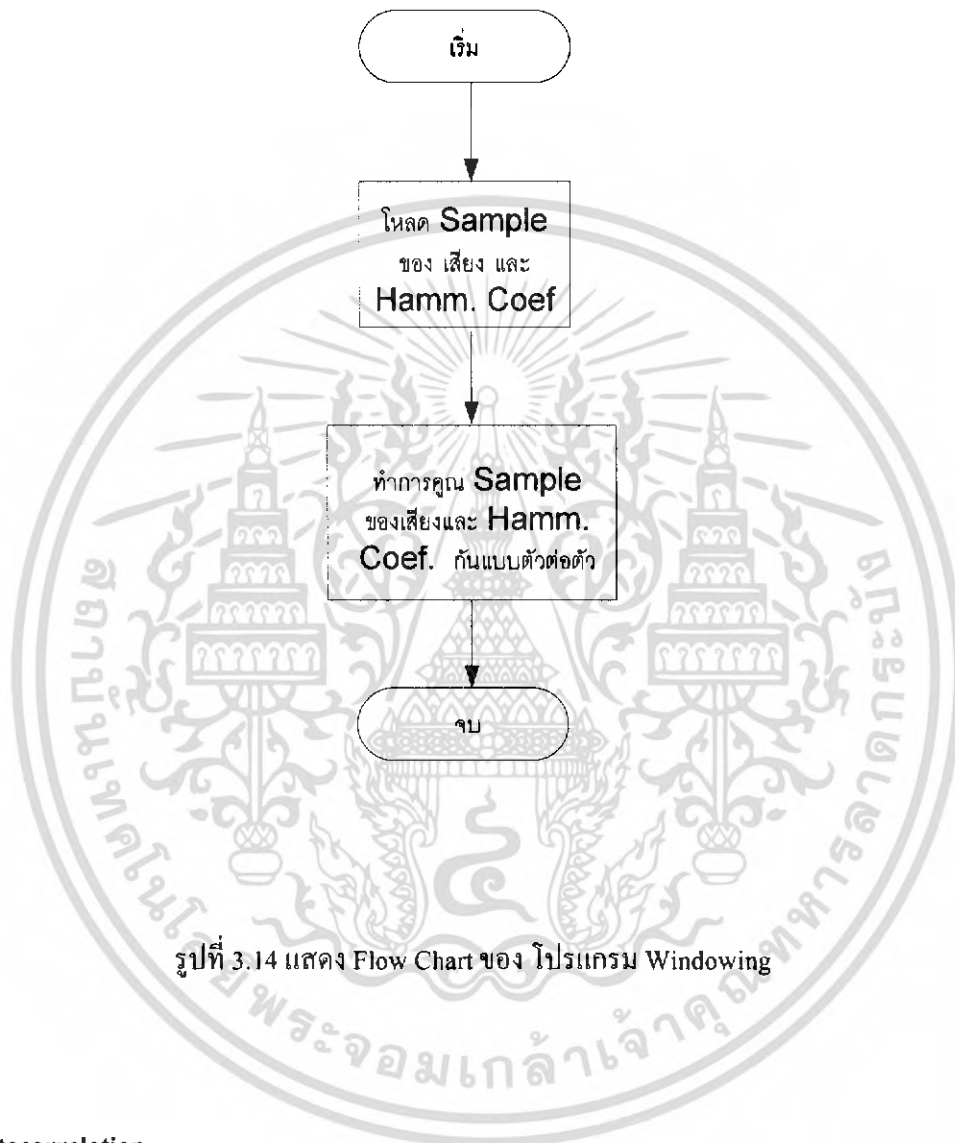
รูปที่ 3.13 แสดง Flow Chart ของ โปรแกรม Segmentation

3.6.3 Windowing

หลังจากทำการตัดข้อมูลเสียงออกเป็นเฟรมๆ แล้วจะมีจำนวนเฟรมละ 80 samples ก็ให้นำ samples ในเฟรมมาคูณกับสัมประสิทธิ์ Hamming ที่กำหนด order ไว้ 80 แบบตัวต่อตัว

สมการในการหาสัมประสิทธิ์ Hamming มีดังนี้
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

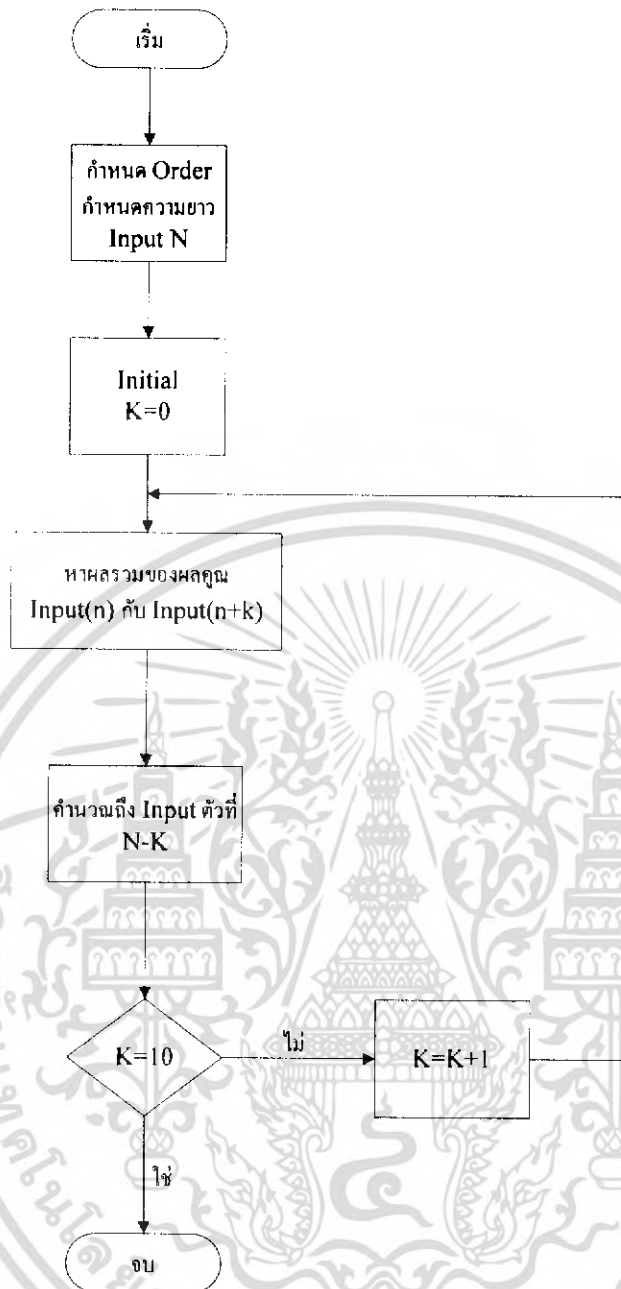
$$w(n) = 0.54 - 0.46 \cos\left(\frac{2\pi n}{N-1}\right) \quad (3.9)$$



3.6.4 Autocorrelation

ในการหาค่า auto correlation ในที่นี้จะใช้ค่า order เท่ากับ 10 ซึ่งเมื่อทำการหาค่า auto correlation แล้ว จะได้ผลลัพธ์ออกมา 10 ค่า ในแต่ละเฟรมของข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

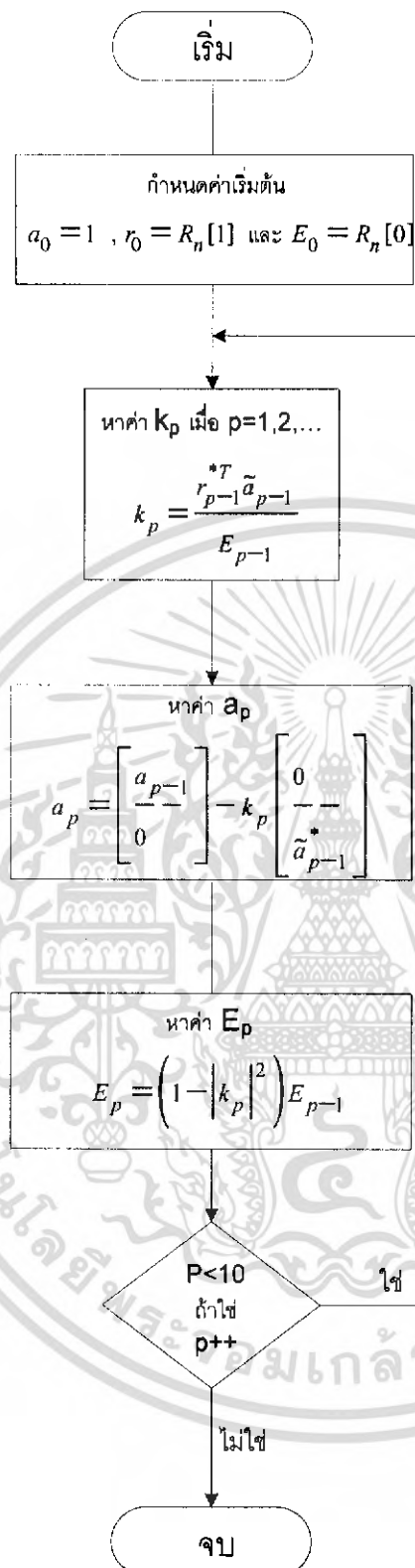


รูปที่ 3.15 Flow Chart ในการเขียน โปรแกรม Autocorrelation

3.6.5 Levinson Durbin

นำค่าที่ได้จาก การทำ auto correlation มาหาสัมประสิทธิ์ LPC ซึ่งมีอัลกอริทึมดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 Flow Chart ในการเขียน โปรแกรม Levinson Durbin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 ผลการทดลอง

4.1 ผลการทดลองในส่วนของการ Simulate ระบบการหาค่าคุณลักษณะเด่นของเสียง

จากการทดลองที่เราทำการ หาค่าสัมประสิทธิ์ LPC จากบอร์ด TMS เปรียบเทียบผลที่ได้จากโปรแกรม MATLAB โดยค่าคุณลักษณะต่างๆ ของสัญญาณเสียงที่นำมาทำการพล็อตนั้น เราจะนำมาจากเฟรมแรกของทุกๆ เสียง ทั้งนี้ เนื่องจากการหาค่าต่างๆ ผลที่ได้ของเสียงหนึ่งเสียงจะมีค่าต่างๆ มาก เราไม่สามารถที่จะนำมาทำการพล็อตได้หมด เราจึงนำเอาข้อมูลของเฟรมแรกของทุกๆ เสียง มาพิจารณา

4.1.1 ผลจากการทดลองหาค่า Autocorrelation

ก่อนที่จะทำการหาค่าสัมประสิทธิ์ LPC จะต้องทำการหาค่า Autocorrelation จากเฟรมเสียงที่ผ่านการ Pre – emphasis มา ซึ่งมีผลการทดลองเปรียบเทียบกับโปรแกรม MTLAB ดังนี้

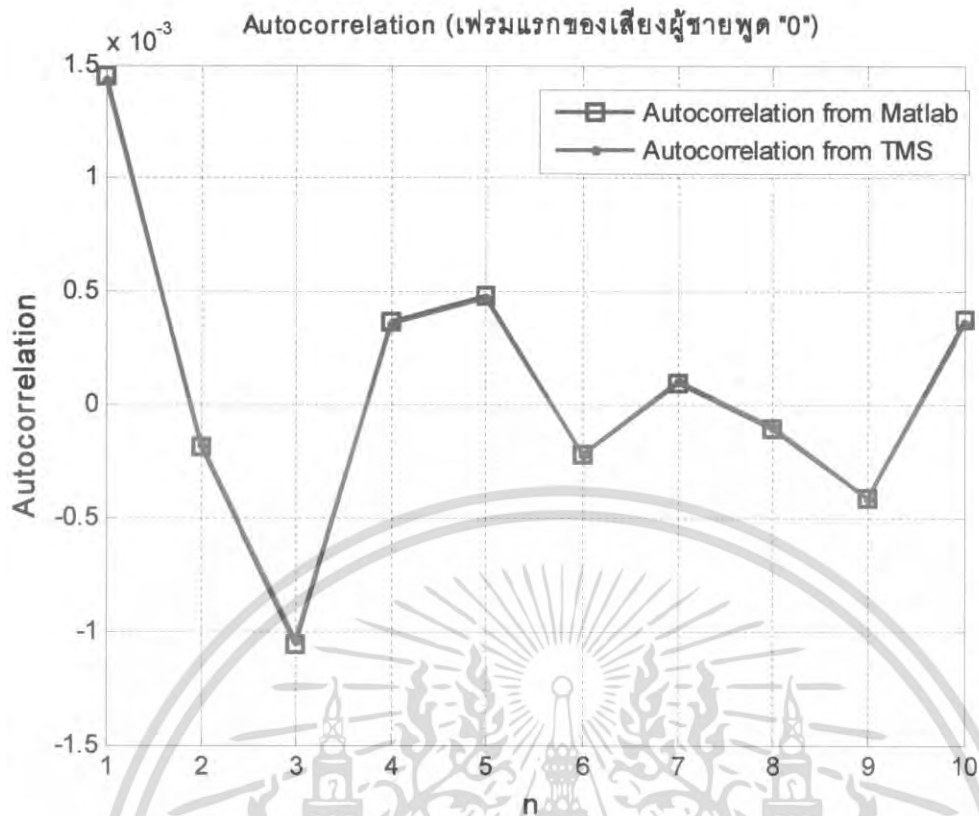
	ค่า Autocorrelation จาก MATLAB	ค่า Autocorrelation จากบอร์ด TMS
1	0.001453385128733	0.001435
2	-0.000180431382916	-0.000177
3	-0.001057569633614	-0.001041
4	0.000369107092244	0.000363
5	0.000482130271423	0.000477
6	-0.000223553848623	-0.000219
7	0.000098038614545	0.000102
8	-0.00010717905674	-0.000101
9	-0.000411698500469	-0.000410
10	0.00037437389113	0.000363

ตารางที่ 4.1 แสดงการเปรียบเทียบค่า Autocorrelation ของเสียงผู้ชายพูด "0" (เฟรมแรก)

จาก โปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS

เมื่อนำค่าในตารางที่ 4.1 มาพล็อตกราฟ เราจะได้กราฟ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 แสดง ค่า Autocorrelation เฟรมแรกของเสียงผู้ชายพูด "0"
จากการเขียนด้วย MATLAB เปรียบเทียบกับผลที่ได้ จากบอร์ด TMS

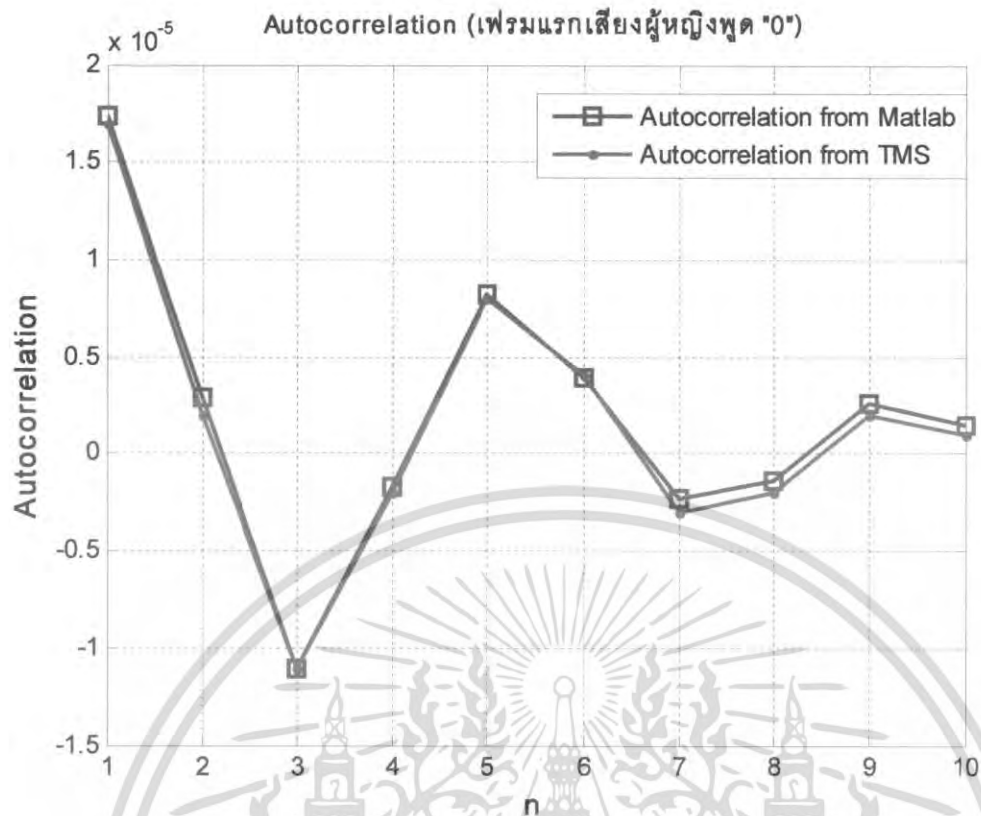
	ค่า Autocorrelation จาก MATLAB	ค่า Autocorrelation จากบอร์ด TMS
1	0.000017369807069	0.000017
2	-0.00000294497143	0.000002
3	-0.000011078188554	-0.000011
4	-0.000001743464201	-0.000002
5	0.00000823031336	0.000008
6	0.00000394119942	0.000004
7	-0.000002350213825	-0.000003
8	-0.00000137891092	-0.000002
9	0.000002598917135	0.000002
10	0.000001506019142	0.000001

ตารางที่ 4.2 แสดงการเปรียบเทียบค่า Autocorrelation ของเสียงผู้ชายพูด "0" (เฟรมแรก)

จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS

เมื่อนำค่าในตารางที่ 4.2 มาพล็อตกราฟ เราจะได้กราฟ ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 แสดง ค่า Autocorrelation เฟรมแรกของเสียงผู้หญิงพูด "0" จากการเขียนด้วย MATLAB เปรียบเทียบกับผลที่ได้ จากบอร์ด TMS

4.1.2 ผลจากการทดลองหาค่าสัมประสิทธิ์ LPC

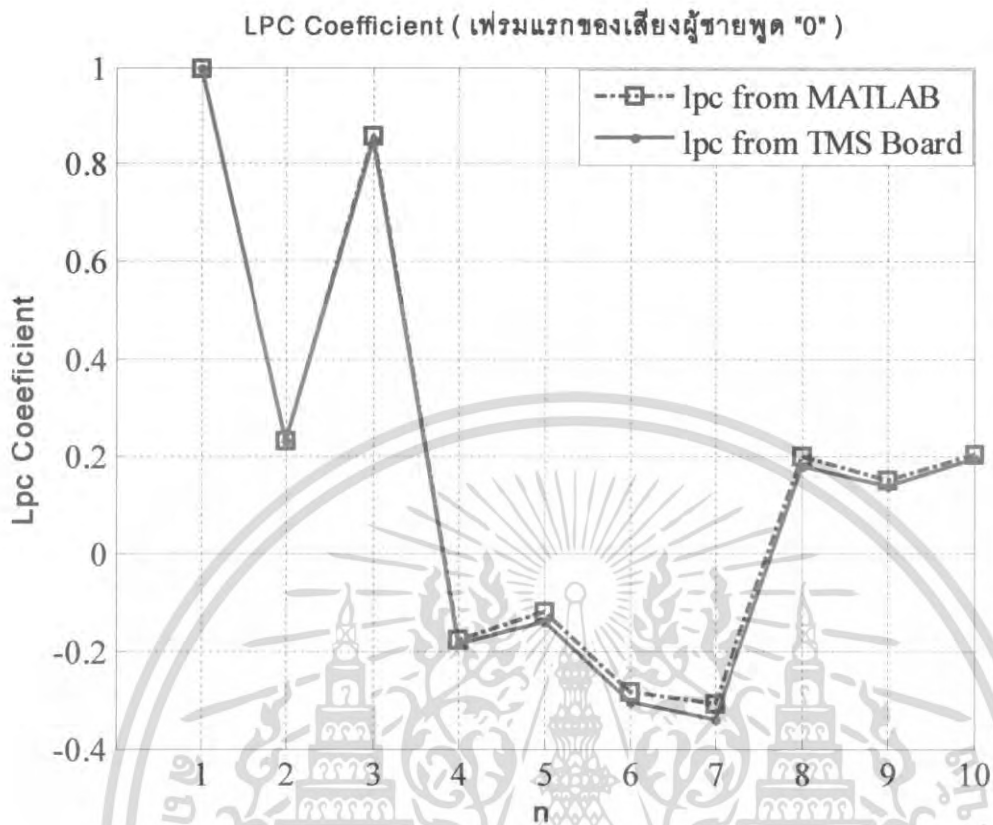
เมื่อทำการหาค่า Autocorrelation ได้แล้ว เราจะนำค่าไปทำการหาค่าสัมประสิทธิ์ LPC ต่อไป โดยใช้ ทฤษฎีของ Levinson-Durbin

	ค่าสัมประสิทธิ์ LPC จาก MATLAB	ค่าสัมประสิทธิ์ LPC จากบอร์ด TMS
1	1	1
2	0.234320073040455	0.232112
3	0.858779422427118	0.849755
4	-0.175775193298666	-0.184900
5	-0.118044540157318	-0.141658
6	-0.282556057720413	-0.304916
7	-0.308491395603839	-0.340107
8	0.201145454067095	0.177413
9	0.152215777658544	0.137136
10	0.203158101440376	0.197434

ตารางที่ 4.3 แสดงการเปรียบเทียบค่าสัมประสิทธิ์ LPC และ Gain ของเสียงผู้ชายพูด "0" (เฟรมแรก)

จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำค่าในตารางที่ 4.3 มาพล็อตกราฟ เราจะได้กราฟ ดังรูป



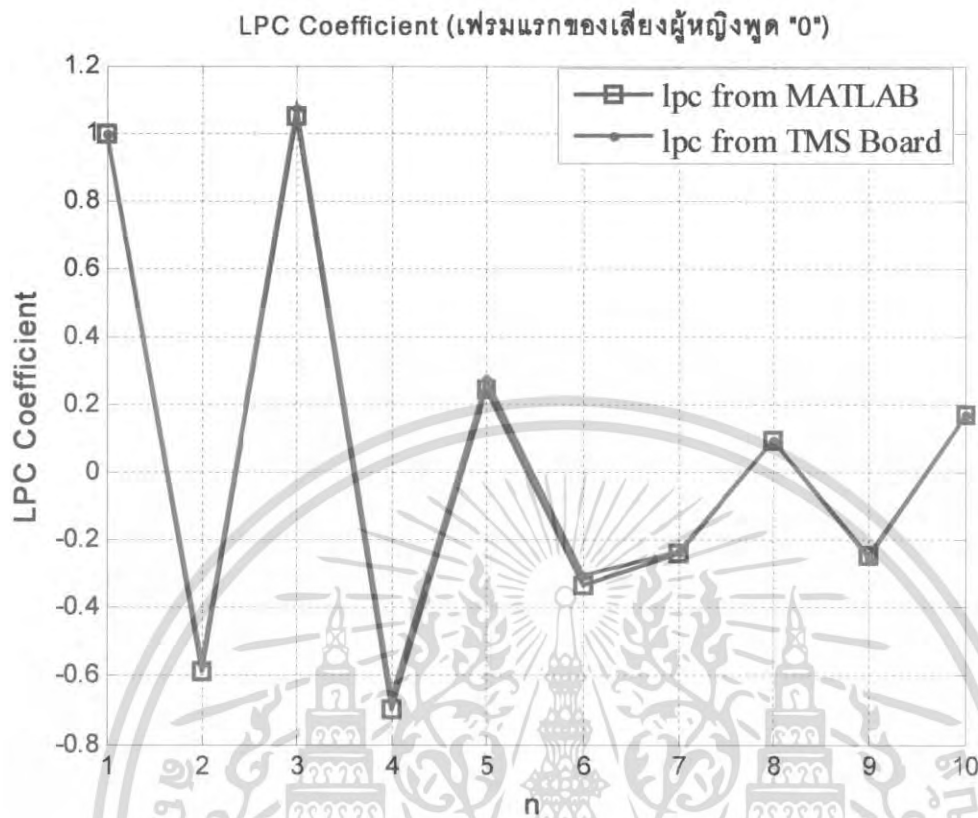
รูปที่ 4.3 แสดง ค่าสัมประสิทธิ์ LPC เฟรมแรกของเสียงผู้ชายพูด "0" จากการเขียนด้วย MATLAB เปรียบเทียบกับผลที่ได้ จากบอร์ด TMS

	ค่าสัมประสิทธิ์ LPC จาก MATLAB	ค่าสัมประสิทธิ์ LPC จากบอร์ด TMS
1	1	1
2	-0.587601220429634	-0.563396
3	1.052544286870074	1.076382
4	-0.697677907486309	-0.660619
5	0.245804643394842	0.275749
6	-0.333119642296959	-0.306705
7	-0.241960498630498	-0.236579
8	0.094899157429876	0.096765
9	-0.244718649294014	-0.250114
10	0.170171431448831	0.168168

ตารางที่ 4.4 แสดงการเปรียบเทียบค่าสัมประสิทธิ์ LPC และ Gain ของเสียงผู้ชายพูด "0" (เฟรมแรก) จากโปรแกรม MATLAB เทียบกับ การหา จากบอร์ด TMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

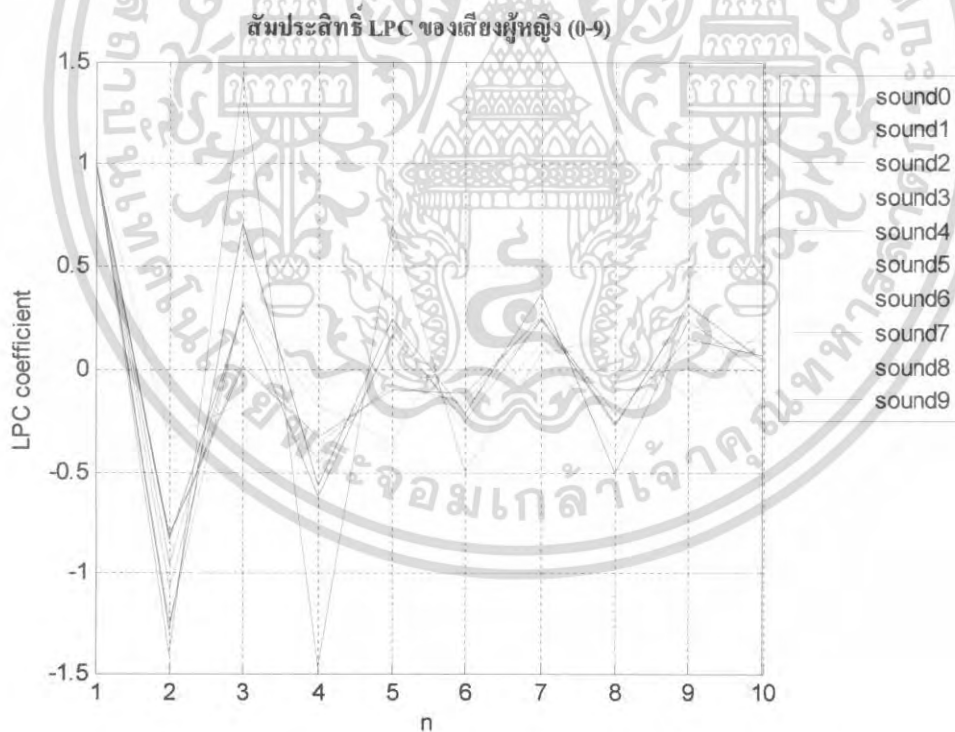
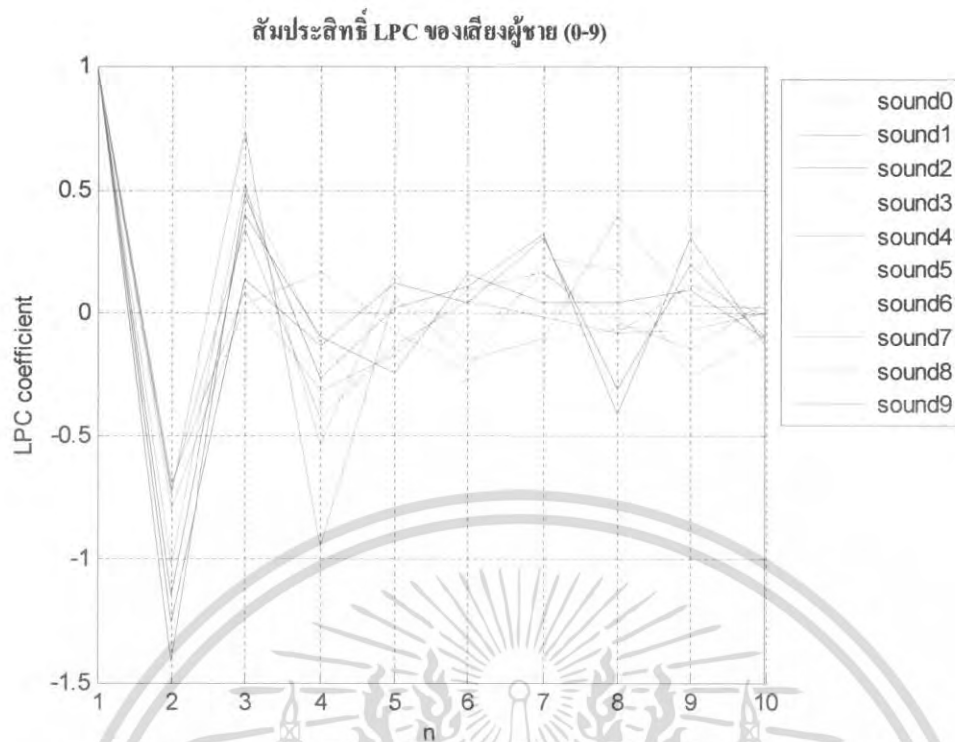
เมื่อนำค่าในตารางที่ 4.4 มาพล็อตกราฟ เราจะได้กราฟ ดังรูป



รูปที่ 4.4 แสดง ค่าสัมประสิทธิ์ LPC เฟรมแรกของเสียงผู้หญิงพูด "0"
จากการเขียนด้วย MATLAB เปรียบเทียบกับผลที่ได้ จากบอร์ด TMS

จากผลการทดลองตามรูปที่ 4.1 – 4.4 ที่เป็นการเปรียบเทียบ ค่าสัมประสิทธิ์ ที่ได้จากโปรแกรม MATLAB และ จากบอร์ด TMS ที่เขียนด้วยภาษา C ซึ่งพบว่าผลที่ได้มีค่าใกล้เคียงกัน และเสียงผู้หญิงและผู้ชาย เมื่อพูดคำเดียวกัน จะมีแนวโน้มของเส้นกราฟเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

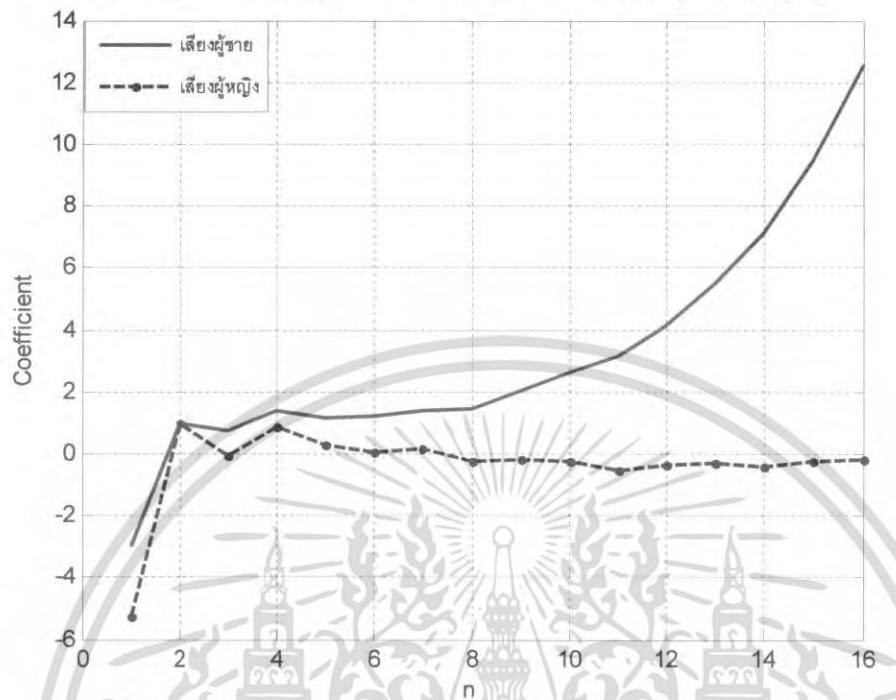


จากผลการทดลองตามรูปที่ 4.5 – 4.6 จะสังเกตเห็นว่า เสียงแต่ละเสียงจะมีค่าสัมประสิทธิ์ที่แตกต่างกัน ซึ่งจะใช้ค่าสัมประสิทธิ์นี้ในการแยกแยะเสียงต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.3 ผลการทดลองหาค่า สัมประสิทธิ์ Cepstrum และ Weight พารามิเตอร์

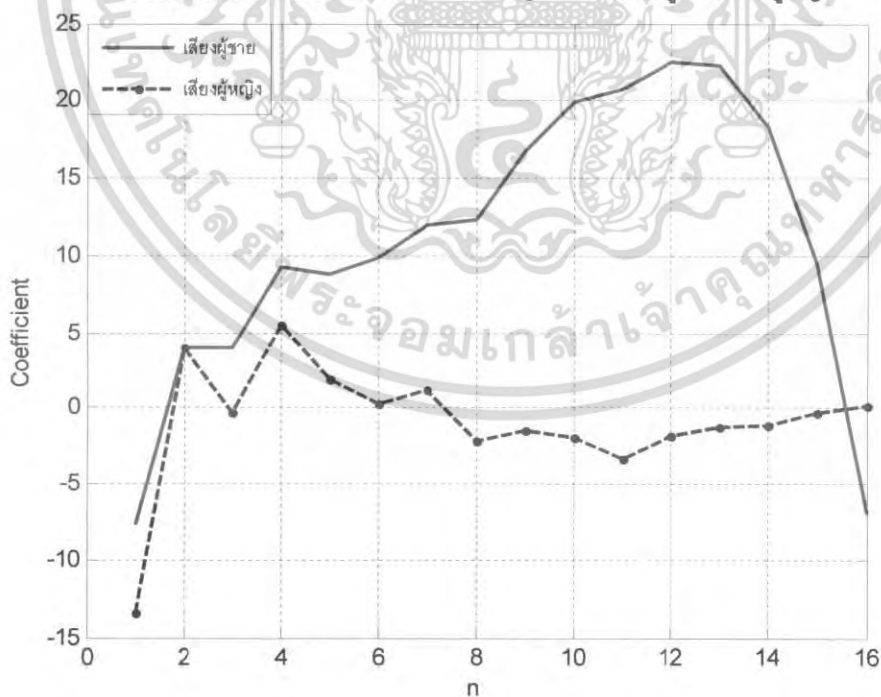
กราฟแสดงการเปรียบเทียบสัมประสิทธิ์ Cepstrum ของเสียงผู้ชายและผู้หญิง



รูปที่ 4.7 ค่าสัมประสิทธิ์ Cepstrum ของเสียงผู้ชายและผู้หญิงพูด "0" แสดงผลการทดลอง ของโปรแกรม

ย่อยที่ทำการหาสัมประสิทธิ์ Cepstrum

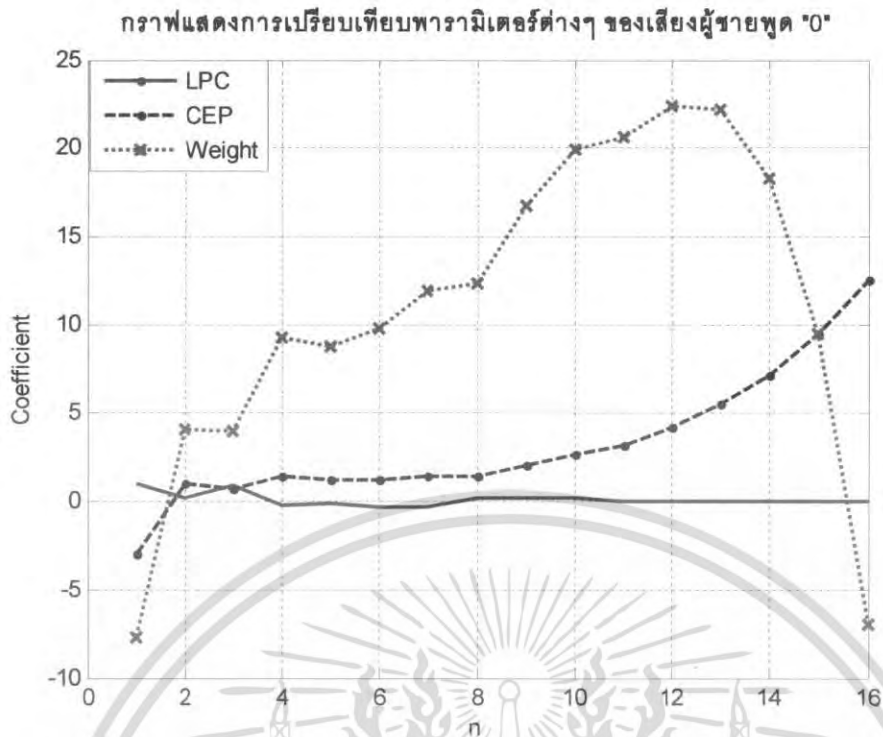
กราฟแสดงการเปรียบเทียบพารามิเตอร์ Weight ของเสียงผู้ชายและผู้หญิง



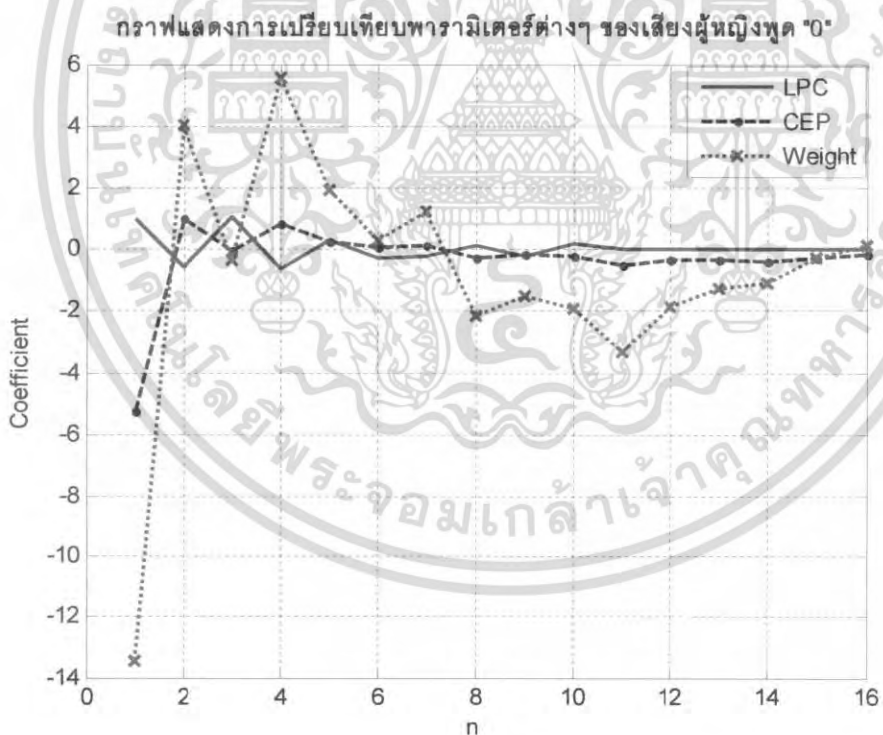
รูปที่ 4.8 ค่าสัมประสิทธิ์ Weight ของเสียงผู้ชายและผู้หญิงพูด "0" แสดงผลการทดลอง ของโปรแกรม

ย่อยที่ทำการหาพารามิเตอร์ Weight

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



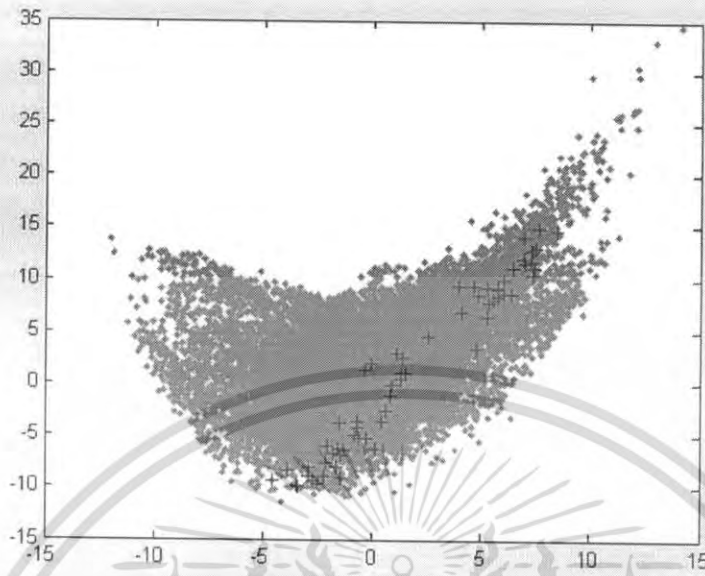
รูปที่ 4.9 ทำการเปรียบเทียบพารามิเตอร์ LPC , Cepstrum, Weight ของเสียงผู้ชายพูด "0" ในเฟรมแรก



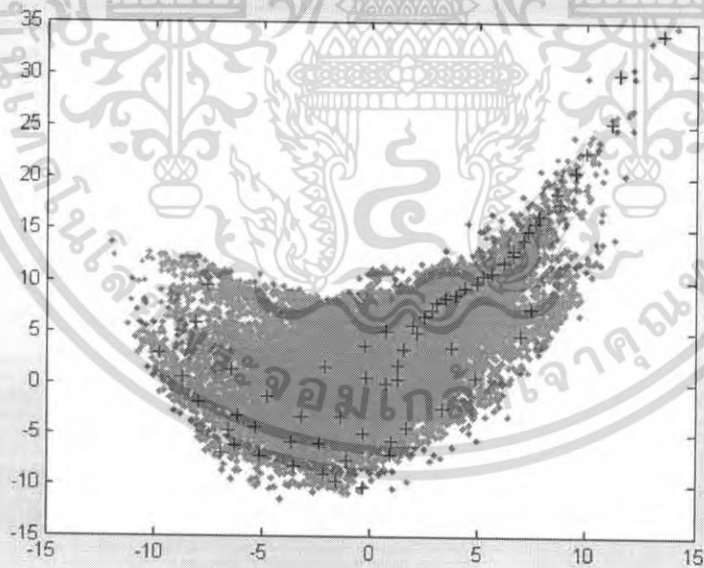
รูปที่ 4.10 ทำการเปรียบเทียบพารามิเตอร์ LPC , Cepstrum, Weight ของเสียงผู้หญิงพูด "0" ในเฟรมแรก

จากรูป 4.9 – 4.10 จะแสดงให้เห็นถึงความแตกต่างกันระหว่างค่าพารามิเตอร์ของเสียงที่ทำการคำนวณได้ตามลำดับซึ่งจะมีความแตกต่างกันอยู่ โดยพารามิเตอร์ที่แทนคุณลักษณะเด่นของเสียงขั้นสุดท้ายก็คือ พารามิเตอร์ Weight ซึ่งจะนำไปใช้ในขั้นตอน Vector Quantization ต่อไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลจากโปรแกรมสร้างโค้ดบุค



รูปที่ 4.11 เวกเตอร์พารามิเตอร์ของเฟรมต่างๆกับโค้ดบุคที่สุ่มได้ใน 2 มิติ ดังนั้นค่าจุดศูนย์กลางของกลุ่มจึงยังรวมตัวกันอยู่และกระจายครอบคลุม อินพุท ไม่ทั่วถึง

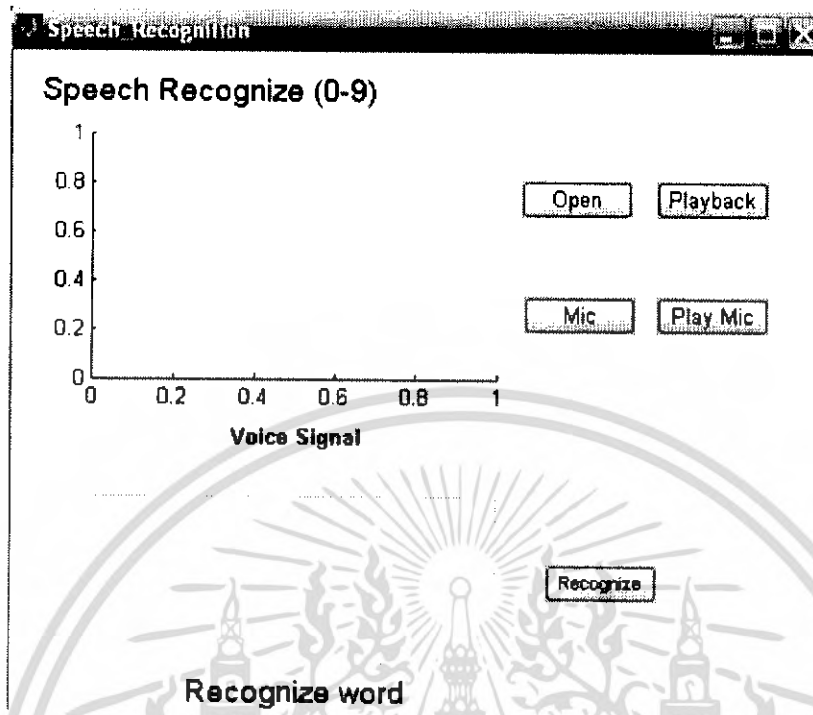


รูปที่ 4.12 เวกเตอร์พารามิเตอร์ของเฟรมต่างๆกับโค้ดบุคที่คำนวณได้ใน 2 มิติ ได้ผ่านกระบวนการ K-mean algorithm แล้วทำให้ codebook ที่ได้ กระจายและสามารถครอบคลุมอินพุทที่เข้ามาได้ทั่วถึงมากขึ้น

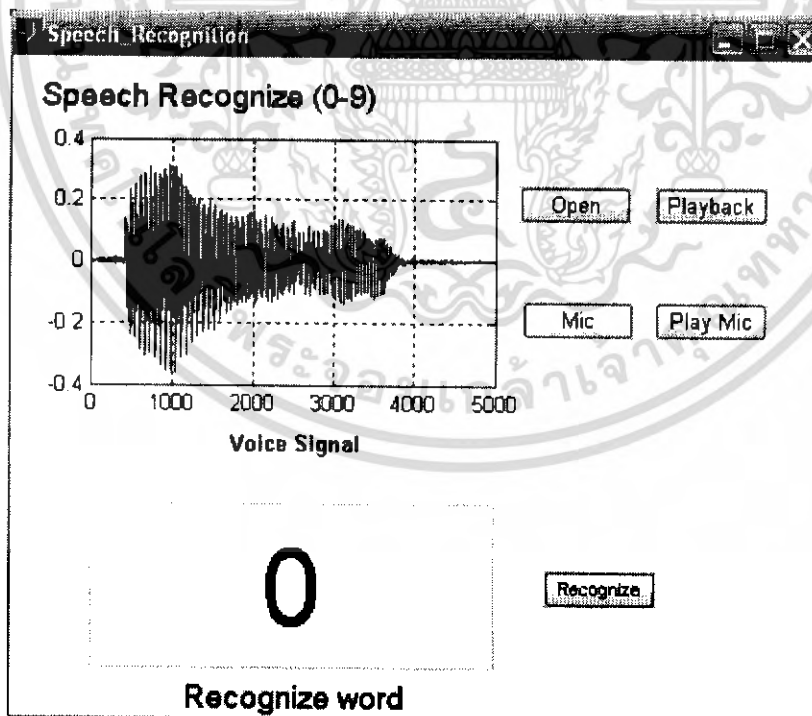
- แทนเวกเตอร์พารามิเตอร์ใน 1 เฟรม + แทนจุดศูนย์กลางของกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดลองระบบการรู้จำเสียง



รูปที่ 4.13 หน้าต่างโปรแกรม Speech Recognition



รูปที่ 4.14 หน้าต่างโปรแกรม Speech Recognition หลังจากทำการพูดเสียง "0" โดยผู้ชายคนหนึ่ง และ ทำการคลิกปุ่ม Recognize แล้ว ซึ่งได้ผลดังแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 1 การทดสอบการรู้จำเสียงโดยนำเสียงจากผู้พูดที่อยู่ในกลุ่มทดลอง โดยกำหนดให้ผู้พูดแต่ละคนพูดทั้งหมด 9 เสียง พูดเสียงละ 3 ครั้ง มาทดสอบ

เสียงทดสอบ	0	1	2	3	4	5	6	7	8	9	ความถูกต้อง(%)
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 1	0	1	2	3	4	5	0	7	8	9	90
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 2	0	4	0	3	4	5	6	7	8	9	80
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 3	0	1	2	3	4	5	6	7	8	9	100
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 1	0	2	2	8	4	5	2	7	5	9	60
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 2	1	1	2	3	4	5	2	7	5	9	70
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 3	0	1	9	8	4	5	2	7	8	9	70
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 1	1	3	2	3	1	8	2	7	8	9	50
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 2	1	3	2	3	4	5	9	7	8	9	70
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 3	1	1	2	3	4	5	2	7	5	9	70
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 1	0	3	2	5	4	5	6	5	8	9	60
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 2	0	1	2	8	4	5	6	7	8	9	90
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 3	0	1	2	5	4	5	0	7	8	9	80
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 1	1	1	2	5	4	5	9	7	8	9	70
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 2	0	1	9	5	4	5	9	7	8	9	70
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 3	3	1	9	5	4	5	9	7	5	9	50
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 1	0	1	2	3	4	5	6	7	8	9	100
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 2	0	1	2	5	4	5	2	7	8	5	70
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 3	0	1	2	5	4	5	6	7	8	9	90

รวมความถูกต้องเฉลี่ย 74.45%

ค่าที่พูด	ความถูกต้อง(%)	ค่าที่พูด	ความถูกต้อง(%)
0	66.67	5	94.44
1	72.22	6	33.33
2	72.22	7	94.44
3	44.44	8	74.78
4	94.44	9	41.67

ตารางที่ 4.5 ผลการทดสอบการรู้จำเสียง

โดยแสดงค่าความถูกต้องเปรียบเทียบระหว่างเสียงพูดแต่ละเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 2 การทดสอบการรู้จำเสียงโดยนำเสียงจากผู้ทดสอบเป็นเพศชายและไม่ได้อยู่ในกลุ่มทดสอบ โดยกำหนดให้ผู้พูดแต่ละคนพูด ทั้งหมด 9 เสียง ผู้เสียงละ 3 ครั้ง มาทดสอบ

เสียงทดสอบ	0	1	2	3	4	5	6	7	8	9	ความถูกต้อง(%)
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 1	0	2	2	2	4	2	2	7	8	2	50
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 2	0	1	2	3	4	7	6	7	8	9	90
เสียงผู้ชายคนที่ 1 พูดครั้งที่ 3	0	1	9	7	4	7	2	2	7	9	40
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 1	2	1	2	3	4	9	2	1	8	9	60
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 2	1	1	9	3	1	9	2	5	5	9	30
เสียงผู้ชายคนที่ 2 พูดครั้งที่ 3	1	1	2	3	4	3	2	7	8	9	70
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 1	2	1	2	3	7	8	6	7	8	9	70
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 2	0	1	2	3	4	5	6	7	5	9	90
เสียงผู้ชายคนที่ 3 พูดครั้งที่ 3	2	1	2	3	4	8	6	7	8	9	80
เสียงผู้ชายคนที่ 4 พูดครั้งที่ 1	2	1	2	2	4	5	6	7	8	9	80
เสียงผู้ชายคนที่ 4 พูดครั้งที่ 2	0	1	2	9	4	5	6	7	7	2	70
เสียงผู้ชายคนที่ 4 พูดครั้งที่ 3	9	1	2	9	1	5	2	1	8	9	50

รวมความถูกต้องเฉลี่ย 65%

ค่าที่พูด	ความถูกต้อง
0	41.67
1	91.67
2	83.33
3	58.33
4	75.00
5	33.33
6	50
7	66.67
8	66.67
9	83.33

ตารางที่ 4.6 ผลการทดสอบการรู้จำเสียง

โดยแสดงค่าความถูกต้องเปรียบเทียบเมื่อผู้ทดสอบเป็นเพศชาย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้จัดทำเห็นว่าไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรณีที่ 3 การทดสอบการรู้จำเสียงโดยนำเสียงจากผู้ทดสอบ เพศหญิงและไม่ได้อยู่ในกลุ่มทดสอบ โดยกำหนดให้ผู้พูดแต่ละคนพูดทั้งหมด 9 เสียง พูดเสียงละ 3 ครั้ง มาทดสอบ

เสียงทดสอบ	0	1	2	3	4	5	6	7	8	9	ความถูกต้อง(%)
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 1	0	1	2	3	4	5	6	8	8	9	90
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 2	2	1	2	3	4	5	6	7	5	9	80
เสียงผู้หญิงคนที่ 1 พูดครั้งที่ 3	2	1	2	3	4	5	6	7	5	9	80
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 1	0	1	9	9	4	5	6	5	5	9	60
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 2	0	1	9	5	5	5	6	7	9	9	60
เสียงผู้หญิงคนที่ 2 พูดครั้งที่ 3	0	3	9	5	4	5	2	7	5	9	50
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 1	1	1	2	5	4	5	9	7	9	9	60
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 2	2	1	9	9	5	5	9	5	8	9	60
เสียงผู้หญิงคนที่ 3 พูดครั้งที่ 3	0	1	8	5	4	5	2	5	9	8	40
เสียงผู้หญิงคนที่ 4 พูดครั้งที่ 1	1	1	9	9	4	5	2	8	8	9	50
เสียงผู้หญิงคนที่ 4 พูดครั้งที่ 2	1	1	9	5	3	5	2	8	8	9	40
เสียงผู้หญิงคนที่ 4 พูดครั้งที่ 3	0	1	9	9	1	5	2	5	8	9	50

รวมความถูกต้องเฉลี่ย 60%

ค่าที่พูด	ความถูกต้อง
0	50
1	91.67
2	33.33
3	25
4	66.67
5	100
6	41.67
7	41.67
8	41.67
9	91.67

ตารางที่ 4.7 ผลการทดสอบการรู้จำเสียง

โดยแสดงค่าความถูกต้องเปรียบเทียบเมื่อผู้ทดสอบเป็นเพศหญิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์การทดลอง

จากการศึกษาในทางทฤษฎี และ ทำการทดลอง ตามขอบเขตของปริญญาโทฉบับนี้ เราได้เรียนรู้ว่าการเข้ารหัสเสียงมีหลายประเภท อันได้แก่ เวฟฟอร์มควอนไทเซชัน (Wave form quantization) และ พารามेटริกควอนไทเซชัน (Parametric quantization) ซึ่งแต่ละชนิดก็เหมาะกับการประยุกต์ใช้งานต่างๆ กัน โดยในโครงการนี้ เราจะใช้การเข้ารหัสแบบพารามेटริกควอนไทเซชัน (LPC) เนื่องจาก ข้อมูลที่ใช้ส่งไปที่ปลายทางมีขนาดเล็ก แต่มีข้อเสีย คือ ต้องอาศัยทฤษฎีในการคำนวณมาก

การประมาณค่าแบบเชิงเส้น (LPC) คือ การประมาณเสียงจากผลรวมเชิงเส้นของสัญญาณเสียงในอดีต โดยอาศัย หลักการของค่ากำลังสองของสัญญาณความคลาดเคลื่อนให้มีค่าต่ำที่สุด โดยในการหาสัมประสิทธิ์การประมาณเชิงเส้น กล่าวคือ จะทำการหาค่าอโตคอร์รีเลชัน เพื่อ นำไปทำการหาสัมประสิทธิ์ LPC และ อัตราการขยาย ที่เหมาะสม เพื่อเก็บเป็นคุณลักษณะเด่นของแต่ละเสียง

ในส่วนของการสร้างแบบอ้างอิง (Code Book) ในปริญญาโทฉบับนี้เลือกใช้หลักการจัดระดับเวกเตอร์ (Vector Quantization) โดยมีหลักการว่าดังนี้คือ จะนำพารามิเตอร์ที่ได้จากขั้นตอนของการประมาณเชิงเส้นมาทำการควอนไทซ์แบบเวกเตอร์กับจุดศูนย์กลางของโหนดบुकที่สุ่มขึ้น และจะนำค่าความคลาดเคลื่อนที่ได้มาทำการจัดกลุ่มของพารามิเตอร์เหล่านั้น จากนั้นจะทำการคำนวณหาค่าจุดศูนย์กลางของกลุ่มใหม่และจะนำค่าของจุดศูนย์กลางใหม่และค่ามาทำการหาค่าความคลาดเคลื่อน และพิจารณาว่าค่าความคลาดเคลื่อนมีค่าน้อยกว่า 0.001 หรือยังหากยังก็จะกลับไปทำขั้นตอนด้านบนใหม่ทั้งหมดจนกว่าค่าความคลาดเคลื่อนจะมีค่าน้อยกว่า 0.001 จึงจะหยุดการทำงาน ซึ่งค่าจุดศูนย์กลางสุดท้ายที่ได้ก็คือค่าของแบบอ้างอิง (Code Book) ที่ต้องการ ซึ่งค่านี้จะถูกนำมาใช้ในการหาค่า Index ของเสียงแต่ละเสียง ค่า Index นี้จะนำไปใช้ในส่วนการคำนวณของ Hidden Markov Model ต่อไป

ในส่วนของ บอร์ดประมวลผล TMS320C6713 จะเป็นการศึกษาการทำงานและโครงสร้างของบอร์ด รวมทั้งศึกษาการใช้งานโปรแกรม Code Composer Studio เพื่อใช้ในการ implement การวิเคราะห์ feature attraction ของเสียง ปัญหาที่พบก็คืออัตราความเร็วประมวลผลของบอร์ดมีอัตราความเร็วอยู่ที่ประมาณ 225 MHz ซึ่งน้อยกว่าความเร็วในการประมวลผลของคอมพิวเตอร์ในปัจจุบัน ดังนั้นเวลาในการประมวลผลตามประสิทธิภาพ LPC จึงจะใช้เวลามากกว่าการหาสัมประสิทธิ์จากคอมพิวเตอร์โดยตรงปัญหาที่พบอีกประการหนึ่งก็คือ หน่วยความจำในบอร์ดมีค่อนข้างจำกัดจึงไม่สามารถที่จะประมวลผล input ที่มีจำนวนมากๆ ได้ จึงต้องมีการลดจำนวน อินพุต ในการคำนวณแต่ละรอบ ซึ่งอาจทำให้ผลการทดลองคลาดเคลื่อนได้

ในส่วนของการสร้างแบบจำลองและตัดสินใจ (Hidden Markov Model) จะเป็นขั้นตอนที่สร้างแบบจำลองของเสียงขึ้นมาและทำการตัดสินใจว่าเสียงที่เข้ามาใหม่นั้นมีแบบจำลองใกล้เคียงกับแบบจำลองใดในฐานข้อมูล โดยจะใช้หลักการของ Probability เข้ามาใช้ ปัญหาของการศึกษาในส่วนนี้ก็คือ ทฤษฎีที่นำมาใช้มีเนื้อหาที่ค่อนข้างซับซ้อนจึงต้องใช้เวลาในการศึกษาอย่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการจำลองการทำงาน(Simulation) ด้วยโปรแกรม MATLAB ซึ่งผลที่ได้นั้นมีปัจจัยหลายอย่างที่เกี่ยวข้องซึ่งมีผลต่อการนำมาวิเคราะห์เสียงพูดเช่น ความดังความเบาของเสียงพูด, ความแหลมหรือต่ำของเสียงพูด, เพศและอายุของผู้พูด, สภาพแวดล้อมที่พูด เป็นต้น



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

- [1] Lawrence Rabiner ,Bimng-Hwang Juang , “Fundamentals Of Speech Recognition” , “Prentice Hall Signal Processing Series” , Alan V. Oppenheim, Series editor
- [2] Rulph Chassaing, “Digital Signal Processing and Applications with the C6713 and C6416 DSK” , A John Wiley & Sons, INC., Publication
- [3] Allen Gersho, Robert M.Gray, “Vector Quatization and Signal Compression ”
- [4]Charles W. Therrien, ”Discrete Random Signals And Statistical Signal Processing”,Pentice Hall Intemational,Inc
- [5]Sanjit K.Mitra ,”Digital Signal Processing A Computer-Based Approach”,McGraw-Hill
- [6]Monson H. Hayes “Statistical Digital Signal Processing And Modeling”,John Wiley & Sons,Inc.
- [7]Levinson , S.E., Rabiner , L.R. ,and Sondhi,M.M. “An introduction to the application of the theory of probability functions of a Markov process to automatic speech recognition.” The Bell system Technical Journal 62(April 1983)
- [8]นิรุช อำนวนศิลป์ “คู่มือเขียนโปรแกรมด้วยภาษา C ” สำนักพิมพ์ Provision
- [9]สุเชียร เกียรติสุนทร. “การประมาณพหุระเชิงเส้นเสียงพูด (Linear Prediction of Speech) สาขาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง,2525.” วิทยานิพนธ์ปริญญา มหบัณฑิต สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง,2525.
- [10]เสาวลักษณ์ อารีย์พงศา “การรู้จำเสียงพูดตัวเลขเป็นภาษาไทยแบบไม่ขึ้นกับผู้พูดโดยวิธี ฮิดเดน มาร์คอฟ โมเดล และ เวกเตอร์ควอนไทซ์เซชัน สาขาวิศวกรรมไฟฟ้า จุฬาลงกรณ์มหาวิทยาลัย,2538” วิทยานิพนธ์ปริญญา มหบัณฑิต.จุฬาลงกรณ์มหาวิทยาลัย,2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ภาคผนวกนี้ได้แบ่งออกเป็น 4 ภาคผนวกด้วยกัน โดยแบ่งออกเป็น

ภาคผนวก ก. เป็นการรวบรวมข้อมูลที่ใช้ในการพิจารณากำหนดค่าพารามิเตอร์ที่ใช้ในการทดลอง

ภาคผนวก ข. เป็นการศึกษาการใช้บอร์ด TMS320C6713

ภาคผนวก ค. เป็นส่วนที่รวบรวม โค้ดโปรแกรมต่างที่ใช้ในการคำนวณและวิเคราะห์โดยแบ่งออกเป็น

- โปรแกรมทดสอบคำนวณหาค่าพารามิเตอร์ต่างๆ ของคุณลักษณะของเสียงจาก MATLAB
- โปรแกรมเตรียมสัญญาณเสียง
- โปรแกรมหาลักษณะเด่นของเสียงจากบอร์ด TMS320C6713
- โปรแกรมสร้าง Code Book
- โปรแกรมหาค่า Index

โดยโปรแกรมทั้งหมดที่ใช้ในการคำนวณและการวิเคราะห์นั้นใช้โปรแกรม MATLAB ในการเขียนแต่ จะมี “โปรแกรมหาลักษณะเด่นของเสียงจากบอร์ด TMS320C6713” จะใช้โปรแกรม Code Composer Studio ซึ่งใช้โครงสร้างของภาษา C

ภาคผนวก ง. รวบรวมผลของ Code Book ที่คำนวณได้

ภาคผนวก ก

การพิจารณาเลือกใช้ค่าพารามิเตอร์ที่เหมาะสมเพื่อใช้ในการทดลอง

1 จำนวน order ของ LPC

มีการวิจัยการประมาณเชิงเส้นเสียงพูด (สุธีธร เกียรติสุนทร 2525) ซึ่งเป็นงานวิจัยที่มีการนำค่าสัมประสิทธิ์ การสะท้อนที่หาโดยวิธีออตสัมพันซ์ มาทำการสังเคราะห์เสียงขึ้นมาใหม่ จากการทดสอบดังกล่าว ให้ผลเป็นที่น่าพอใจ งานวิจัยชิ้นนี้มีการใช้ LPC ด้วยจำนวน order 10 ซึ่งเพียงพอสำหรับการหาสัมประสิทธิ์การสะท้อนเพื่อการสังเคราะห์เสียงที่ต้องใช้ความละเอียดมาก ในวิทยานิพนธ์นี้เป็นการรู้จำเสียงเท่านั้นย่อมต้องการความละเอียดในการวิเคราะห์น้อยกว่าการสังเคราะห์เสียง ดังนั้น order 10 จึงเพียงพอสำหรับการรู้จำเสียง

2 ขนาดของ codebook

ในการทดสอบหาค่าขนาดของ codebook ที่เหมาะสม (Rabiner, Levinson, Sondhi, 1982) เป็นการทดสอบโดยใช้เวกเตอร์ของ LPC จำนวน 39708 เวกเตอร์ จากคน 100 คน (ชาย 50 หญิง 50) พูคตัวเลขภาษาอังกฤษ 0-9 จากนั้นนำเวกเตอร์มาหาค่าควอนไทซ์ที่ codebook ขนาดต่างๆ $M = 2, 4, 8, 16, 32, 64, 128$ ใช้ในการพิจารณาดังนี้

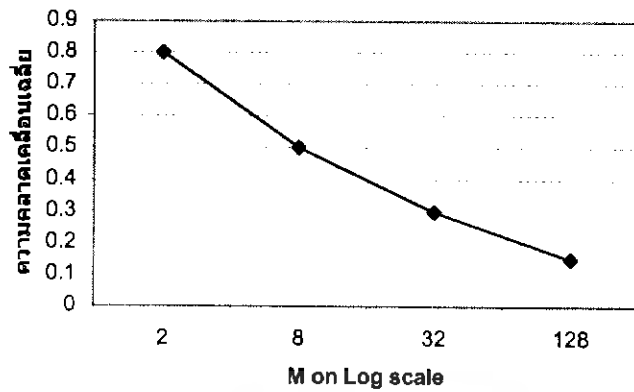
ค่าความคลาดเคลื่อนเฉลี่ย $\|D_m\|$

$$\|D_m\| = \min_{\hat{a}_m} \left\{ \frac{1}{I} \sum_{i=1}^I \min \left[d(\hat{a}_m, a_i) \right] \right\}; 1 \leq m \leq M \quad (ก.1)$$

โดย a_i เป็นเวกเตอร์ LPC $i = 1, 2, 3, \dots, I$ \hat{a}_m เป็น codebook LPC เวกเตอร์ $m = 1, 2, \dots, M$

จากผลการทดสอบจะได้ดังรูป ก.1 โดยรูป ก.1 เป็นการพล็อต ค่า $\|D_m\|$ กับค่า M (ขนาด codebook โดยใช้ log scale) โดย M มีค่า 2-128 เราจะเห็นว่าที่ $M \geq 32$ ค่าความคลาดเคลื่อนเฉลี่ยจะต่ำกว่า 0.3 $M = 64$ จะมีค่า $\|D_m\|$ ประมาณ 0.2 ซึ่งค่า $\|D_m\| < 0.3$ เป็นค่าความคลาดเคลื่อนที่น้อยมากสำหรับ vector quantization (Levinson, Rabiner, et al. 1979) ดังนั้นสำหรับวิธีของ HMM ควรใช้ $M = 64$ เพราะจะมีค่าความคลาดเคลื่อนเฉลี่ยน้อยเมื่อ $M = 64 - 128$ (Rabiner, Levinson, Sondhi, 1982)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

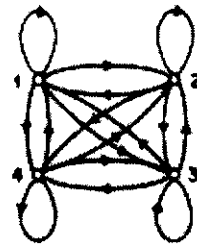


รูปที่ ก.1 เปรียบเทียบความคลาดเคลื่อนเฉลี่ยกับขนาด codebook

3. โครงสร้างของ HMM (Hidden Markov Model)

เป็นที่ทราบกันแล้วว่ามีพารามิเตอร์ต่างๆ ที่มีผลต่อความถูกต้องในการวิเคราะห์ รูปแบบโครงสร้างของ HMM ก็เป็นอีกส่วนหนึ่งที่มีผลกระทบสำคัญ

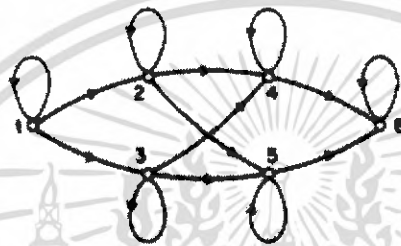
จากรูป ก.2 ได้แสดงโครงสร้างของ HMM แบบต่างๆ รูป ก.2 (a) เป็น Fully connect HMM ที่มี 4 state รูป ก.2(b) เป็น constrained serial ที่มี 4 state รูป ก.2 (c) เป็น constrained parallel ที่มี 6 state ใช้โครงสร้างทั้ง 3 โครงสร้าง ผลที่ได้คือมีค่าความคลาดเคลื่อนใกล้เคียงกันทั้ง 3 โครงสร้าง ($3.5\% \pm 0.1\%$) จึงสรุปได้ว่าโครงสร้างแบบ serial ก็เพียงพอในการพิจารณา (Rabiner, Levinson, Sondhi, 1982) อีกทั้งการใช้โครงสร้างแบบ serial ยังมีความซับซ้อนน้อยกว่าโครงสร้างแบบอื่นๆ อีกด้วย



(a)



(b)

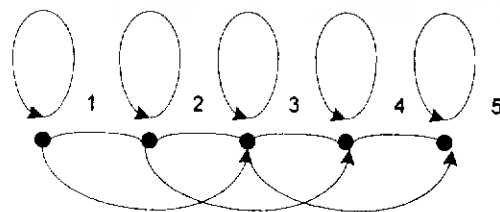


(c)

รูป ก.2 แสดงโครงสร้างของ HMM

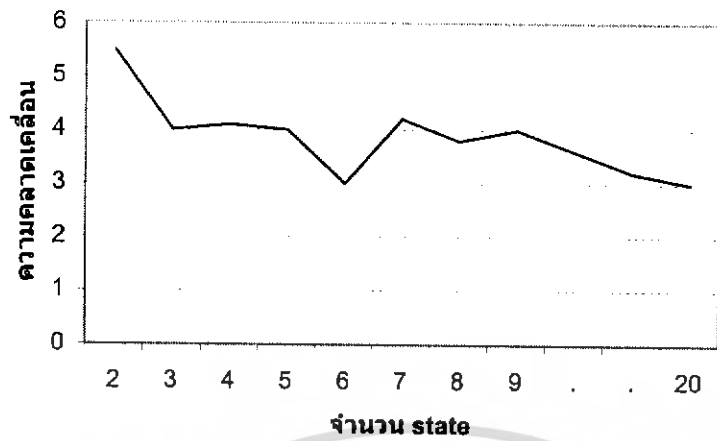
4. จำนวน state

จากรูป ก.3 เป็นโครงสร้างแบบ serial ที่มี 5 state ที่จะใช้พิจารณา นำโครงสร้างนี้มาหาโมเดล โดยใช้จำนวน state ตั้งแต่ 2-9 และ 20 โดยจะทดสอบด้วยเซตทดสอบเดียวกัน ผลการทดสอบเป็นดังรูป ก.4 สามารถกล่าวได้ว่าความสัมพันธ์ของความถูกต้องในการวิเคราะห์ จำนวนเสียงที่รู้จำและจำนวน state ไม่ใช่ความสัมพันธ์ ที่ชัดเจน (Rabiner, Levinson, Sondhi, 1982) แต่จากการสังเกตจากรูปที่ ก.4 แล้ว จำนวน state เป็น 6 จะ มีความคลาดเคลื่อนน้อยที่สุด เมื่อเปรียบเทียบกับ จำนวน state ใกล้เคียงกัน ใน วิทยานิพนธ์นี้จึงเลือกใช้จำนวน state เป็น 6 ในการทดลอง



รูป ก.3 โครงสร้างแบบ serial

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก.4 เปรียบเทียบความคลาดเคลื่อนกับจำนวน state

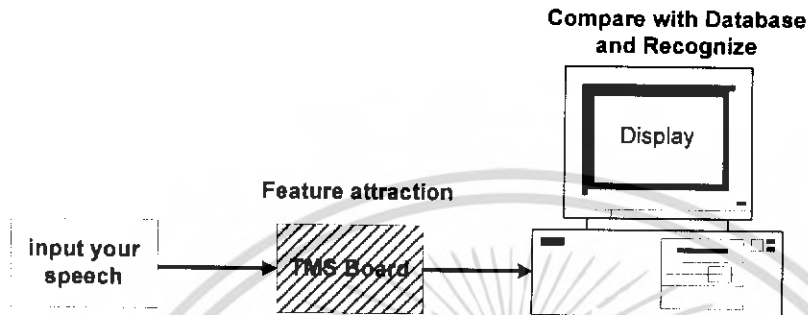


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข.

การศึกษาการใช้บอร์ด TMS320C6713 เบื้องต้น

1.) การออกแบบระบบรู้จำเสียงพูดด้วย TMS320C6713



รูปที่ ข.1 บล็อกไดอะแกรมแสดงขั้นตอนการทำงาน

ในการออกแบบระบบรู้จำเสียงพูด เราจะใช้ บอร์ด TMS320C6713 มาใช้ในการหา ลักษณะเด่นของเสียง (Feature Attraction) ซึ่งจะใช้การเขียนโปรแกรม แล้วจึงทำการload ลงไปในบอร์ด โดยโปรแกรมที่เขียนขึ้น จะเขียนขึ้นด้วย อัลกอริทึม ของภาษา C โปรแกรมที่เขียนขึ้นจะประกอบไปด้วย

- การทำ Pre-emphasis โดยใช้ Highpass Filter
- การแบ่งเฟรมข้อมูล
- การทำ Windowing
- การหาลักษณะเด่นของเสียง โดย อัลกอริทึม ของ LPC (Linear Prediction Coding)

1.1) เกี่ยวกับ บอร์ด TMS320C6713

บอร์ดประมวลผล TMS320C6713 เป็นผลิตภัณฑ์ของบริษัท Texas Instrument (TI) ที่ใช้หน่วยประมวลผล เป็น chip DSP ตระกูล TMS320C6X ซึ่งจะมีโครงสร้างเป็นสถาปัตยกรรมแบบ very-long-instruction-word (VLIW) และทำการ operate ที่ความถี่ 225 MHz

Chip ตระกูล C67xx (C6701 ,C6711 และ C6713) จะเป็นโครงสร้าง แบบ floating point processor ซึ่งทำการประมวลผลแบบเลขทศนิยม ขณะที่ ตระกูล C62XX และ C64XX จะเป็นแบบ Fix point processor ดังนั้น บอร์ด TMS320C6713 จึงทำการประมวลผลได้ทั้งแบบ Fix point และ Floating point

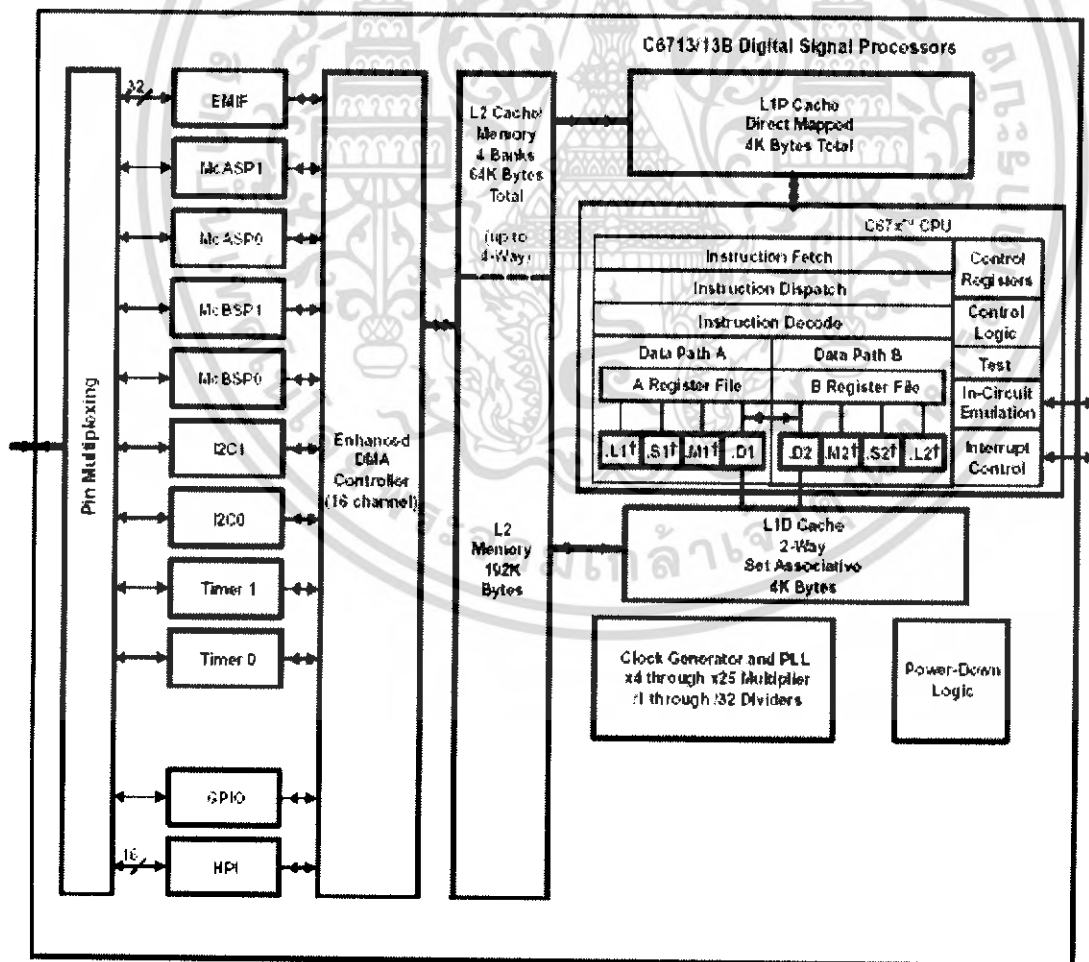
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บอร์ดประมวลผล TMS320C6713 มีความสามารถที่หลากหลาย ในการประมวลผลแบบ Digital Signal Processing จึงเหมาะแก่การนำไปประยุกต์ใช้งานกับ application ที่หลากหลาย ไม่ว่าจะเป็น ด้าน Audio Processing , Speech Processing , Image processing เป็นต้น

1.2) หน่วยความจำภายใน(Internal Memory) และ Memory Map

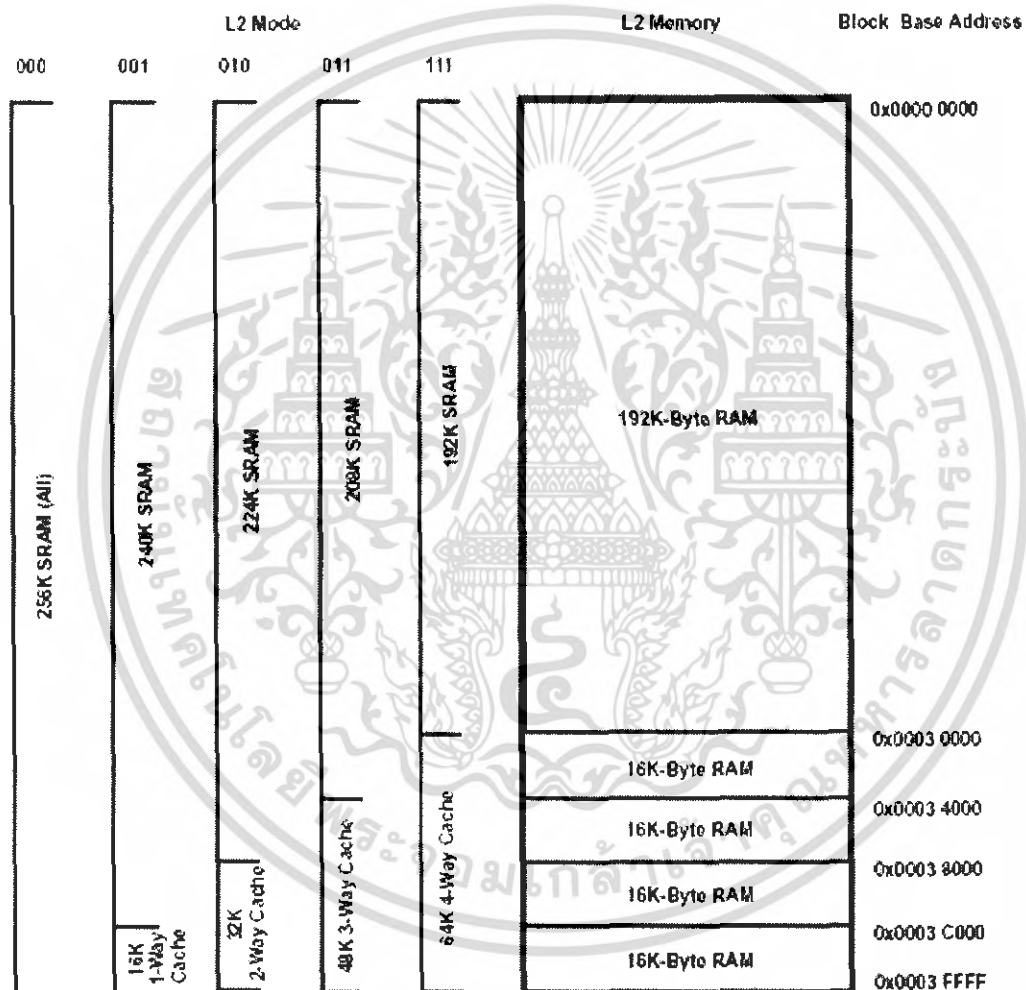
หน่วยความจำภายใน บอร์ดประกอบด้วย หน่วยความจำแคช 2 ระดับ เป็น แคช L1 ที่ใช้เก็บใน ส่วนโปรแกรม 4 kB แคช L1 ที่เก็บส่วน data 4 kB และส่วนของ แคช L2 ที่ใช้ร่วมกันระหว่างโปรแกรม และข้อมูลขนาด 256 kB

บัสภายในประกอบด้วย โปรแกรม แอดเดรสบัสขนาด 32 บิต , โปรแกรม คำสั่งบัสขนาด 256 บิต เพื่อให้สามารถจัดให้จัดให้เพียงพอกับคำสั่งขนาด 32 บิตได้ 8 คำสั่ง , คำสั่งแอดเดรสบัสขนาด 32 บิต 2 เส้น , คำสั่งบัส ขนาด 64 บิต 2 เส้น และ สตอร์จ่าบัสขนาด 64 บิต 2 เส้น ด้วยแอดเดรสบัสขนาด 32 บิต พื้นที่ของหน่วยความจำภายนอกอีก 4 อัน คือ CE0,CE1,CE2 และ CE3 รูปที่ 2 แสดง บล็อกไดอะแกรม ของฟังก์ชันของตัวประมวลผล C6713



รูปที่ ข.2 บล็อกไดอะแกรมของฟังก์ชันของตัวประมวลผล C6713
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวประมวลผล C6713 บนบอร์ด DSK ประกอบด้วย หน่วยความจำภายในขนาด 264 kB ซึ่งเริ่มที่ address 0X00000000 และ 16 MB ของ SDRAM ภายนอก ซึ่งจะถูกระบุไปยัง CE0 โดยเริ่มที่ address 0X80000000 หน่วยความจำของบอร์ด DSK นี้ ยังรวมไปถึง หน่วยความจำแฟลชขนาด 512 kb(พร้อมให้ใช้งาน 256 kb) ซึ่งจะถูกระบุไปยัง CE1 โดยเริ่มต้นที่ address 0X90000000 รูปที่ ข.3 แสดงโครงสร้างของ แคล L2 internal memory การประกาศตัวแปรหรือค่าตัวแปรต่าง ๆ ที่ทำการประมวลผลนั้นอาจจะ ถูกเก็บอยู่ในหน่วยความจำภายนอกหรือหน่วยความจำแฟลช จึงต้องมีการ map memory ลงไปเก็บ ใน internal memory ดังแสดง รูปที่ ข.5 ที่แสดงการ map memory ของ address ต่าง ๆ



รูปที่ ข.3 โครงสร้าง internal memory ของ บอร์ด TMS320C6713

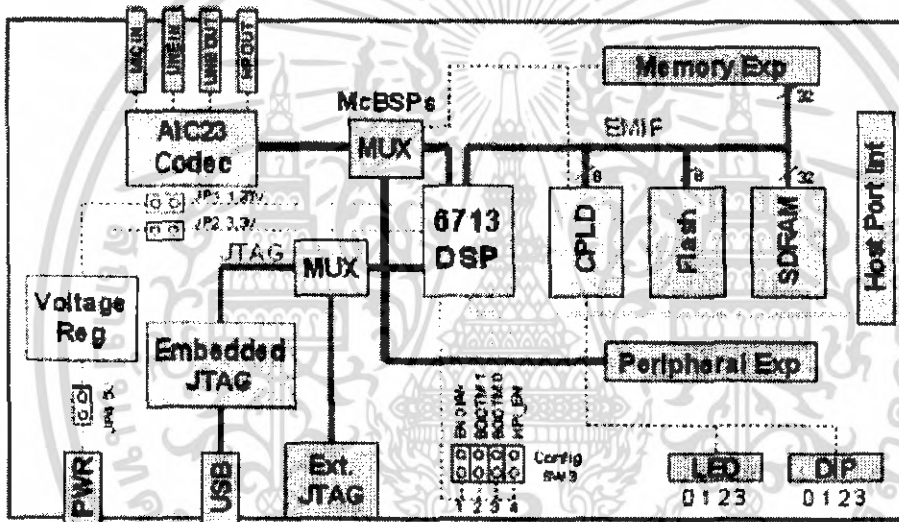
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3) Board Interface

บอร์ดประมวลผล TMS320C6713 จะทำการเชื่อมต่อ กับ computer ผ่านทาง USB interface และจะมีพอร์ตสำหรับ input/output อยู่ 4 พอร์ต ได้แก่

- 1 Line in ใช้ในการรับ สัญญาณ input เช่น สัญญาณจาก function generator
- 2 Mic in ใช้ในการรับ สัญญาณเสียง จากไมโครโฟน
- 3 Line out ใช้ในการแสดงผลสัญญาณ output เช่น แสดงผลออกไปยัง Oscilloscope
- 4 Headphone out ใช้สำหรับนำสัญญาณเสียง output ออกไปยังลำโพง

ในการประมวลผลของบอร์ดนั้น จะเป็นแบบ Digital Signal Processing จึงต้องมีการแปลงจากสัญญาณ analog ไปเป็น digital และ จาก digital ไปเป็น analog ซึ่งส่วนที่จัดการในส่วนนี้ก็คือ วงจร AIC23 Codec ที่มีอยู่บนบอร์ดอยู่แล้ว



รูปที่ ข.4 โครงสร้างของบอร์ดประมวลผล TMS320C6713

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMORY BLOCK DESCRIPTION	BLOCK SIZE (BYTES)	HEX ADDRESS RANGE
Internal RAM (L2)	192K	0000 0000 - 0002 FFFF
Internal RAM/Cache	54K	0003 0000 - 0003 FFFF
Reserved	24M - 256K	0004 0000 - 017F FFFF
External Memory Interface (EMIF) Registers	256K	0180 0000 - 0183 FFFF
L2 Registers	128K	0184 0000 - 0185 FFFF
Reserved	128K	0186 0000 - 0187 FFFF
HPI Registers	256K	0188 0000 - 018B FFFF
McBSP 0 Registers	256K	018C 0000 - 018F FFFF
McBSP 1 Registers	256K	0190 0000 - 0193 FFFF
Timer 0 Registers	256K	0194 0000 - 0197 FFFF
Timer 1 Registers	256K	0198 0000 - 019B FFFF
Interrupt Selector Registers	512	019C 0000 - 019C 01FF
Device Configuration Registers	4	019C 0200 - 019C 0203
Reserved	256K - 512	019C 0204 - 019F FFFF
EDMA RAM and EDMA Registers	256K	01A0 0000 - 01A3 FFFF
Reserved	768K	01A4 0000 - 01AF FFFF
GPIO Registers	16K	01B0 0000 - 01B0 3FFF
Reserved	240K	01B0 4000 - 01B3 FFFF
I2C0 Registers	16K	01B4 0000 - 01B4 3FFF
I2C1 Registers	16K	01B4 4000 - 01B4 7FFF
Reserved	16K	01B4 8000 - 01B4 BFFF
McASP0 Registers	16K	01B4 C000 - 01B4 FFFF
McASP1 Registers	16K	01B5 0000 - 01B5 3FFF
Reserved	160K	01B5 4000 - 01B7 BFFF
PLL Registers	8K	01B7 C000 - 01B7 DFFF
Reserved	264K	01B7 E000 - 01BB FFFF
Emulation Registers	256K	01BC 0000 - 01BF FFFF
Reserved	4M	01C0 0000 - 01FF FFFF
QDMA Registers	52	0200 0000 - 0200 0033
Reserved	16M - 62	0200 0034 - 02FF FFFF
Reserved	720M	0300 0000 - 2FFF FFFF
McBSP0 Data Port	84M	3000 0000 - 33FF FFFF
McBSP1 Data Port	84M	3400 0000 - 37FF FFFF
Reserved	84M	3800 0000 - 3BFF FFFF
McASP0 Data Port	1M	3C00 0000 - 3C0F FFFF
McASP1 Data Port	1M	3C10 0000 - 3C1F FFFF
Reserved	1G + 62M	3C20 0000 - 7FFF FFFF
EMIF CE0†	256M	8000 0000 - 8FFF FFFF
EMIF CE1†	256M	9000 0000 - 9FFF FFFF
EMIF CE2†	256M	A000 0000 - AFFF FFFF
EMIF CE3†	256M	B000 0000 - BFFF FFFF
Reserved	1G	C000 0000 - FFFF FFFF

รูปที่ ๕.5 ตาราง memory map

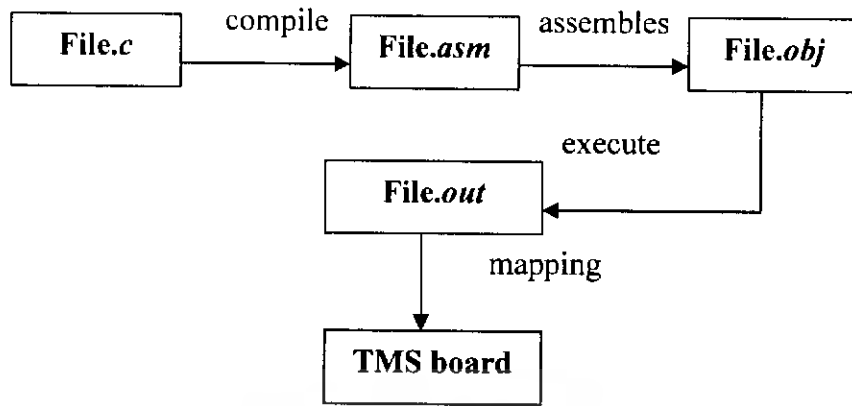
1.4) โปรแกรม Code Composer Studio (CCS)

โปรแกรม CCS เป็นโปรแกรมที่มากับ บอร์ด TMS ซึ่งจะรวม tool ต่างๆ เอาไว้ อาทิเช่น tool สำหรับการสร้าง Code ภาษา C, C compiler, assembler และ linker

ขั้นตอนการทำงานจะเป็นดังนี้

- 1 C compiler จะทำการ compile โค้ด ภาษา C (file.c) ที่เขียน ให้เป็น file.asm
- 2 ทำการ assembles file.asm ให้เป็น machine language ซึ่งจะเป็น file.obj เพื่อใช้สำหรับการควบคุม Hardware
- 3 Execute file.obj ให้เป็น file.out ซึ่ง file นี้ จะทำการ map ลงสู่บอร์ด TMS เพื่อทำการประมวลผลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.6 ขั้นตอนการทำงานในการ load ลง TMS board

1.5) Support File

การเขียนโปรแกรมควบคุม Hardware ของ บอร์ดนั้น จะใช้โครงสร้างการเขียนโปรแกรมของ ภาษา C ซึ่งจะต้องทำการสร้าง โปรเจก ใน Code Composer Studio แล้วจึงนำ File.C ที่เขียนขึ้น ใส่เข้าไปรวมทั้งต้องทำการ include Support File เข้าไปด้วย ทุกครั้งเพราะเป็น file พื้นฐาน ที่ต้องใช้ ซึ่ง ได้แก่

1 Initialization/communication File (c6713dskinit.c) จะบรรจุฟังก์ชัน ที่ initial ให้แก่บอร์ด DSK เพื่อที่จะทำการควบคุม การ Codec , การติดต่อผ่าน Port Input/Output ซึ่งไม่ได้ถูกบรรจุอยู่ใน โปรแกรม CCS

2 Header File (c6713dskinit.h) จะบรรจุฟังก์ชัน Prototype ให้แก่ register ใน AIC23 Codec ซึ่งจะใช้ในการปรับ กำลังขยายของ Input หรือ enable ให้รับ input จาก port Mic IN

- การ enable port Mic IN ทำโดย การเปลี่ยนค่า register 4 จาก 0X0011 เป็น 0X0015
- การเพิ่มกำลังขยายของ Input ทำโดย การเปลี่ยนค่า register 0 จาก 0X0017 เป็น 0X0011

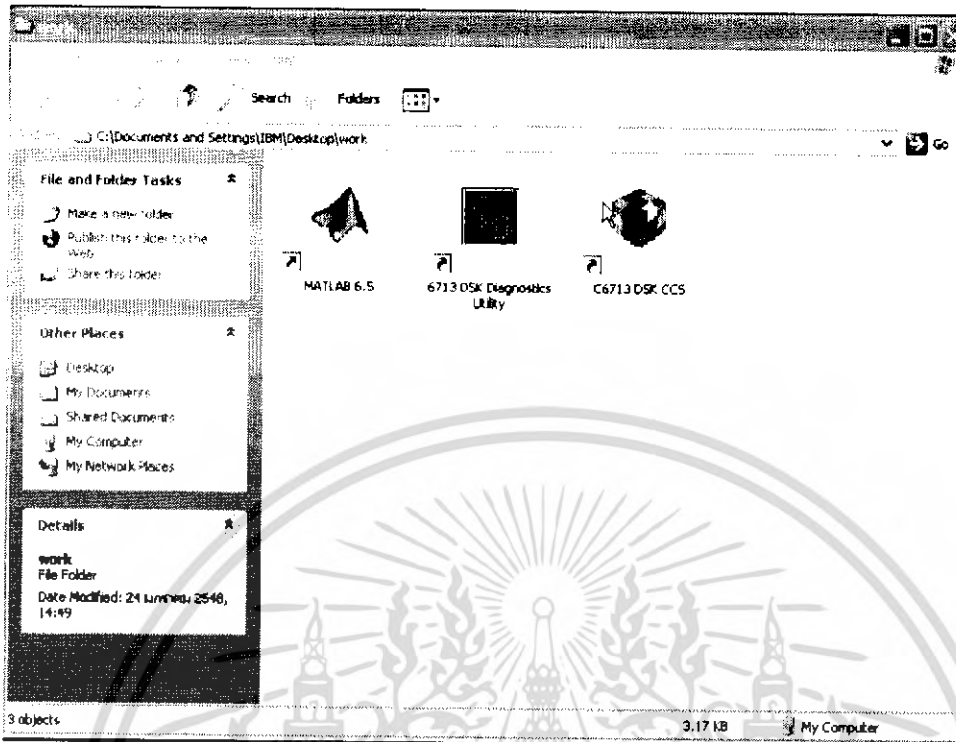
ซึ่งถ้าต้องการให้รับ input จาก port Line in ทำโดย เปลี่ยนค่า register 4 ให้เป็น 0X0011 เหมือนเดิม ก็จะสามารถรับ input จาก function generator ผ่าน port Line in ได้

3 Linker Command File (c6713dsk.cmd) ใช้ในการ Map memory จาก Parameter ที่เราทำการ กำหนดค่าในโปรแกรม เข้าไป address ใน Internal Memory

4 Vector File (vectors_intr.asm) การทำ interrupt จะใช้คำสั่ง c_int11 ซึ่งจะบรรจุอยู่ใน address INT11 ใน file vectors_intr.asm ซึ่งจะใช้คำสั่งในการ โปรแกรมภาษา C ที่ต้องการให้มีการ interrupt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

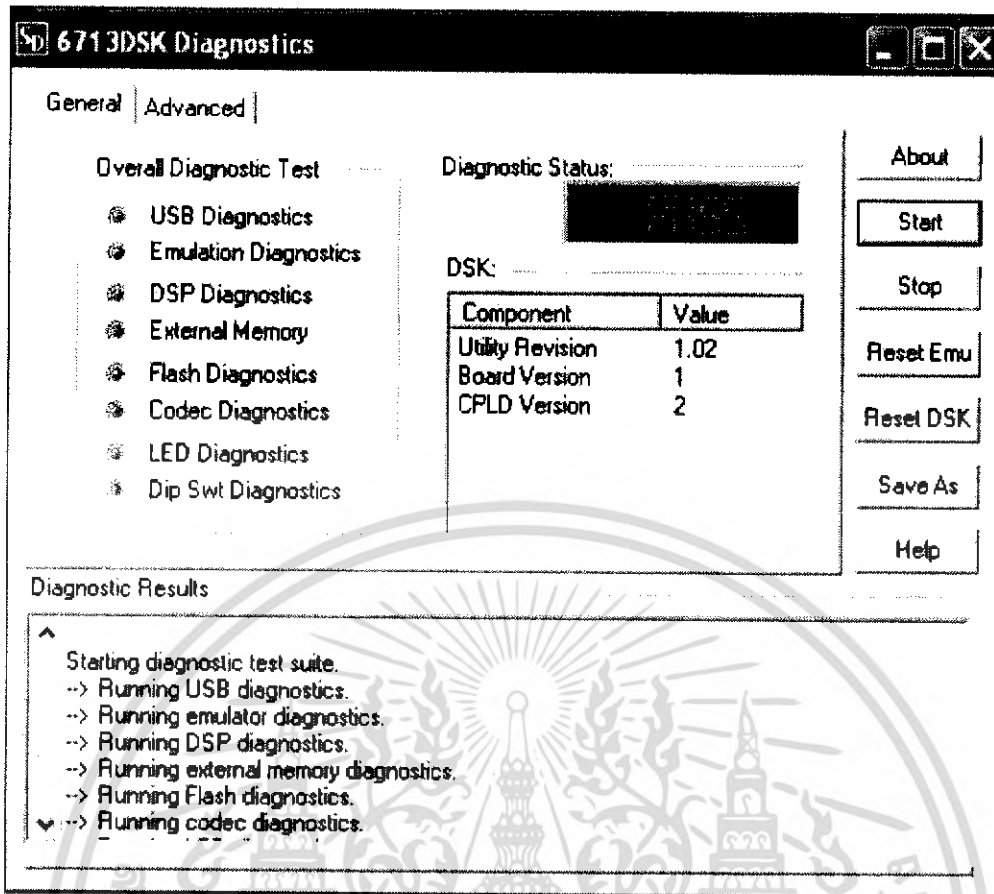
1.6) การทดลองการใช้บอร์ด TMS320C6713



รูปที่ ข.7 การเข้าสู่โปรแกรม Code Composer Studio V2.2 (CCS)

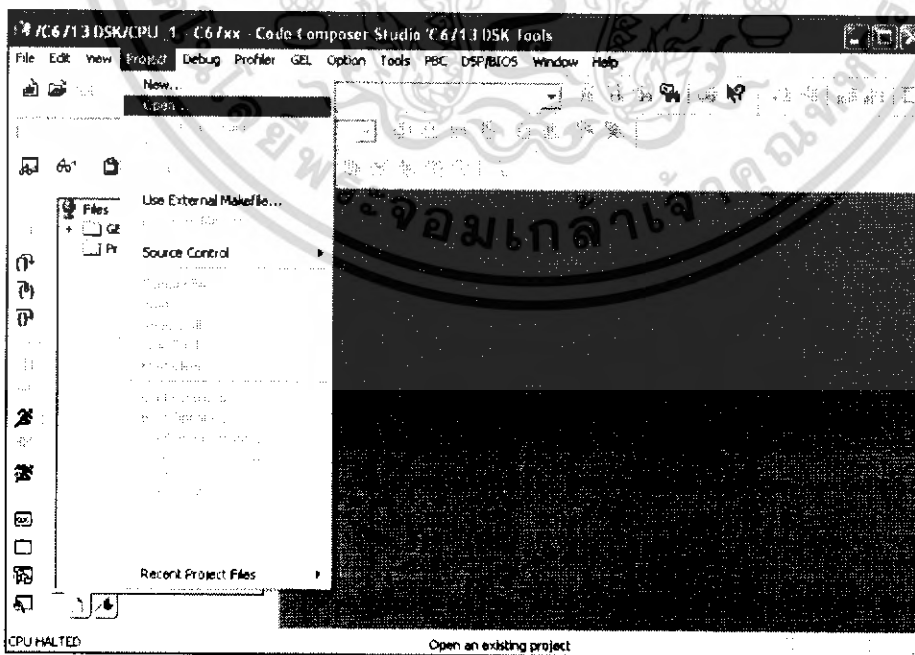
โปรแกรม CCS จะเป็นโปรแกรมใช้ในการควบคุมการทำงานของบอร์ด TMS320c6713 ซึ่งใช้ในการ compile , simulate โปรแกรมที่เสร็จสมบูรณ์แล้ว ในการใช้โปรแกรมนั้นจำเป็นต้องทำการเชื่อมต่อบอร์ดเข้ากับ computer เสียก่อนจึงจะสามารถเปิดใช้งานโปรแกรมได้ โดยในการเชื่อมต่อจะทำการเชื่อมต่อผ่านพอร์ต USB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.8 โปรแกรม 6713DSK Diagnostics

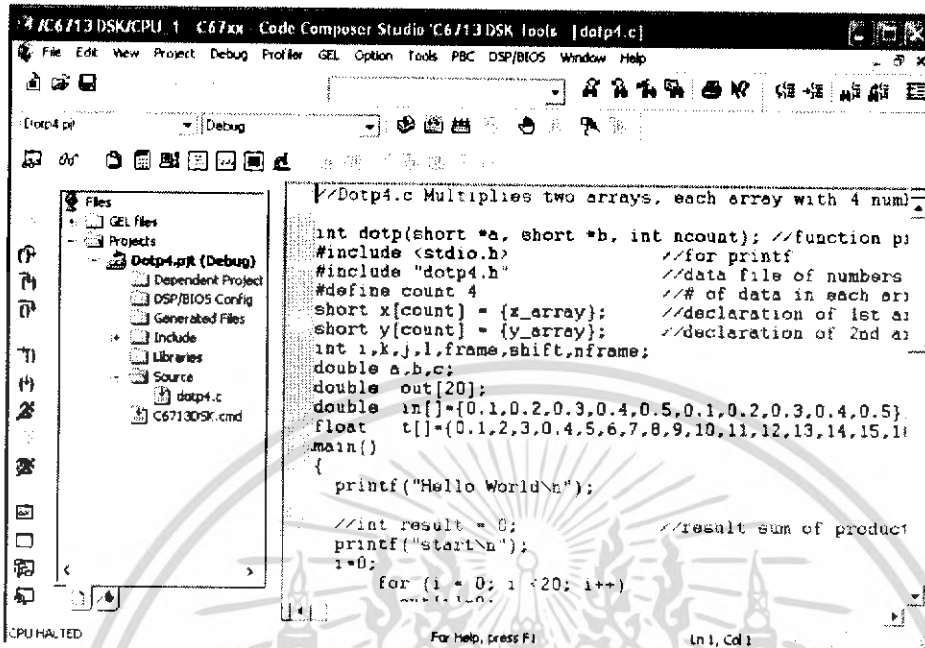
โปรแกรม 6713DSK Diagnostics เป็นโปรแกรมที่ใช้ทำการ check สถานะต่างๆของบอร์ดว่าพร้อมใช้งานหรือไม่



รูปที่ ข.9 แสดงการเปิด Project ขึ้นมาใช้งาน

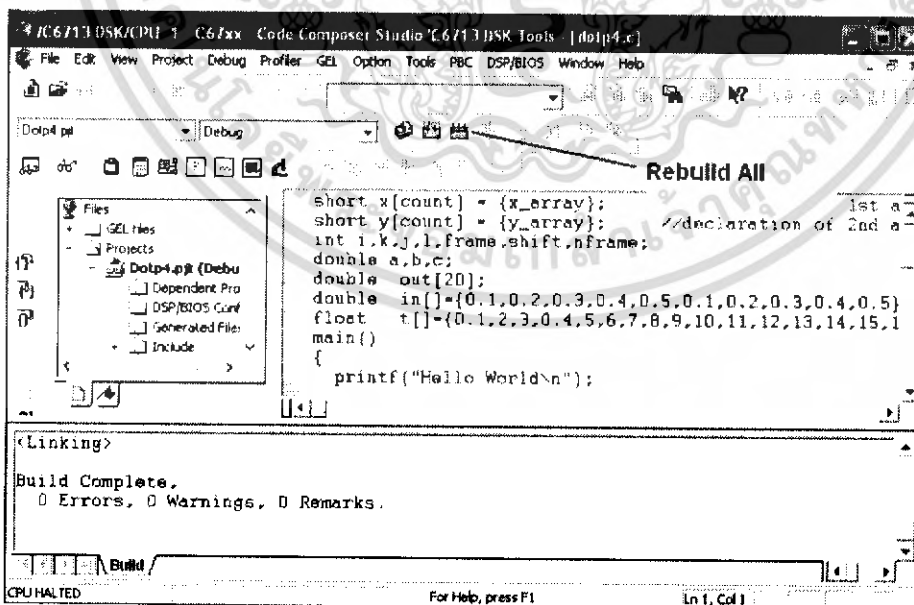
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการการเปิด Project โดย คลิกที่ Project → Open → file.pjt



รูปที่ ข.10 แสดงตัวอย่างโปรเจก dotp4

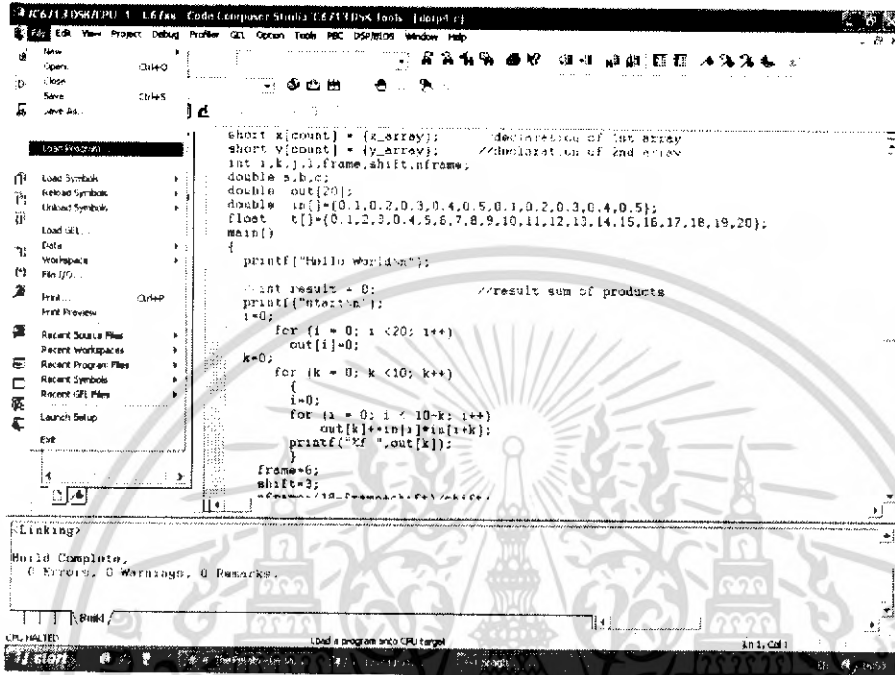
ภายในโปรเจก ก็จะประกอบไปด้วย ไฟล์ include ต่างๆ ที่ใช้ในโปรเจก ซึ่งก็จะเป็นไฟล์ Header ที่ใช้ในการควบคุมโปรแกรม ส่วนโปรแกรมหลักนั้นก็จะเป็น โปรแกรมที่เขียนขึ้นจาก โครงสร้างของภาษา C



รูปที่ ข.11 แสดงการ compile โปรแกรม

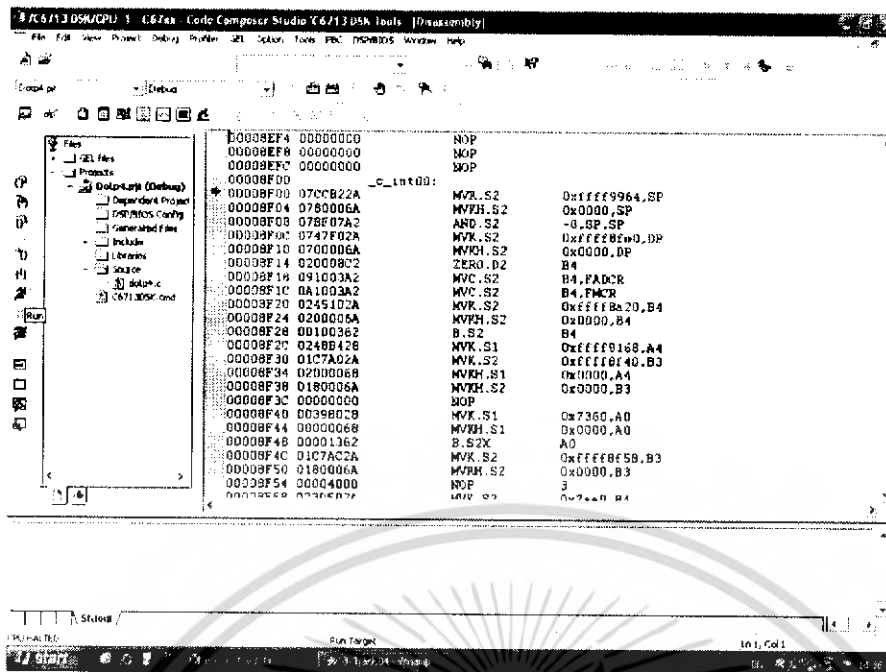
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำการ compile โปรแกรม โดยคลิกที่ปุ่ม Rebuild All เมื่อโปรแกรมทำการ compile ผ่านแล้ว ก็
จะแสดง ข้อความขึ้นมาว่า Build Complete และถ้าไม่ผ่านจึงจะแสดง Errors ที่เกิดขึ้น เมื่อทำการ compile
เสร็จแล้ว Code Composer จะทำการสร้าง File.out ซึ่งอยู่ใน Folder Debug เพื่อเตรียมไว้สำหรับการ load
program ลงไปที่ บอร์ด



รูปที่ ข.12 แสดงการ load Program ลงไปที่ บอร์ด

ทำการ Load Program เข้าไปในบอร์ด โดย เลือกไปที่ File → Load Program แล้วจึงเลือก
File.out ที่อยู่ใน Folder Debug เมื่อทำการ load Program แล้ว โปรแกรมจะถูกแปลงเป็นภาษา Assembly
ซึ่งเป็นภาษาที่ใช้ในการควบคุม Hardware



รูปที่ ข.13 แสดง การ run Program

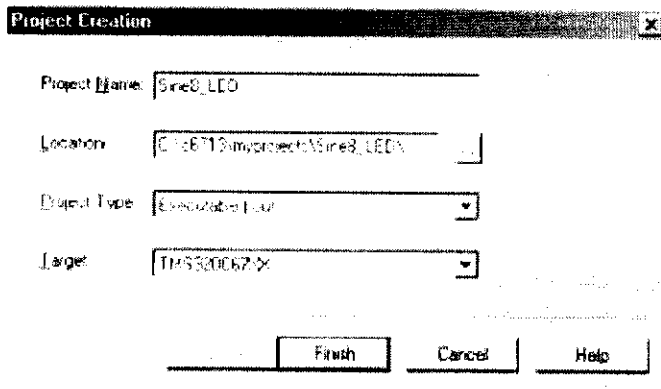
เมื่อทำการ load Program เสร็จเรียบร้อยแล้ว จึงกดปุ่ม run ที่อยู่ทางซ้ายมือ บอร์ด TMS ก็ทำการประมวลผลและแสดงผลมาที่หน้าต่าง Program Code Composer และเมื่อต้องการที่จะหยุดการประมวลผล ให้คลิกที่ปุ่ม Halt ที่อยู่ติดกับ ปุ่ม run บอร์ดจะหยุดการประมวลผลไว้

1.7) การสร้างโปรเจกใหม่

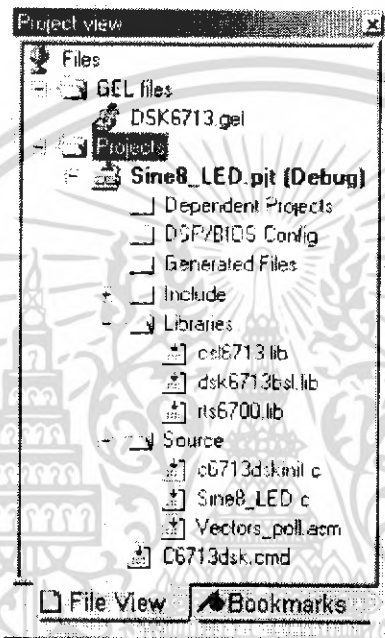
ในการสร้าง โปรเจก ขึ้นมาใหม่นั้น มีขั้นตอน ดังนี้

- 1 เลือก Project → New แล้วพิมพ์ ชื่อ Project ที่ ช่อง Project Name โปรเจกที่สร้างขึ้นจะถูกสร้าง Folder ที่ชื่อเดียวกับ โปรเจก แล้วไปเก็บไว้ใน Folder myprojects
- 2 ทำการ include support file และ File ที่อ้างอิงใน Project เข้าไปใน project เมื่อ open file ขึ้นมาแล้ว ให้เลือกไปที่ Project → Add Files to Project ไฟล์ที่ทำการ include แล้วจะถูกใส่อยู่ใน Folder Project ที่สร้างขึ้น
- 3 เมื่อทำการเขียน ส่วนของ main program (File.c) เสร็จแล้ว ทำการ Add Files เหมือนขั้นตอนที่ 2 File.c ก็จะถูกบรรจุอยู่ใน Folder Source ใน Project

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(a)



(b)

รูปที่ ข.14 รูปหน้าต่าง Project (a) หน้าต่างการสร้างโปรเจก (b) หน้าต่างแสดง file ในโปรเจก

2.) การทดลองการเขียนโปรแกรม ด้วย บอร์ด TMS320c6713

2.1) โปรแกรม Hello World (project ../ti/tutorial/hello1)

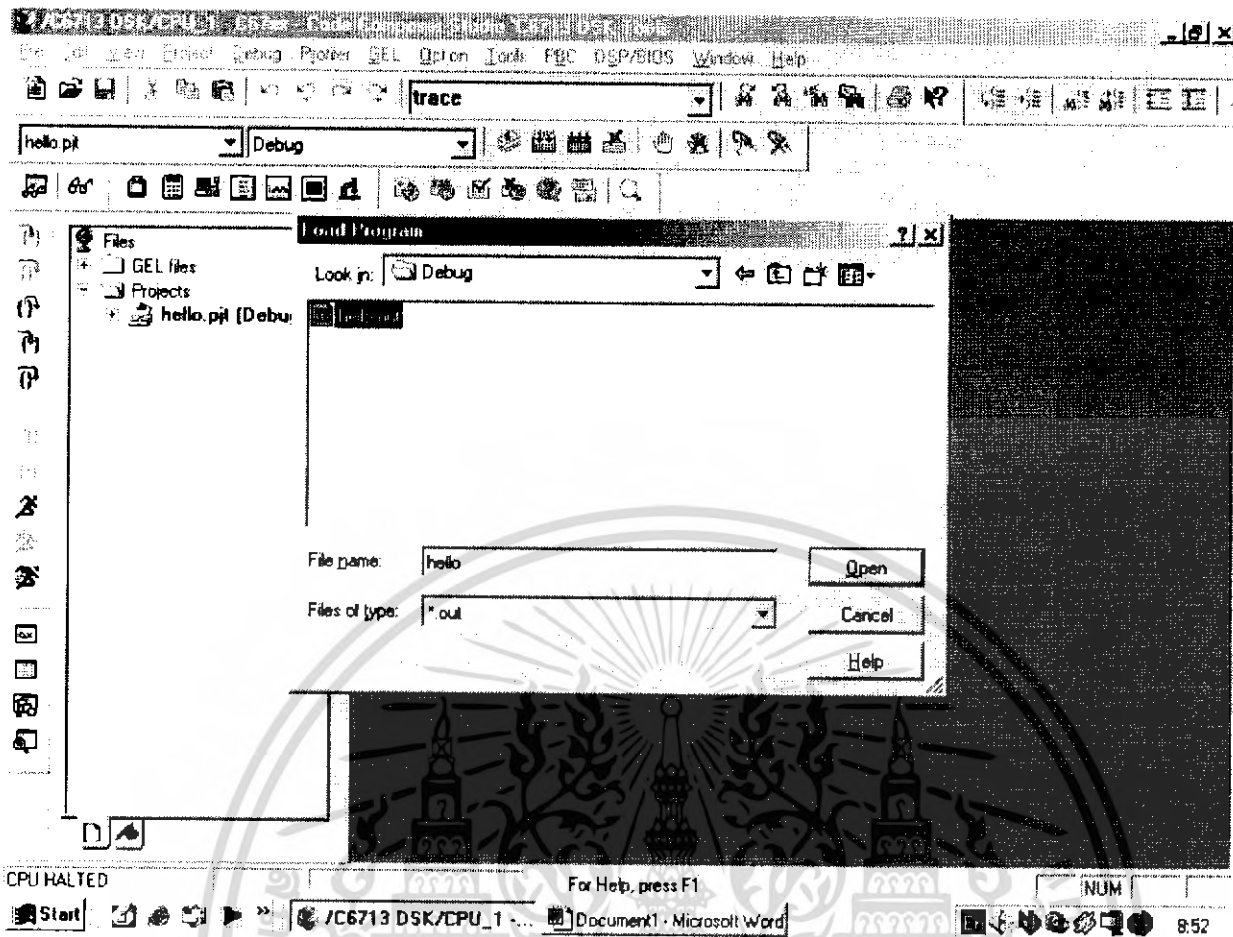
เป็นการเรียกใช้ Library พื้นฐานของภาษา C ธรรมดา ได้แก่ printf, scanf โดยผ่าน lib stdio.h

วิธีการ Compile

เลือก Project -> Rebuild all

ทำการ load program โดยผ่าน FILE -> Load Program -> Debug -> *.out คลิกที่ Open

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.15 แสดง วิธีการ Load Program

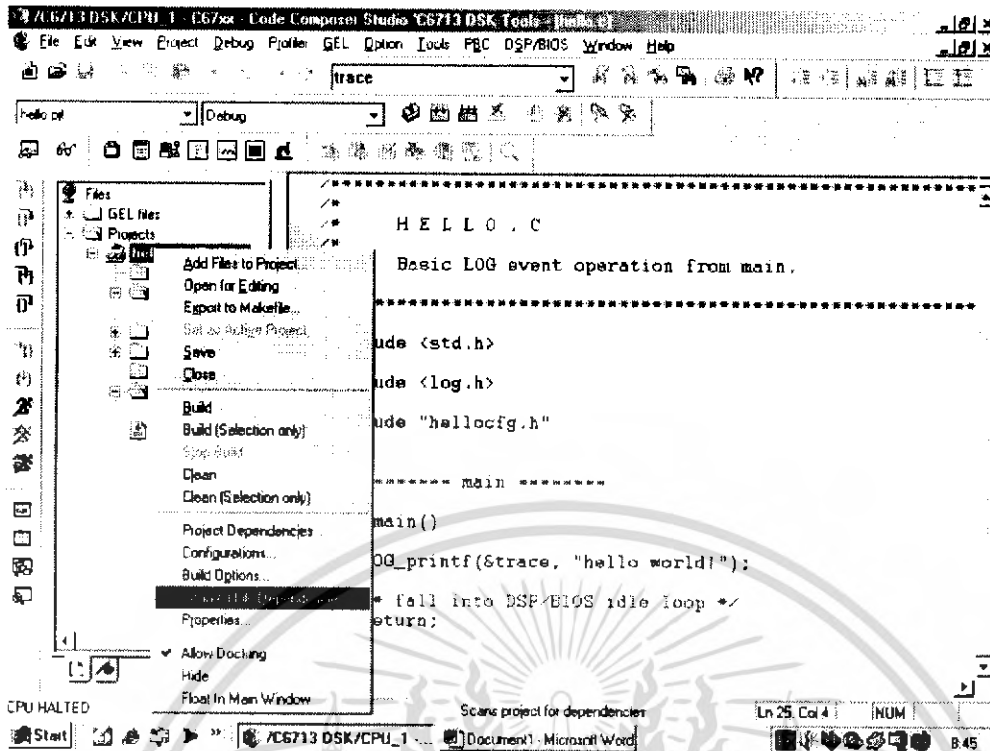
2.2) โปรแกรม Hello World 2 (project ../ti/tutorial/hello2)

เนื่องจากการเรียกใช้ Library พื้นฐานของภาษา C มี Overhead ในการทำงานสูงดังนั้นจึงเปลี่ยนมาใช้ function ของ DSP BIOS สำหรับการเรียกแทน function printf โดยใช้ LOG_Obj

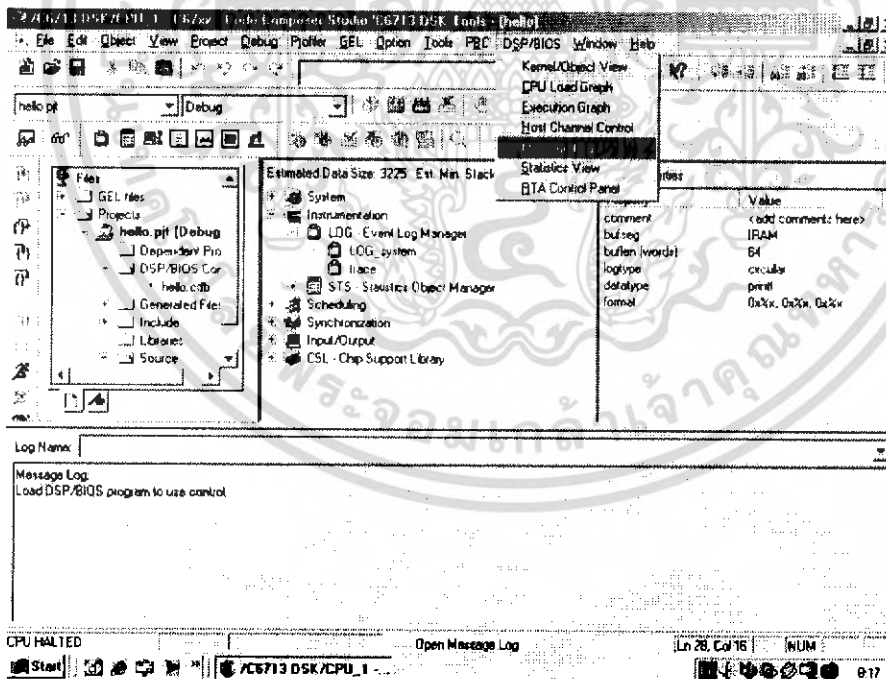
```
LOG_printf(&trace, "hello world!"); // trace เป็นชื่อของ Log Object
```

สำหรับวิธีการสร้างนั้นเริ่มต้นเข้าไปที่ FILE -> NEW -> DSP BIOS Configuration เลือกตระกูลของบอร์ดทดลองที่ใช้ ในที่นี้ได้ใช้ dsk6713.cdb ซึ่งเป็น file สำหรับเก็บ Configuration จากนั้นเลือก Instrumentation -> LOG - Event Log Manager click ขวาเลือก Insert Log ทำการตั้งชื่อของ Object และกำหนด Parameter โดย Click ขวาแล้วเลือก property หลังจากกำหนด parameter เรียบร้อยแล้วจะต้องทำการ scan dependency เพื่อนำ file อื่น ๆ ที่เกี่ยวข้อง Include เข้ามาใน project โดย Click ขวาที่ ชื่อ Projects แล้วเลือก Scan All File Dependency

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.16 แสดง วิธีการ Scan dependency
Tool ที่ใช้ในการ view message อยู่ที่ DSP/BIOS -> Message Log ดังรูป



รูปที่ ข.17 แสดง วิธีการ ดู Log message

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3) โปรแกรม Loop_store

โปรแกรมนีเป็นการนำสัญญาณจาก ช่อง Microphone และ Line in แล้วนำมาเก็บไว้ใน array buffer โดยวิธีการทำ Loop Interrupt โดย input sample ที่ read เข้ามาจาก codec's ADC จะถูก write ออกไปที่ output sample ด้วย และ ข้อมูล 512 sample ล่าสุดจะถูกเก็บไว้ในตัวแปร buffer

โดยมี code โปรแกรมดังนี้

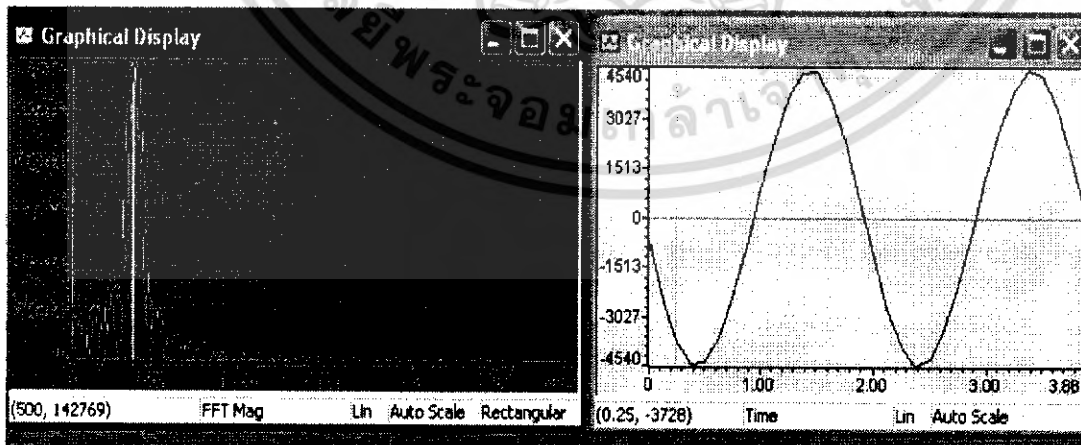
```
//Loop_store.c Data acquisition. Input data also stored in buffer
#include "DSK6713_AIC23.h" //codec-DSK interface support
uint32 fs=DSK6713_AIC23_FREQ_8KHZ; //set sampling rate
#define BUFFER_SIZE 512 //buffer size
short buffer[BUFFER_SIZE]; //buffer where data is stored
int i = 0;

interrupt void c_int11() //interrupt service routine
{
    output_sample((short)input_sample()); //input/output data

    buffer[i] =((short)input_sample()); //store data in buffer
    i++; //increment buffer index
    if (i==BUFFER_SIZE) i = 0; //reinit index if buffer full
    return; //return from ISR
}

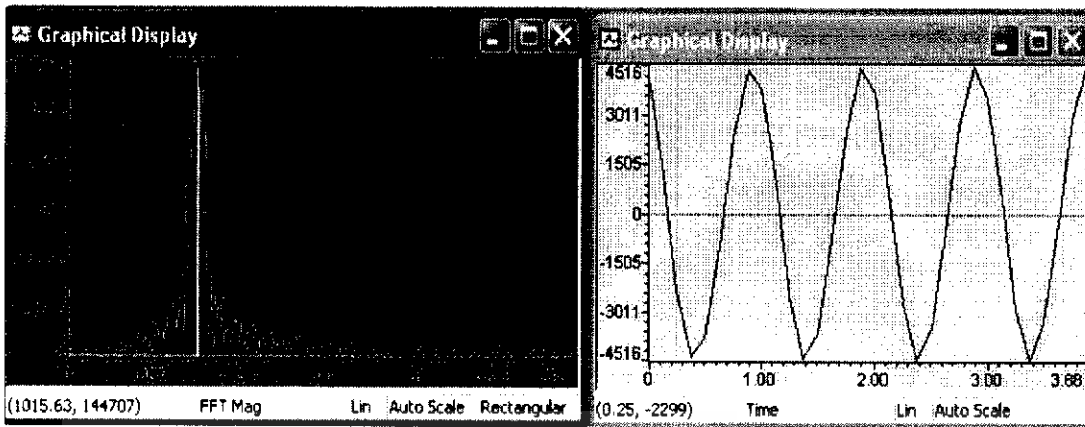
void main()
{
    comm_intr(); //init DSK, codec, McBSP
    while(1); //infinite loop
}
```

ทำการทดลอง โดยนำสัญญาณ จาก Sine จาก Generator เข้าที่ช่อง Line In แล้ว นำข้อมูลจาก buffer array มาพล็อตกราฟในแกนเวลา และ ทำการแปลง FFT พล็อตกราฟในแกนความถี่



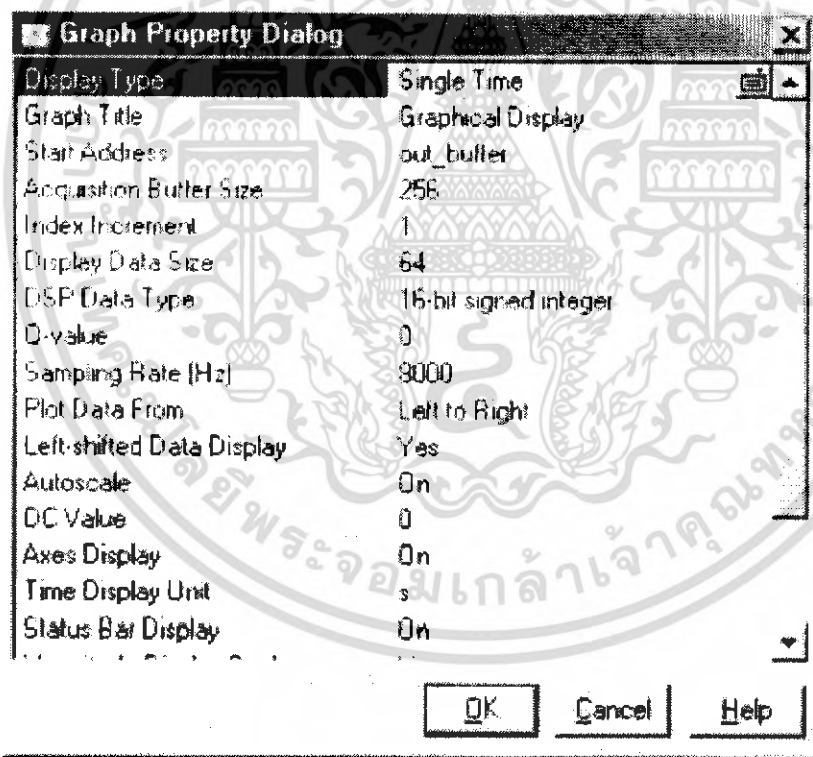
รูปที่ ข.18 แสดงสัญญาณ Sine ความถี่ 500 Hz ในแกนความถี่และเวลาที่ได้จาก input sample ใน buffer array

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ข.19 แสดงสัญญาณ Sine ความถี่ 1000 Hz ในแกนความถี่และเวลาที่ได้ออก input sample ใน buffer array

ในการพล็อตกราฟ ในแกนเวลาและความถี่จะต้องเลือก tool โดยเลือกไปที่ View → Graph → Time/frequency จะปรากฏ dialog ขึ้นดังรูป

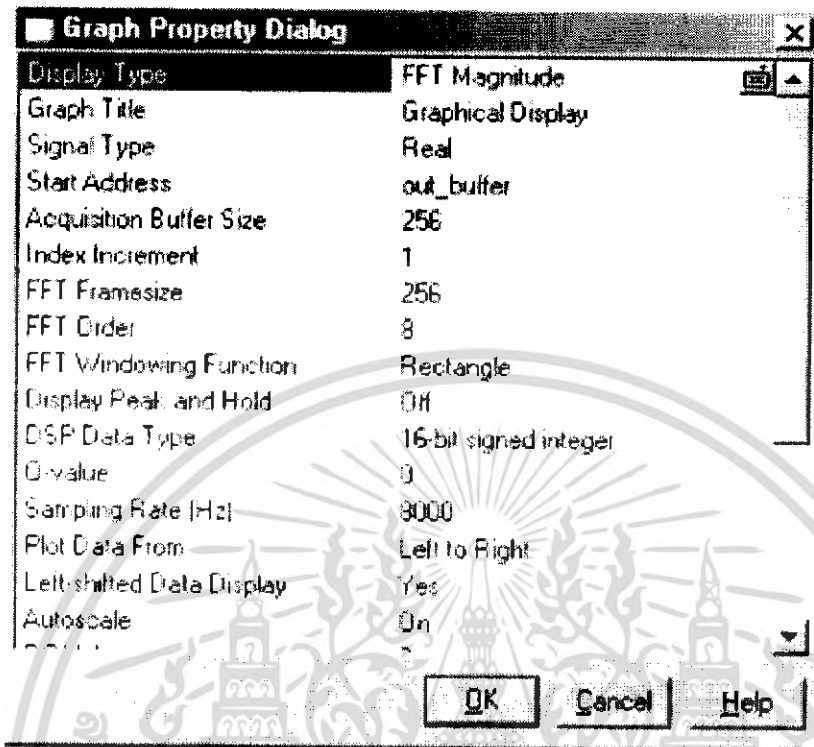


รูปที่ ข.20 การตั้งค่า dialog ในการพล็อตกราฟ ในแกนเวลา

สำหรับในแกนเวลาจะ set display type เป็น Single Time และเลือก Start Address เป็นตัวแปรที่เราต้องการนำมา display ในที่นี้ก็คือ ตัวแปร “buffer” และเลือก Acquisition buffer เป็น 512 points ,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

data display size เป็น 64 points , DSP data type เป็น 64 และ เลือก sampling rate เป็น 8 kHz ตามที่ได้ตั้งไว้ที่ Header File



รูปที่ ข.21 แสดงการตั้งค่า dialog ในการพล็อตกราฟ ในแกนความถี่

ทำการ set Display Type เป็น FFT Magnitude เพื่อที่จะทำการพล็อตกราฟ ในแกนความถี่ Buffer size และ FFT Frame size เป็น 256 และ FFT order ของ $M = 8$ เพราะว่า $2^M = 256$

2.4) โปรแกรม Fir_Filter

```
//Fir.c FIR filter. Include coefficient file with length N
#include "bs2700_cof" //coefficient file
#include "dsk6713_aic23.h" //codec-dsk support file
uint32 fs=DSK6713_AIC23_FREQ_8KHZ; //set sampling rate
int yn = 0; //initialize filter's output
short dly[N]; //delay samples

interrupt void c_int11() //ISR
{
    short i;

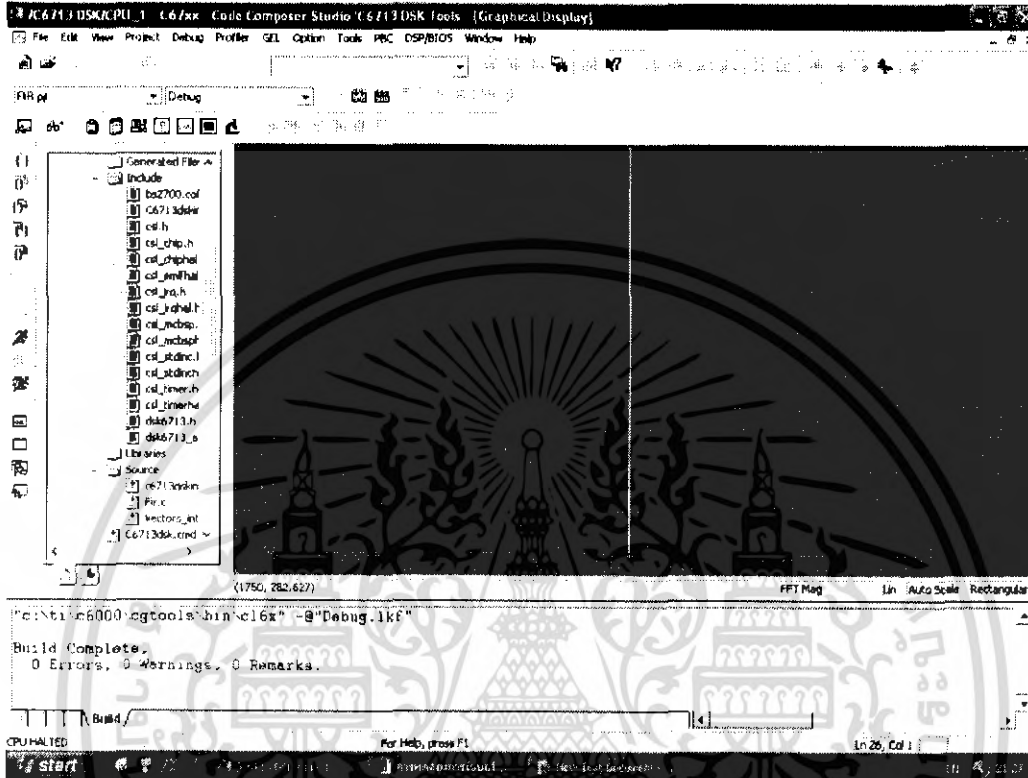
    dly[0]=input_sample(); //input newest sample
    yn = 0; //initialize filter's output
    for (i = 0; i < N; i++)
        yn += (h[i] * dly[i]); //y(n) += h(i)* x(n-i)
    for (i = N-1; i > 0; i--) //starting @ end of buffer
        dly[i] = dly[i-1]; //update delays with data move
    output_sample(yn >> 15); //scale output filter sample
    return;
}

void main()
{
    com1_intr(); //init DSK, codec, McBSP
    while(1); //infinite loop
}
```

รูปที่ ข.22 แสดง Code โปรแกรม ในการสร้าง FIR Filter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโปรแกรม จะมี include file bs2700.cof ซึ่งเป็น file ที่ใส่ สัมประสิทธิ์ ของ filter ที่ต้องการนำมาสร้าง โดยทำการออกแบบด้วย ฟังก์ชัน Sptool ในโปรแกรม Matlab ซึ่งการทดลองนี้จะทำการออกแบบ High Pass Filter order 20 ทำการ load program และ run แล้วจึงนำ h[n] มาพล็อตหา impulse response แล้วทำการแปลงฟูเรียร์ จะได้ Frequency Response ของระบบ



รูปที่ ข.23 frequency response ของ high pass filter ที่มี ความถี่ช่วงความถี่สูง

ในการหา Feature Attraction เมื่อนำสัญญาณเสียงเข้ามาแล้วเราจะนำเสียงที่เก็บมาได้ มาทำการปรับแอมพลิจูด

2.5) โปรแกรม Segmentation

```
//Program Segmentation
#include <stdio.h> //for printf
int i,j,k,l,nframe,shift,frame;
int out[20];
int in[]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20}; //Generate input sequence = 20
main()
{
    frame=10; //assign sample/frame
    shift=5; //assign shift
    nframe=(20-frame+shift)/shift; //calculate number of frame
    j=0;
    for (j=0; j<=nframe ;j++)
    {
        i=0;
        for (i=0+j*shift ;i<=(frame-1)+j*shift ;i++) //display output
            printf("%d",in[i]);
    }
}
```

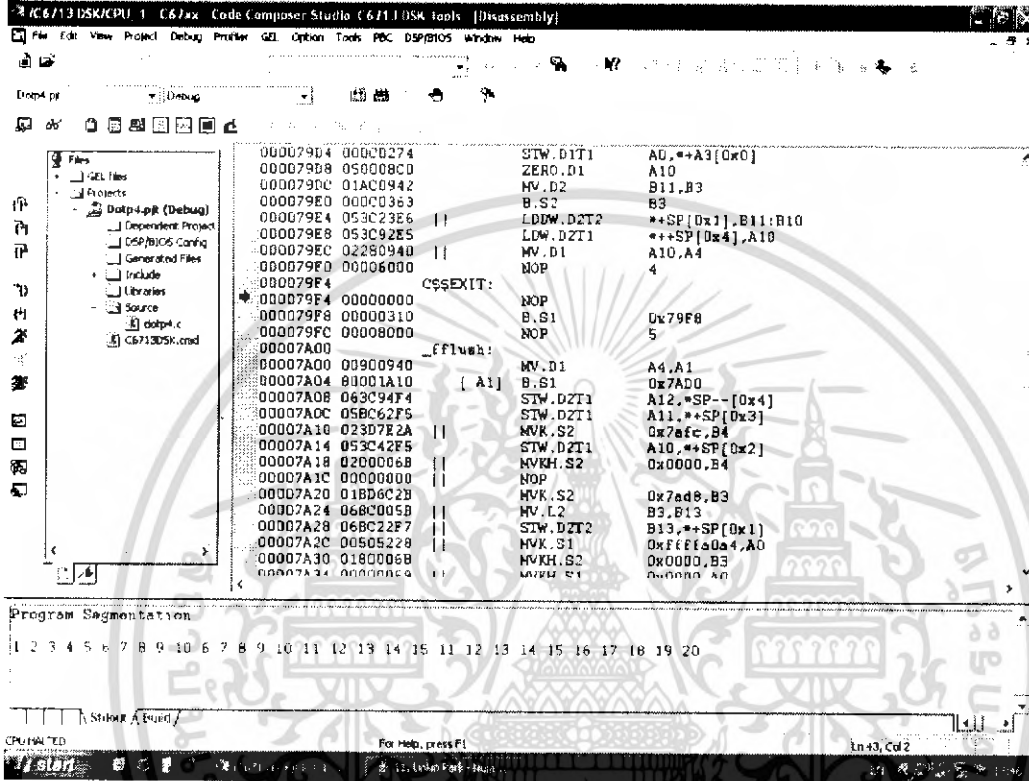
รูปที่ ข.24 แสดง Code โปรแกรม Segmentation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

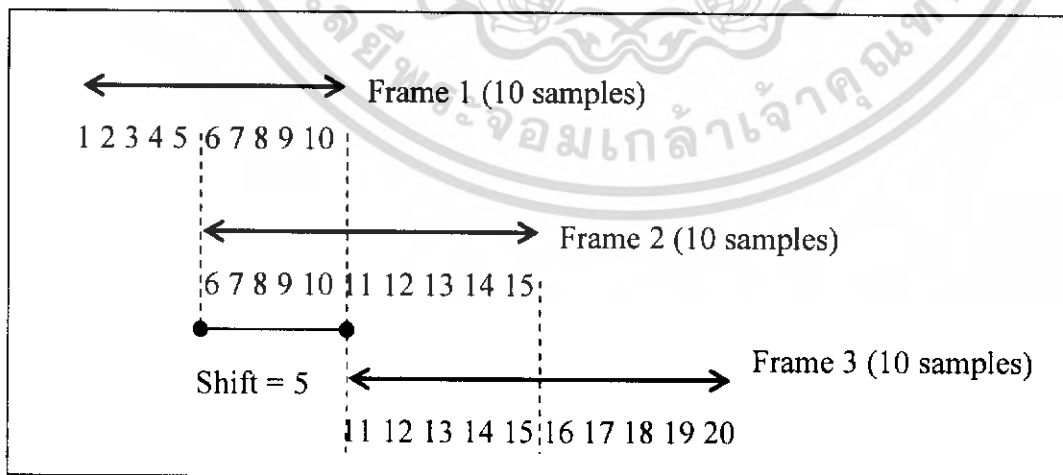
ในการประมวลผลสัญญาณเสียงนั้นจะมี samples ของข้อมูลยาวมาก จึงจำเป็นต้องมีการแบ่งเฟรม ข้อมูลออกเป็นข้อยๆ โดยเราทำการทดลอง จะทำการจำลองสร้าง sequence ขึ้นมา แทนข้อมูลเสียงเป็น จำนวน $l = 20$, $frame = 10$ และ $shift = 5$ และมีสูตรในการคำนวณหาจำนวน frame ข้อมูลคือ

$$n_{frame} = \frac{(l - frame + shift)}{shift} \quad (ข.1)$$

ซึ่งจะมีผลการทดลองดังนี้



(a)



(b)

รูปที่ ข.25 (a) และ (b) แสดงผลการทดลองโปรแกรม Segmentation จาก บอร์ด TMS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6) โปรแกรม Autocorrelation

samples ที่ได้จากการตัด frame ข้อมูลแล้ว จะนำมาทำการหาค่า Autocorrelation ซึ่งกำหนด order ไว้ 10 จะได้ ค่า $r[i]$ ทั้งหมด 10 ค่า โดยในการทดลองจะสร้าง Code โปรแกรม ที่สมมุติ input เป็น matrix ขนาด 1X20 แทนเป็น input จำนวน 20 samples ซึ่งจะสร้างจากความสัมพันธ์

$$R_n = \sum_{n=0}^{n-k} s(n)s(n+k) \quad (ข.2)$$

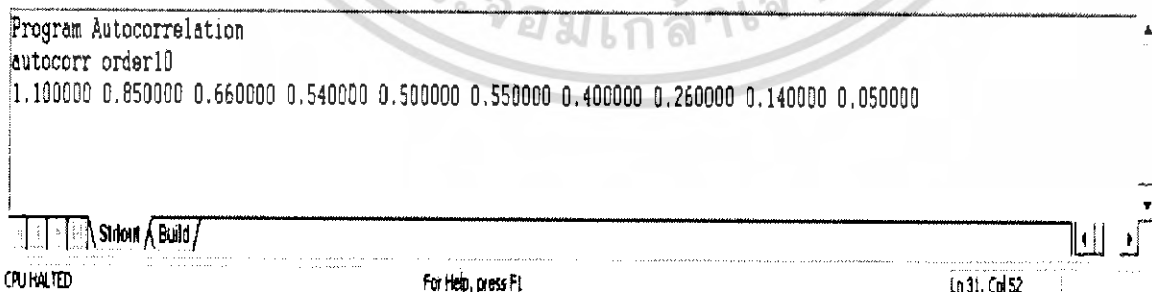
ซึ่งจะได้ Code โปรแกรม ดังนี้

```
//Prpgram Autorcorrelation

#include <stdio.h> //for printf
int i,k,j,l,frame,shift,nframe;
double a,b,c;
double out[20];
double in[]={0.1,0.2,0.3,0.4,0.5,0.1,0.2,0.3,0.4,0.5};
main()
{
    printf("Program Autocorrelation\n");

    printf("autocorr order10\n");
    i=0;
    for (i = 0; i < 20; i++) //clear all out array
        out[i]=0;
    k=0;
    for (k = 0; k < 10; k++) //fix orders autocorr=10
    {
        i=0;
        for (i = 0; i < 10-k; i++)
            out[k]+=in[i]*in[i+k];
        printf("%f ",out[k]);
    }
}
```

รูปที่ ข.26 แสดง Code โปรแกรม Autocorrelation



```
Program Autocorrelation
autocorr order10
1.100000 0.850000 0.660000 0.540000 0.500000 0.550000 0.400000 0.260000 0.140000 0.050000
```

รูปที่ ข.27 แสดงผลการหาค่า Autocorrelation จากบอร์ด TMS C6713

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Current Directory: C:\MATLAB65\work

>> z=[0.1 0.2 0.3 0.4 0.5 0.1 0.2 0.3 0.4 0.5];
>> y=xcorr(z);
>> y

y =

Columns 1 through 10
    0.0500    0.1400    0.2600    0.4000    0.5500    0.5000    0.5400    0.6600    0.8500    1.1000

Columns 11 through 19
    0.8500    0.6600    0.5400    0.5000    0.5500    0.4000    0.2600    0.1400    0.0500

```

รูปที่ ข.28 การหาค่า Autocorrelation จาก โปรแกรม Matlab

ทำการตรวจสอบความถูกต้องของการหาค่า Autocorrelation จากบอร์ดโดยการหาค่าจากโปรแกรม Matlab ซึ่งจะใช้ คำสั่ง xcorr ซึ่งเป็นการหา Autocorrelation แบบ Two sides แต่เราต้องการใช้ค่าเพียงด้านเดียวดังนั้นจะใช้ค่าตั้งแต่ Column ที่ 11-19 ซึ่งจะเห็นว่าค่าตรงกันกับการหาค่า Autocorrelation จากบอร์ด TMS320c6713

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Source Code โปรแกรม

1. โปรแกรมทดสอบคำนวณหาค่าพารามิเตอร์ต่างๆ ของคุณลักษณะของเสียงจาก MATLAB

```

clear;
%load file.wav
[input,fs]=wavread('a_0.wav');
%set parameter
l=length(input);
lpc=[];
cep=[];
p=10;
frame=80;
shift=40;
q=15;
%preemphasis
inlpc=filter([1,-0.9375],[1],input);
for i=1:(length(input)-frame+shift)/shift;
    format long;
    %frame blocking
    for m=1:frame;
        z(m)=inlpc(m+(shift*(i-1)));
    end
    %windowing
    for m=1:frame;
        x(m)=z(m)*(0.54+0.46*cos(2*pi*(m-1)/(frame-1)));
    end
    %autocorrelation
    r=xcorr(x);
    for m=1:p,
        for j=m:p,
            s(m,j)=r(frame+(j-m));
            s(j,m)=s(m,j);
        end
    end
    for m=1:p,
        phi(m)=r(frame+m);
    end
    a=inv(s)*phi';
    b=[1,-a'];
%LPC coefficient
    for m=1:p+1,
        if m==1
            a(m)=1;
        else
            a(m)=b(m);
        end
    end
    lpc(:,i)=a;
end
fid = fopen('C:\MATLAB7\bin\lpc\a_0.m','wb');
count=fwrite(fid,lpc,'float64');
status=fclose(fid);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมเตรียมสัญญาณเสียง

```
%Read sound and store sample data to file
%generate hamming window and store
Fs=8000;
[inputs,Fs] = wavread('s.wav'); %load ไฟล์เสียงเข้าที่นี้
sound(inputs,Fs);
plot(inputs);grid;
div=round(Fs/8000); %in same directory as .m file
inputs=resample(inputs,1,div); %sample rate =8k
start=1;
frame_length=80;
given=length(inputs);
input=filter([1,-0.9375],[1],inputs); %preemphasis
input(given+1:4880)=0;
zeros_to_add=4880-given;
fid=fopen('C:\ti\myprojects\Dotp4\sound.h','w');
disp('The value of Def_total in Speech.c is');
disp(given-start+1+zeros_to_add);
fprintf(fid,'float input[%d]={',given-start+1+zeros_to_add);
for index=start:given+zeros_to_add
    if index==given+zeros_to_add
        fprintf(fid,'%i\n',input(index));
    else
        fprintf(fid,'%i,\n',input(index));
    end
end
fprintf(fid,'};\n');
fclose(fid);

ham=hamming(frame_length); %Hamming Window
fid=fopen('C:\ti\myprojects\Dotp4\ham.h','w');
fprintf(fid,'int ham[%d]={',frame_length); %coefficients->hamming.h
for index=1:frame_length
    if index==frame_length
        fprintf(fid,'%i\n',ham(index));
    else
        fprintf(fid,'%i,\n',ham(index));
    end
end
fprintf(fid,'};\n');
fclose(fid);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมหาคุณลักษณะเด่นของเสียงด้วย บอร์ด TMS 320C6713

```
//Program Feature Attraction//
int dotp(short *a, short *b, int ncount); //function prototype
#include "dsk6713_aic23.h"
Uint32 fs=DSK6713_AIC23_FREQ_8KHZ;
#include <stdio.h>
#include <math.h>
#include "sound.h" /* Load sound sample*/
#include "hammingr.h" /* Load Hamming Coef.*/
#define Def_overlap 40 /* Set overlap length */
#define Def_framelength 80 /* Set frame length */
#define Def_total 4840 /*Set length of the speech samples */

/**** SEGMENTATION FUNCTION ****/
/**** input : start:Index of input array defining the start of the
segment *****/
/**** return: segment:pointer to the first element of the segment
*****/
/**** This function takes in the starting index and returns a
*****/
/**** segment whose length is Def_framelength
*****/
float/*f*/ * segmentation(int start)
{
int loop;
int seg_loop=0;
float/*f*/ * segment;
segment=(float *)malloc(sizeof(float)*Def_framelength);//f
for(loop=start;loop<start+Def_framelength;loop++ )
{
segment[seg_loop]=input[loop];
seg_loop++;
}
return segment;
}
//WINDOW FUNCTION//
float* window(float* sp_input)//f
{
int loop=0;
float/*f*/ * out;

out=(float *)malloc(sizeof(float)*Def_framelength);//f
for(loop=0;loop<Def_framelength;loop++)
out[loop]=sp_input[loop]*ham[loop]; /* เอา sample input มาคูณ กับ ตาราง Hamm*/
return (out);
}
///// Function AutoCorrelation/////

float* xcorr(float* sp_input, short p)//f
{
short i;
int j;
float/*f*/ * corr;

corr=(float*)malloc((sizeof(float)* p));//f
for(i=0;i<p;i++)
{
corr[i]=0;
// Loop till Def_framelength-i so that index does not spill over

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ญาติเห็นไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for (j=0;j<Def_framelength-i;j++)
            corr[i]=corr[i]+ (sp_input[j]*sp_input[j+i] );
    }
    return (corr);
}

```

```

////////// function Levinson Durbin //////////
float* levinson(float/*f*//* r, short order)
{
    float *e;
    float *k;
    float *alpha;
    float *alpha_prev;
    int i=1,j=1 ;
    float sum=0.0;\
    short loop;
    e=(float *)malloc((order+1)*sizeof(float));
    k=(float *)malloc((order+1)*sizeof(float));
    alpha=(float *)malloc((order+1)*sizeof(float));
    alpha_prev=(float *)malloc((order+1)*sizeof(float));
    // Initialising the Array variables
    for (loop=0;loop<order+1;loop++)
    {
        e[loop]=0;
        k[loop]=0;
        alpha[loop]=0;
        alpha_prev[loop]=0;
    }
    // End initialising
    e[0]=r[0];
    // Iterate with i=1 to order as iteration variable
    i=1;
    for(i=1;i<=order;i++)
    {
        // start step 1
        sum=0;
        for (j=1;j<=i-1;j++)
            sum+=alpha_prev[j]*r[i-j];
        k[i]=(r[i]-sum)/e[i-1];
        // End step 1
        //start step 2
        alpha[i]=k[i];
        for(j=1;j<=i-1;j++)
            alpha[j]=alpha_prev[j]- ( k[i]*alpha_prev[i-j] ) ;
        //end step 2
        //start step 3
        e[i]=e[i-1]*(1-(k[i]*k[i]));
        //end step 3
    }
    for( loop=0;loop<order+1;loop++)
    {
        alpha_prev[loop]=alpha[loop];
    }
} // End For levinson
for( loop=0;loop<order+1;loop++)
alpha[loop]=-1*alpha[loop];
alpha[0]=1;
free(alpha_prev);
free(e);
free(k);
return (alpha);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

////////// main Program //////////
void main()
{
FILE *fp1; /* เปิดfile สำหรับLPC*/
FILE *fp2; /* เปิดfile สำหรับAutocorr*/
int loopvar=0;
float *present_seg;//f
float *corr;//f
float* lpc;
float *windowed;//f
int i,j,n_frame;
fp1= fopen("C:/Program Files/MATLAB71/work/pro/lps.h","w+b");
fp2= fopen("C:/Program Files/MATLAB71/work/pro/autocor.h","w+b");
n_frame=(Def_total-Def_framelength+Def_overlap)/Def_overlap;
printf("total frame=%d\n",n_frame);

/*Start of main segmentation*/
for (loopvar=0;loopvar<=Def_total-
Def_framelength;loopvar+=Def_overlap)
{
present_seg=segmentation(loopvar); //segmentation
windowed=window(present_seg); //multiply by contents of hamming
free(present_seg);
corr=xcorr(windowed,10);//call function autocorr
lpc=levinson(corr,9); //call function LPC coefficients
free(corr);
free(windowed);
free(lpc);
/* display LPC ในworkspace */
for (j = 0 ; j<10 ; j++)
printf(" %f",lpc[j]);
printf("\n");
/*Write LPC to file*/
if(fp1 == NULL)
{
printf("Error opening file for read");
exit(0);
}
else
{
for (i=0; i< 10 ; i++)
fprintf(fp1,"%f,",lpc[i]);
fprintf(fp1,"\n");
}
/*Write Autocorr to file*/
if(fp2 == NULL)
{
printf("Error opening file for read");
exit(0);
}
else
{
for (j=0; j< 10 ; j++)
fprintf(fp2,"%f,",corr[j]);
fprintf(fp2,"\n");
}
}

printf("finish");
fclose(fp1);
fclose(fp2);}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. โปรแกรมหาค่าสัมประสิทธิ์ Cepstrum & Weight

```
clc
c=[];
r=[];
a=[];
p=9;
q=15;
load lps.h;
a=lps';
format long;
for j=1:120
for i=11:16
    a(i,j)=0;
end
end
load autocor.h
r=autocor';
% for j=1:120
% for i=1:10
%     r(i,j)=r(i,j)/(2^15);
% end
% end

for j=1:120
%Gain
    eo(j)=r(1,j);
    for m=2:p+1
        er(m,j)=a(m,j)*r(m,j);
        e1(j)=sum(er(:,j));
    end
    e(j)=eo(j)-e1(j);
    g(j)=sqrt(e(j));
    if g(j)==0
        g(j)=0.000001;
    end
%cepstrum
    for m=0:1,
        if m==0
            v(m+1,j)=log(g(j));
        else
            v(m+1,j)=a(m,j);
        end
    end
    for m=2:q
        if m<=p
            v(m+1,j)=a(m,j);
        else
            v(m+1,j)=0;
        end
        d(m,j)=0;
        c(m-1,j)=0;
        for k=1:m-1
            c(m-1,j)=(k/m)*v(k+1,j)*a(m-k,j);
            d(m,j)=c(m-1,j)+d(m,j);
        end
        v(m+1,j)=v(m+1,j)+d(m,j);
    end
end
%
    cep=v;
%weight parameter
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        for m=1:q+1
            w(m,j)=1+((q/2)*sin(pi*((m/q))));
            w1(m,j)=cep(m,j)*w(m,j);
        end
    end
end

```

5. โปรแกรมสร้าง Code Book

- ฟังก์ชันการสร้าง Code Book

%โปรแกรมสร้าง Code Book

```

function
[codebook,distortion]=VQ_AJ_func(training_seq,dimension,codebook_size
,tolerance)
if (nargin==3)
    tolerance=0.01;
end
m=round(length(training_seq)/dimension);
if (m*dimension-length(training_seq)<0)
    error('length of training_seq is not a multiple of dimension')
end
% initialize the codebook
initial=training_seq(1:dimension*codebook_size);
initialcodebook=(reshape(initial,dimension,codebook_size));
update_codebook=initialcodebook;
% first update
newdistortion=0;
distortion=0;
for k=1:length(training_seq(1,:))
    distortion_block=[];
    distort=[];
    for j=1:codebook_size
        distort=sum((training_seq(:,k)-update_codebook(:,j)).^2);
        distortion_block=[distortion_block distort];
    end
    [distortion_min,ind]=min(distortion_block);
    newdistortion=newdistortion+distortion_min;
    index(k)=ind;
end
% cluster
for l=1:codebook_size
    partition=(index==l);
    if sum(partition)>0
        update_codebook1(:,l)=(training_seq/sum(partition))*(partition);
    end
end
total=0;
% calculate-error
for m=1:codebook_size
    ww=0;
    for j=1:length(training_seq(:,1))
        ww=abs(update_codebook1(j,m)-update_codebook(j,m));
        total=total+ww;
    end
end
if
total/(length(update_codebook(1,:))*length(update_codebook(:,1)))>tol
erance
    tu=1;
end
disp('distortion1=')

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

disp(total/(length(update_codebook(1,:))*length(update_codebook(:,1))
))
% further update until the desired tolerance is met
while tu==1
    index=0;
    for k=1:length(training_seq(1,:))
        distortion_block=[];
        distort=[];
        for j=1:codebook_size
            distort=sum((training_seq(:,k)-
update_codebook1(:,j)).^2);
            distortion_block=[distortion_block distort];
        end
        [distortion_min,ind]=min(distortion_block);
        newdistortion=newdistortion+distortion_min;
        index(k)=ind;
    end
% cluster
    for l=1:codebook_size
        partition=(index==l);
        if sum(partition)>0
            update_codebook2(:,l)=(training_seq/sum(partition))*(partition');
        end
    end
    total=0;
% calculate-error
    for m=1:codebook_size
        ww=0;
        for j=1:length(training_seq(:,1))
            ww=abs(update_codebook2(j,m)-update_codebook1(j,m));
            total=total+ww;
        end
    end
    disp('distortion2=')
disp(total/(length(update_codebook(1,:))*length(update_codebook(:,1))
))
    if
total/(length(update_codebook(1,:))*length(update_codebook(:,1)))<tol
erance | update_codebook2==update_codebook1
        tu=0;
    else
        update_codebook1=update_codebook2;
        clear update_codebook2
    end
end
codebook=update_codebook;
distortion=newdistortion;

```

- โปรแกรมเรียกใช้ฟังก์ชันการสร้าง Code Book

```

clear
clc
%Load training sequence
fid=fopen('C:\Program
Files\MATLAB71\work\Ruj\weight_float\total_weight\total(16x21600).m',
'rb');
training_seq=fread(fid,[16,30000],'float64');
status=fclose(fid);
%define dimension & code book size
dimension=16;
codebook_size=64;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
[codebook,distortion]=VQ_AJ_func(training_seq,dimension,codebook_size
)
```

```
*****
```

6. โปรแกรมหาค่า Index

- ฟังก์ชันหาค่า Index

%โปรแกรมหาค่า Index

```
function[index]=VQ_index_func(training_set,codebook)
```

```
newdistortion=0;
```

```
distortion=0;
```

% ทำการเปรียบเทียบข้อมูลเชิงกับ Codebook ที่คำนวณไว้

```
for k=1:length(training_set(1,:));
```

```
    distortion_block=[];
```

```
    distort=[];
```

```
    for j=1:length(codebook(1,:));
```

```
        distort=sum((training_set(:,k)-codebook(:,j)).^2);
```

```
        distortion_block=[distortion_block distort];
```

```
    end
```

```
    [distortion_min,ind]=min(distortion_block);
```

```
    newdistortion=newdistortion+distortion_min;
```

```
    index(k)=ind;
```

```
end
```

- โปรแกรมเรียกใช้ฟังก์ชันหาค่า Index

```
clear
```

```
clc
```

% Load Codebook

```
load codebook.mat codebook
```

% Load สัมประสิทธิ์ของเสียงแต่ละเสียง

```
fid=fopen('C:\MATLAB7\work\Ruj\weight_float\aa_9.m','rb');
```

```
training_set=fread(fid,[16,200],'float64');
```

```
status=fclose(fid);
```

```
[index]=VQ_index_func(training_set,codebook)
```

```
*****
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. โปรแกรมสร้างโมเดลเสียงโดยใช้วิธีการ HMM

- ฟังก์ชันสร้างโมเดลเสียงโดยใช้วิธีการ HMM

```
function [a_new,b_new,TT]=HMM_func(O)
% กำหนดค่าของจำนวนเซตที่ใช้ในการสร้างโมเดล
N=6;
Plog=0;
% กำหนดค่าของ index ที่อาจจะปรากฏในแต่ละเซต
K=64;
% ทำการกำหนดค่าเริ่มต้นที่ใช้ในการคำนวณ
x=zeros(6,6);
y=zeros(6,K);
z=zeros(1,6);
% ทำการสุ่มค่า a,b เริ่มต้น
a=rand(6,6);
a(1,4)=0; a(1,5)=0; a(1,6)=0; a(2,1)=0; a(2,5)=0; a(2,6)=0;
a(3,1)=0; a(3,2)=0; a(3,6)=0; a(4,1)=0; a(4,2)=0; a(4,3)=0;
a(5,1)=0; a(5,2)=0; a(5,3)=0; a(5,4)=0; a(6,1)=0; a(6,2)=0;
a(6,3)=0; a(6,4)=0; a(6,5)=0;
sm1=sum(a)';
for i=1:N
    for j=1:N
        a_prime(i,j)=a(i,j)/sm1(i);
    end
end
r=rand(6,K);
sr=sum(r)';
for j=1:N
    for k=1:K
        b_prime(j,k)=r(j,k)/sr(j);
    end
end
% กำหนดค่าความน่าจะเป็นของการเริ่มต้นที่เซตต่างๆ
TT=[1 0 0 0 0 0];
T=[1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18];
% ทำการกำหนดกรอบการทำงานของโปรแกรม
for w=1:50
    a=a_prime;
    b=b_prime;
    A_prime=zeros(6,6);
    AM_prime=zeros(1,6);
    B_prime=zeros(6,K);
    BM_prime=zeros(1,6);
% กำหนดค่าของค่าออกผล
    for v=1:120
        for xx=1:18
            lfps=zeros(T(xx),6);
            cB=zeros(1,T(xx));
            for i=1:N
                lf(1,i)=TT(i)*b(i,O(1,v));
                Bt(T(xx),i)=1;
            end
            for t=1:T(xx)-1
                for j=1:N
                    slf=0;
                    for i=1:N
                        slf=slf+lf(t,i)*a(i,j);
                    end
                    lf(t+1,j)=slf*b(j,O(t+1,v));
                end
            end
        end
    end
end
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
end
% คำนวณหาค่าเบต้า
for t=T(xx) -1:-1:1
    for i=1:N
        for j=1:N
            Bt(t,i)=Bt(t,i)+a(i,j)*b(j,O(t+1,v))*Bt(t+1,j);
        end
    end
end
% ทำการสเกลลิ่งค่าเอนโทรปีที่ได้
s1=sum(lf');
for i=1:N
    c(1)=1/s1(1);
    lfps(1,i)=c(1)*lf(1,i);
end
for t=2:T(xx)
    for i=1:N
        lfps(t,i)=0;
        for j=1:N
            lfps(t,i)=lfps(t,i)+lfps(t-
1,j)*a(j,i)*b(i,O(t,v));
        end
    end
end
q=sum(lfps');
for t=1:T(xx)
    c(t)=1/q(t);
    for i=1:N
        lfs(t,i)=c(t)*lfps(t,i);
    end
end
% ทำการสเกลลิ่งค่าเบต้าที่ได้คำนวณได้
sb=sum(Bt');
for t=1:T(xx)
    d(t)=1/sb(t);
    for i=1:N
        Bts(t,i)=d(t)*Bt(t,i);
    end
end
% หาค่า log ของ P จากค่าเอนโทรปีและเบต้าที่ผ่านการสเกลลิ่งแล้ว
log_c=log10(c);
log_p=(-sum(log_c));
% คำนวณหาค่า A_prime และ AM_prime
for i=1:N
    for j=1:N
        x(i,j)=0;
        for t=1:T(xx)-1
            x(i,j)=x(i,j)+lfs(t,i)*a(i,j)*b(j,O(t+1,v))*Bts(t+1,j);
        end
    end
end
A_prime=A_prime+x;
for i=1:N
    z(i)=0;
    for t=1:T(xx)-1
        q=0;
        for j=1:N
            q=q+lfs(t,i)*a(i,j)*b(j,O(t+1,v))*Bts(t+1,j);
        end
    end
end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        z(i)=z(i)+q;
    end
end
AM_prime=AM_prime+z;
for j=1:N
    for k=1:K
        y(j,k)=0;
        l(j)=0;
        for t=1:T(xx)-1;
            G=0;
            for i=1:N
                G=G+lfs(t,j)*a(j,i)*b(i,O(t+1,v))*Bts(t+1,i);
            end
            if O(t,v)==k
                y(j,k)=y(j,k)+G;
            end
            l(j)=l(j)+G;
        end
        if O(T(xx),v)==k
            y(j,k)=y(j,k)+lfs(T(xx),j);
        end
        l(j)=l(j)+lfs(T(xx),j);
    end
end
B_prime=B_prime+y;
BM_prime=BM_prime+l;
Plog=Plog+log_p;
end
end
Plog;
imp=0;
* คำนวณหาค่า a', b' ใหม่
for i=1:N
    for j=1:N
        a_prime(i,j)=A_prime(i,j)/AM_prime(i);
    end
end
for j=1:N
    for k=1:K
        b_prime(j,k)=B_prime(j,k)/BM_prime(j);
    end
end
* ฟังก์ชันปรับค่า b_prime
for j=1:N
    for k=1:K
        if b_prime(j,k)<1.00e-4
            b_prime(j,k)=1.00e-4;
        end
    end
    imp=sum(b_prime');
    b_prime(j,:)=b_prime(j,:)/imp(j);
end
end
* แสดงผลและจัดเก็บโมเดลของเตียงที่คำนวณได้
a_new=a_prime;
b_new=b_prime;
TT;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรแกรมเรียกใช้ ฟังก์ชันสร้างโมเดลเสียงโดยใช้วิธีการ HMM

```
clear
clc
% Load index ของเสียงแต่ละเสียง
fid=fopen('C:\Program
Files\MATLAB71\work\Ruj\weight_float\Code_AJ\index_9.m', 'rb');
O=fread(fid, [18,120], 'float64');
status=fclose(fid);
[a_new,b_new,TT]=HMM_func(O)
```

8. โปรแกรม Recognize

- ฟังก์ชัน Recognize

```
% โปรแกรม Recognize
function [Recognize_word]=Recognize_func(training_set)
N=6;
K=64;
% ทำการโหลดค่า Codebook ที่กำหนดไว้
load codebook.mat codebook
[index]=VQ_index_func(training_set,codebook)
% กำหนดค่า Index ที่ได้ให้เป็น Input ของส่วนของ Recognize
O=index;
% กำหนดค่า T ให้เท่ากับจำนวนของกนในกลุ่มทศสองที่ใช้ในการสร้างโมเดลเสียง
T=120;
a=0;
% ทำการโหลดค่า a_new, b_new, TT จากแต่ละโมเดลตั้งแต่ 0-9
for c=1:10
    if c==1
        load HMM_0.mat
    end
    if c==2
        load HMM_1.mat
    end
    if c==3
        load HMM_2.mat
    end
    if c==4
        load HMM_3.mat
    end
    if c==5
        load HMM_4.mat
    end
    if c==6
        load HMM_5.mat
    end
    if c==7
        load HMM_6.mat
    end
    if c==8
        load HMM_7.mat
    end
    if c==9
        load HMM_8.mat
    end
    if c==10
        load HMM_9.mat
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end
% clear ค่าตัวแปรที่ใช้ในการคำนวณ
ar=0;
dt=0;
d=0;
% ส่วนของการตัดสินใจโดยใช้Viterbi Algorithm
% ส่วนของการ Initialization
for i=1:N
    dt(1,i)=TT(i)*b_new(i,O(1));
    ar(1,i)=0;
end
% ส่วนของการ Recursion
for t=2:T
    for j=1:N
        for i=1:N
            d(i)=dt(t-1,i)*a_new(i,j);
        end
        [dm,ar(t,j)]=max(d);
        dt(t,j)=dm*b_new(j,O(t));
    end
end
% ส่วนของการ Termination
dt=dt';
[p_st(c),q_st(T)]=max(dt(:,T));
for t=T-1:-1:1
    q_st(t)=ar(t+1,q_st(t+1));
end
q_st;
p_st(c);
% clear ค่าของ a_new, b_new
clear a_new b_new
end
% โปรแกรมจะทำการเปรียบเทียบเสียงที่ต้องการทดสอบกับ โมเดลของแต่ละเสียงจนครบทุกโมเดล และจะทำ
% การหาว่าเสียงที่เข้ามามีค่าใกล้เคียงกับโมเดลเสียงใดมากที่สุดก็จะชี้เป็นค่าของเสียงนั้นๆ
[p,num]=max(p_st);
Recognize_word=num-1;

- โปรแกรมเรียกใช้ฟังก์ชัน Recognize
clear
clc
% Load สมบัติของเสียงที่ต้องการทดสอบ
fid=fopen('C:\Program
Files\MATLAB71\work\Ruj\weight_float\yui\yui_9.m','rb');
training_set=fread(fid,[16,200],'float64');
status=fclose(fid);
[Recognize_word]=Recognize_func(training_set)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ๑.

Codebook ลำดับที่	1	2	3	4	5	6	7	8	9	10	11	12	13
1	-4.5159	-4.858	-5.0685	-6.2102	-7.2648	-7.2442	-7.3629	-7.7076	-5.0875	-6.7865	-6.982	-7.0891	-7.246
2	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505
3	3.0465	2.8837	1.9436	5.2315	11.477	8.6181	7.0891	7.3145	3.995	8.7436	6.6293	6.1821	7.7553
4	7.9012	7.1371	5.7499	10.498	29.918	18.597	13.966	14.956	8.6014	17.522	13.067	11.808	16.326
5	10.593	9.5943	7.0682	16.188	59.978	35.122	24.084	25.668	12.886	31.987	22.272	19.287	29.732
6	14.816	12.966	8.8706	25.092	124.29	64.453	40.116	43.872	18.826	57.503	36.409	30.467	52.302
7	19.956	17.065	11.171	37.047	251.82	115.86	65.685	73.077	26.302	101.43	58.826	47.002	90.286
8	27.428	23.131	14.148	56.065	513.64	207.24	107.58	121.3	37.628	177.74	94.827	72.706	155.37
9	36.298	29.884	17.335	81.47	1025.7	363.22	171.59	196.92	51.749	304.71	148.53	109.63	261.32
10	46.099	37.145	20.014	115.14	1995.1	620.46	267.12	311.54	69.129	509.13	226.84	160.96	428.46
11	56.341	44.255	22.039	156.84	3740.6	1020.9	399.56	474.56	88.657	819.25	333.08	227.87	676.03
12	65.887	50.399	23.407	202.81	6641.9	1592.8	567.48	685.74	108.24	1250.2	464.69	305.87	1012.2
13	70.735	52.815	22.986	240.89	10828	2280.7	739.84	909.49	121.51	1751.2	595.03	376.99	1390.9
14	64.402	46.948	19.071	242.97	14998	2774.5	819.37	1024.7	115.79	2083.8	647.15	394.67	1623.6
15	36.516	25.973	9.8494	152.47	12922	2099.7	564.5	718.2	68.641	1542.6	437.84	257.04	1179
16	-29.774	-20.667	-7.3121	-137.67	-16021	-2286.4	-559.59	-724.32	-58.536	-1643.1	-426.25	-240.88	-1231.9

ตารางที่ 1 ค่าโคตบิตตัวที่ 1-13 ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Codebook ที่ ลำดับที่	14	15	16	17	18	19	20	21	22	23	24	25	26
1	-7.3489	-7.9204	-7.0712	-12.614	-5.4706	-4.405	-13.66	-7.7744	-6.3801	-7.7219	-6.8791	-5.6271	-4.6972
2	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505
3	7.6349	9.9626	5.6532	2.1866	4.8491	7.4228	1.2754	11.186	13.527	6.5882	9.5361	4.931	4.3785
4	15.685	22.505	10.875	4.9301	0.59445	7.2854	1.7906	25.263	33.801	12.597	20.643	9.7311	9.2776
5	27.704	44.312	17.752	5.389	-7.8749	2.1642	-1.2186	50.451	72.724	20.678	39.651	14.934	13.901
6	47.959	84.974	27.709	6.5472	-13.04	-3.3563	-2.7332	99.179	156.15	33.605	74.748	22.87	20.63
7	80.876	161.29	41.806	7.7143	-12.108	-8.1276	-3.9497	194.44	333.59	52.94	138.12	33.277	29.684
8	136.59	304.72	63.946	9.2963	-5.0805	-11.619	-3.7755	379.27	712.1	83.45	254.31	49.128	43.151
9	225.79	564.68	94.749	10.521	4.9861	-14.779	-2.6999	725.3	1490.6	128.42	459.1	69.968	60.362
10	363.47	1019.3	136.74	11.398	16.85	-14.311	-1.083	1351.2	3039.5	192.69	807.5	97.08	82.246
11	563.08	1773.2	189.65	11.876	23.081	-10.743	0.18277	2426.2	5972.1	278.13	1369	129.69	107.69
12	827.61	2923.6	250.05	11.791	19.293	-5.1465	0.94836	4127.7	11117	380.86	2199.4	164.45	134.01
13	1116.5	4424.5	302.68	10.879	8.0134	0.9911	1.1756	6446.6	18998	478.86	3243.2	191.69	153.35
14	1279.7	5688.8	311.16	8.4428	-3.492	4.911	0.82519	8554.4	27585	511.44	4063.3	189.73	149.01
15	912.39	4550.4	199	4.1071	-5.7277	3.9085	0.28196	7062	24918	339.79	3167	116.8	90.053
16	-936.02	-5237.5	-183.12	-2.8712	5.2549	-3.3081	-0.094453	-8389.4	-32389	-324.84	-3551.9	-103.47	-78.3

ตารางที่ 2 ค่าโคตบคตวที่ 14-26 ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Codebook ที่	27	28	29	30	31	32	33	34	35	36	37	38	39
1	-3.4808	-3.6256	-1.6829	-2.7171	-5.1341	-0.52598	-3.3993	-2.8359	-5.4628	-13.523	-35.359	-0.55017	-3.3464
2	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505
3	3.4955	6.9533	-8.0884	3.7739	3.3588	-7.541	-6.6273	-8.7176	2.5574	1.5251	1.31	0.70915	-6.161
4	8.2809	4.656	5.7976	3.4277	-2.5469	9.458	-4.7632	0.55445	6.5728	3.3276	0.41153	5.1157	-3.1973
5	11.784	-2.5752	6.9995	-1.1384	-9.6395	0.38461	12.762	12.386	8.4427	3.4051	-2.7291	5.6223	9.3269
6	16.758	-9.3833	-7.0771	-3.8941	-11.186	-6.9297	12.191	0.085359	10.936	3.6811	-4.1672	6.0934	8.4164
7	23.273	-13.418	-3.4876	-5.0674	-4.9466	9.83	-18.42	-17.583	14.013	3.8124	-4.9779	7.8006	-10.737
8	32.484	-14.038	11.605	-3.7573	5.7769	3.094	-22.261	-0.23163	18.536	4.4975	-3.9095	9.9494	-12.839
9	43.881	-12.266	2.0499	-0.78403	14.803	-8.453	22.058	24.334	23.347	4.5692	-2.5739	11.328	10.42
10	57.21	-4.8135	-11.313	3.0569	19.171	7.4097	37.614	0.76373	28.101	4.4174	-0.061425	12.603	19.114
11	71.779	5.4036	3.6157	5.3968	12.754	5.3001	-19.476	-29.42	32.41	4.1378	1.7627	13.507	-6.1752
12	85.902	13.593	10.119	6.2512	-0.49769	-7.6056	-51.804	-1.0127	35.775	3.6207	2.5503	13.415	-21.491
13	94.427	17.281	-4.9174	5.1893	-11.847	2.8917	8.1091	28.192	36.36	2.9811	2.4015	12.434	0.60414
14	88.112	14.044	-4.3258	2.8101	-13.286	4.0375	44.139	0.70046	31.326	2.0547	1.4406	9.9375	15.187
15	51.15	5.492	2.5049	0.64978	-5.2503	-1.8104	1.8856	-11.135	16.804	0.90212	0.39491	4.8684	1.701
16	-42.709	-2.067	-0.85598	-0.0086492	1.0698	0.037494	16.885	0.24356	-12.963	-0.56449	-0.057997	-3.408	4.7012

ตารางที่ 3 ค่าโคตบคิ้วที่ 27-39 ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Codebook ที่	มิตที่	40	41	42	43	44	45	46	47	48	49	50	51	52
1		-5.7333	-7.0036	-1.0032	-3.4109	-13.386	-5.306	-4.3063	-4.2091	-3.4294	-3.1278	-2.2511	-4.2888	-1.0869
2		4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505
3		1.6924	0.70969	-0.21416	-6.4896	-0.16367	0.97018	-4.8321	-0.33032	-2.1003	-7.9744	-9.7779	-1.3593	0.90145
4		-4.4608	-0.025479	3.5724	1.2976	0.50945	-5.7417	-1.2739	-4.9402	1.5275	-1.9328	2.9606	-3.342	-7.077
5		-10.609	-5.0182	3.5286	7.1402	0.4022	-9.9487	5.0682	-4.5489	0.098739	13.15	13.85	-1.0397	-10.607
6		-7.6173	-4.39	3.8703	0.3351	0.9033	-2.0368	4.6741	4.2851	0.49269	4.9609	-6.2474	5.7223	3.0305
7		2.4639	-2.6799	4.2974	-7.4399	0.65409	9.812	-3.7562	9.0595	0.41406	-19.005	-19.276	5.5448	17.673
8		13.651	0.71482	5.2795	1.1316	0.87275	16.264	-4.5022	6.162	0.65349	-9.4375	11.595	0.54552	19.147
9		16.932	4.4558	5.8004	9.3571	0.51108	9.5471	3.9392	-3.7251	1.6658	25.676	26.998	-4.4881	0.1661
10		11.038	5.5036	6.0662	-0.26464	0.0090252	-5.4549	5.9376	-8.6943	1.2107	16.337	-19.1	-2.0372	-23.137
11		-4.0303	4.3643	5.9863	-8.1188	-0.067897	-19.008	-1.156	-5.2388	0.83982	-30.01	-32.351	2.982	-31.497
12		-16.495	1.5908	5.3893	0.38048	-0.15041	-16.849	-4.451	3.2845	0.57941	-22.204	26.414	4.5593	-9.2827
13		-16.725	-0.92538	4.496	6.3821	-0.083286	-1.9294	0.15432	6.9545	0.20803	27.536	29.657	1.684	20.162
14		-6.2976	-1.6104	3.2913	-0.04428	-0.057822	9.8682	2.635	3.1669	0.11979	19.741	-24.115	-1.1374	24.747
15		1.9968	-0.80293	1.4987	-1.8587	-0.015094	6.569	0.34433	-0.65318	0.088901	-10.239	-10.871	-0.69173	5.8904
16		-3.9625	0.344	-0.95261	-0.074559	0.010358	-2.2926	0.56346	1.2677	-0.049491	8.0654	-10.017	-0.028281	4.1072

ตารางที่ 4 ค่าได้ตัวที่ 40-52 ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Codebook ที่	53	54	55	56	57	58	59	60	61	62	63	64
1	-4.191	-1.8826	-3.4224	-1.6292	-2.8804	-0.47397	-2.7649	-0.56524	-2.5625	-1.9325	-2.3999	-1.4962
2	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505	4.0505
3	-3.2099	-1.0843	-5.3897	-0.3121	-6.3367	-2.1905	-3.6711	-1.5447	-5.1437	-2.4126	-6.9396	-3.5418
4	-3.3132	-7.5472	-4.3532	-10.204	-6.1236	-8.964	-5.9031	-9.7141	-7.2016	-6.0006	-6.846	-8.1463
5	2.096	-5.3669	5.8821	-8.9459	8.8402	-3.4888	1.9861	-5.3495	5.4241	-1.6441	11.431	0.027878
6	6.917	8.9434	11.018	10.436	15.656	15.985	12.211	14.431	15.944	11.159	18.606	16.93
7	1.8801	15.364	-3.676	24.628	-6.965	17.119	4.5175	20.868	1.5527	9.8687	-11.979	11.307
8	-4.8945	5.0395	-14.832	14.036	-26.856	-5.6539	-11.748	1.3962	-22.12	-4.2943	-37.074	-14.675
9	-4.477	-15.211	-2.7518	-19.695	-2.1616	-30.374	-13.7	-29.316	-16.785	-16.212	4.6881	-28.742
10	3.8415	-19.974	16.945	-42.358	35.014	-19.317	6.0179	-30.791	20.444	-6.439	54.965	-1.5749
11	7.055	-3.4261	11.441	-22.783	19.491	21.767	19.524	7.1825	33.688	12.689	18.154	35.614
12	1.2605	18.67	-12.045	28.801	-32.836	42.38	6.5164	44.655	-3.8987	16.784	-64.209	29.13
13	-3.6775	19.121	-14.241	51.437	-32.981	11.363	-12.795	30.787	-34.834	1.3767	-45.018	-15.481
14	-1.8593	0.65354	3.6042	18.859	14.967	-25.873	-10.374	-13.41	-12.134	-10.477	41.511	-31.553
15	0.6826	-6.7851	5.8641	-12.688	17.026	-15.303	1.601	-19.096	10.933	-4.1927	29.009	-4.6603
16	-0.76874	4.6258	-0.58287	16.736	0.46006	-0.60698	-4.0059	7.3404	-8.7511	-1.1527	8.8105	-9.3077

ตารางที่ 5 ค่าตัดปกติที่ 53-64 ที่คำนวณได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้